

---

*The Application Of Active Contour Models  
To MR and CT Images*

*D.N. Davis*

*Medical Vision Group  
Department Of Computer Science,  
University Of Birmingham,  
Edgbaston,  
Birmingham,  
B15 2TT*

*February 1995*

---

---

## Contents

1.	Introduction	3
2.	Domain	4
3.	Introduction To Active Contour Models	5
4.	Adopted Model	6
5.	Advances On The Adopted Model	8
6.	Future Work	15
7.	Conclusion	17
8.	References	18
9.	Appendix A Using The Snake Testbed	20
10.	Appendix B Using The Advanced Segmentation Tool	32

---

---

## 1. Introduction

This document reports on the work on active contour models (or snakes) undertaken as part of the SAMMIE project (AIM project A2032) on the development of advanced segmentation aides for object demarcation in MR and CT images [1,2].

Active contour models (or *snakes*) [3] are a pattern matching method which is gaining favour in many machine vision application areas. Snakes are a special form of deformable models, and are characterised by their property of dynamic deformation to an image from an original given shape. This deformation is controlled through the minimisation of an energy function. The active contour models discussed here are visually represented as closed contours (like an irregular balloon or bubble).

The following section gives a brief overview of the Computer Tomography (CT) and Magnetic Resonance (MR) imaging modalities. This is followed by a section introducing active contour models, or snakes. The adopted active contour model is then detailed, followed by a breakdown of the work undertaken in developing this for CT and MR images. The document then considers what future work on active contour models may be useful. The document finishes with a few conclusions that can be drawn from our work on active contour models. The use of the snake within the Snake Testbed for CT, MR and other image types is covered in appendix A. The use of the snake within the advanced segmentation tool for CT, MR image types is covered in appendix B.

---

---

## 2. Domain

Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are two very differing methods for producing three-dimensional radiological image data sets. CT [4] is a digital radiological technique using low dose X-rays and allow some degree of flexibility including the facility to control and manipulate the formation and presentation of images. MRI is different in that X-ray radiations are not used, but relies on very high magnetic fields and radio-frequency of a specific wavelength. A good overview of MRI can be found in [5] and [6]. The MR signal depends on four almost different physical parameters: the density of hydrogen nuclei, their movement and the relaxation times (T1 and T2) related to biochemical environment of the nuclei. This multi-parametricity gives rise to a great degree of freedom in the imaging possibilities associated with MRI.

They are increasingly popular and favoured digital imaging modalities, and are particularly useful for imaging both soft and hard (bony) tissue. MRI typically gives better soft tissue definition than CT, but they are both favoured in medical domains for producing volumetric image sets; the granularity of the z axis can be quite fine (1.5 mm or less). A good overview of atlases and both CT and MRI can be found in [7], while [8] gives an overview of cranial computed tomography. The type of images produced can be controlled in both cases to suit the type of investigation required, highlighting different aspects of the imaged body part.

MR and CT vision work has been under investigation since the early 1980s. The methods for the interpretation of these forms of image are more complex than for normal projective radiology. Techniques range from threshold-based techniques for extracting the MR imaged brain [9] to knowledge-based classification and tissue labelling [10,11]. Earlier work include the IBIS system [12] for the labelling of anatomical regions in 3D MRI brain data sets, chest CT image interpretation [13], a blackboard system for both CT and MR images of the head [14], the finding of the thalamus and VL nucleus in CT images [15], region labelling in CT images of the brain [16], 3D image analysis of CT brain scans [17] and the use of neural nets in an expert system for MRI analysis [18]

There has also been considerable work (world wide) on producing idealised and pathological digital atlases that are of use in both training and diagnosis using both MR and CT images. These include sophisticated environments such as the Brainworks environment (AIM project A2032)

---

---

### 3. Introduction To Active Contour Models

Snakes were originally proposed by Katz, Terzopoulos and Witkin [3,19,20] for a number of image analysis tasks. Numerous researchers have developed these initial ideas [see 21], looking to improve the computational ease through the use of finite elements [22], or combining the deformable nature of active contour models with deformable models [23].

The standard active contour energy function is usually given with relation to the entire contour. The active contour model is defined as an energy minimising spline; with its energy dependant upon its shape and location within an image. Typically, local energy minima correspond to desired image properties. The energy function to be minimised is a weighted combination of internal and external energy forces. Snakes lend themselves to object delineation, particularly where guided by higher level understanding processes (whether interaction with human or guidance from an automatic system). At the very least, they require a starting shape and location, preferably near to that of the desired object. Snakes can be open (like a length of string) or closed (more like balloons).

It is possible to use an active contour model that does not look at the overall energy of the snake, but the effect of moving nodes (shape defining points) within the model, minimising the energy function for each node in turn, and allowing the snake as a whole to adapt dynamically through repeated iteration. Where this is the case, the energy at any node is given as:-

$$E_{node} = E_{int}(v(s)) + E_{image}(v(s)) + E_{constraint}(v(s))$$

where  $v(s)$  represents a node on the curve,  
 $E_{int}$  the internal energy of the contour,  
 $E_{image}$  the image forces,  
 $E_{constraint}$  the external constraints.

Summing the energy at every node (and performing some normalisation) gives the energy value for the snake as a whole.

The internal energy at any given node is combination of forces controlling elasticity and stiffness, and tends to produce smooth contours. These forces are calculated using the position of the node under consideration and its neighbours together with minimum and maximum values for such measures over the entirety of the active contour model.

The image energy of the snake at any given node is typically derived from some image processing technique, that is suitable for any chosen image domain, such as a first order edge gradient operator like the Sobel.

The external forces affecting the active contour model are typically based on higher order constraints relating to more global strategies, such as the relation to other objects in the image or coercive forces forcing the snake towards or away from particular areas local to the snake. The provision of a means for specifying these external energy factors allows a degree of flexibility in the use of the active contour model for finding accurate contours in noisy or busy image areas.

---

---

## 4. Basic Adopted Model

The active contour model developed here was originally based on that proposed by Williams and Shah [24], and implemented by the Multi-Media Research Information Systems Group at The University Of Manchester. This active contour model works in terms of local energy functions at shape defining nodes. In its initial state (as acquired from Manchester), the snake used an energy function of the form:

$$E_{node} = \text{Alpha} * V_a + \text{Beta} * V_b + \text{Gamma} * V_g$$

where	<i>Alpha</i>	the elasticity coefficient
	<i>Beta</i>	the stiffness coefficient
	<i>Gamma</i>	the image energy coefficient
	$V_a, V_b, V_g$	the measured values at the node.

There are a considerable number of parameters that affect the behaviour of the active contour model, these can be split into four main groups: internal energy parameters; image energy parameters; constraint energy parameters; and a more disparate set of general behaviour constraints. These parameters are detailed below in the section on the implementation adopted in the project, be will be referenced in the next section.

The original algorithm worked as follows:-

### 1. Initialisation.

- i) Choose source image
- ii) Draw seed contour in approximate shape and location
- iii) Ensure control nodes are within minimum and maximum distance thresholds
  - a) Remove node too close to neighbour
  - b) Add node when neighbours too far apart
- iiii) Set up parameters

### 2. Call minimisation routine

- i) Perform Sobel operator on source image
- ii) Start iteration
- iii) Call williams-shah algorithm
- iiii) Continue iteration until
  - a) no change
  - b) iteration parameter exceeded

### 3, Williams-Shah algorithm

- i) Find max-min values for curvature, gradient, distance etc
- ii) Start at first node on snake
- iii) Calculate energy at this node
- iiii) Calculate energy if node were to move to first point over  $n*n$  grid.
- v) Node migrates to new snake if energy lower at this point
- vi) Continue for all possible points in  $n*n$  grid
- vii) Continue for all nodes
- viii) Return new snake (may be exactly same as given snake)

There are two components to the internal energy measure at any node; both take into consideration the neighbouring nodes. The effect of these positive measures (elasticity and stiffness) is controlled by the *elasticity* and *stiffness* coefficients. As *alpha* is decreased (from a nominal value of 1.0), the strength of the elasticity of the contour at a node is decreased, giving nodes greater freedom in movement and so possibly producing more irregular contours. As *beta* is decreased (from a nominal value of 1.0), the stiffness of the contour at a node is

decreased, allowing sharper corners to develop. It has been found through experimentation that any change in the elasticity co-efficient should typically be accompanied by a similar change in the stiffness coefficient. The image energy component reflects the value in the Sobel operator image and is controlled through the *gamma* coefficient. The (Sobel) gradient image term has the effect of trying to pull nodes towards steep gradients. As the value of *gamma* is decreased (from a nominal value of 1.0), the energy minimising effect of steep gradients is decreased. Setting *gamma* to zero turns off the effect of the gradient image all together.

There are a number of other parameters that affected the nature of the adopted active contour model. The *Iterations* parameter controls the maximum number of overall contour energy minimisations that will take place on each call; giving it a value of 0 will not cause any changes to be made. If the contour has not changed after an iteration (that is, there has been no change in the position of any contour defining node) the active contour module returns.

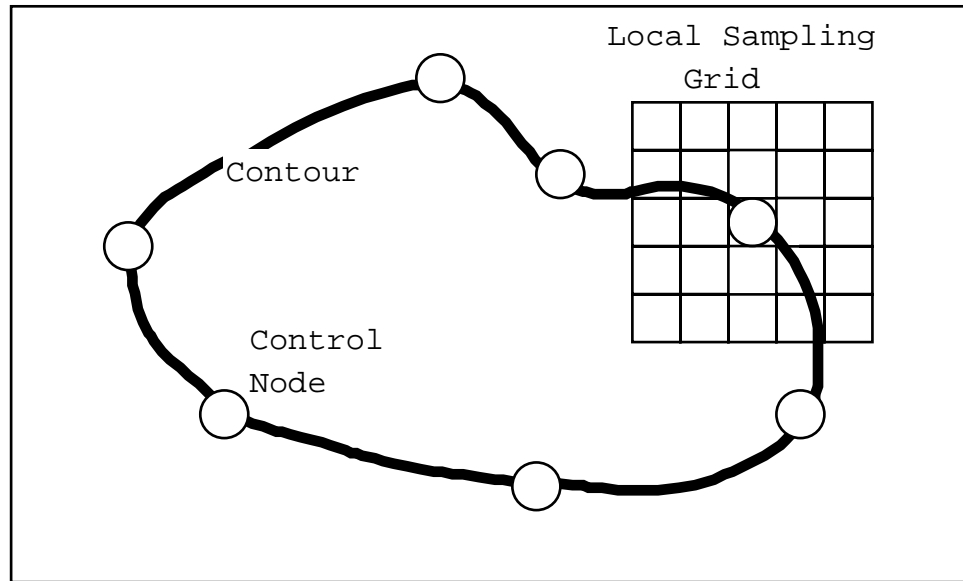


Figure 1. Closed contour with control nodes and sampling grid

The *window* controls the sample size at each node (using a *windows* sized square discrete grid over which a node could move (see Figure 1); giving it a value of 0 will result in the contour failing to move, irrespective of any energy function parameters, as no alternative points for any node will be considered. A too large value will result in the contour making dramatic and erratic shapes in its morphology.

Two other important parameters were defined as *Minus* and *Maxus*; these two parameters controlled the minimum and maximum allowed (straight line) distance between nodes; any drawn initial contour and any contour resulting from a call to the active control model would be redefined in terms of these two distances.

The adopted active contour model algorithm would provide reasonable contours to well-defined high contrast features, such as the internal ventricles in MR and CT images but would fail to provide acceptable candidates for less well defined objects such as the thalamus and other white and grey-matter structures to be found in the imaged brain. For this reason a number of experiments were run to find changes to the original algorithm that would produce more acceptable contours; these are detailed below.

## 5. Advances On The Basic Model

There are a number of simple modifications that were tried in order to improve the efficacy of the snake for our application. These include changes in the curvature measures, changes to the gradient operators and the addition of an extra component to the image energy. We also allowed the use of attractor, repulsor and static nodes. These initial changes are grouped together under the different energy components and the functional aspects of the active contour model algorithm. In its final state, the snake used an energy function of the form:

$$E_{node} = \quad \text{Alpha} * V_a + \text{Beta} * V_b + \\ \quad \text{Gamma1} * V_g + \text{Gamma2} * V_x + \\ \quad \text{CoRepulse} * V_R + \text{CoAttract} * V_A$$

where	<i>Alpha</i>	the elasticity coefficient
	<i>Beta</i>	the stiffness coefficient
	<i>Gamma1</i>	the edge gradient image energy coefficient
	<i>Gamma2</i>	the zero-crossing image coefficient
	<i>CoRepulse</i>	the Repulsor coefficient
	<i>CoAttract</i>	the Attractor coefficient
	<i>V<sub>a</sub>, V<sub>b</sub></i>	the measured internal values at the node
	<i>V<sub>g</sub></i>	the edge gradient image value at the node
	<i>V<sub>x</sub></i>	the zero crossings image value at the node
	<i>V<sub>A</sub></i>	the measured attraction value at the node
	<i>V<sub>R</sub></i>	the measured repulsion value at the node

The meaning and use of these terms becomes clearer in the following subsections.

### *New Algorithm*

The final algorithm for the active contour model developed under the auspices of the SAMMIE project works as follows:-

#### *1. Initialisation.*

- i) Choose source image*
- ii) Draw seed contour in approximate shape and location*
- iii) Ensure control nodes are within minimum and maximum distance thresholds*
  - a) Remove node too close to neighbour*
  - b) Add node when neighbours too far apart*
- iiii) Set up parameters*

#### *2. Call minimisation routine*

- i) Perform gradient operator on source image*
- ii) Perform zero-crossing or canny approximation*
- iii) Perform basic node checking*
- iiii) Start iteration*
- v) Call Global Shift*
- vi) Call variation on Williams-Shah algorithm*
- vii) Perform node checking if flagged*
- viii) Continue iteration until*

- a) no change
- b) iteration parameter exceeded

### 3. Global Shift

- i) Find max-min values for curvature, gradient, distance etc
- ii) Move all nodes to first offset on  $m*m$  global shift grid
- iii) Store position if lower energy
- iv) Repeat for all positions on global shift grid
- v) If lower energy snake found, move snake to that position.

### 4. Variation on Williams-Shah algorithm

- i) Find max-min values for curvature, gradient, distance etc
- ii) Start at first node on snake
- iii) Calculate energy at this node
- iiii) Calculate energy if node were to move to first point on profile.
- v) Node migrates to new snake if energy lower at this point
- vi) Continue for all possible points in profile
- vii) Continue for all nodes
- viii) Return new snake (may be exactly same as given snake)

We can now turn the various components of this changed algorithm.

### Sampling Space

One change made to the active contour model, was the adoption of a profile based sampling space for the consideration of alternative coordinates for node placement, rather than a  $n*n$  discrete grid. Our early experiments with the original sampling method showed that ill-formed snakes were being produced too easily, so we constrained the search space.

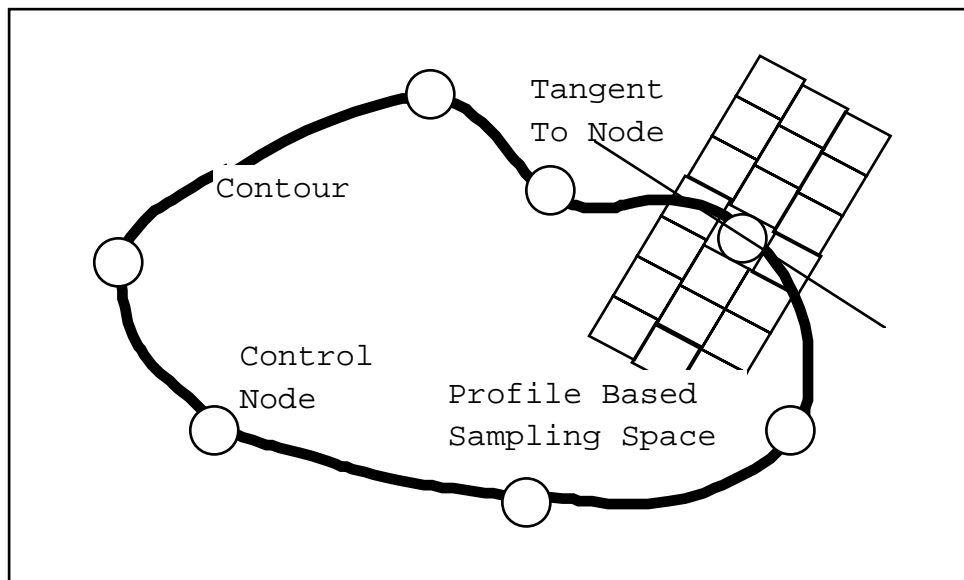


Figure 2 The New Sampling Space

The tangent at a node is easily computed, so is the perpendicular to the tangent. The perpendicular to the tangent provides the basis for the profile space-sampling model adopted. Here the *windows* parameter rather than govern the size of a rectangular grid, is used to establish the point length of the profile; we actually use the original node coordinate plus a (*windows* length) profile out, and (*windows* length) in from that point. At each point of the profile, we consider a *windows* length tangent, with the central pixel on the normal profile (see figure 2). This actually gives us a sampling space of  $2*windows+1$  by *windows*, but unlike the original scheme

orientated about the tangent (and its normal) at the control node. This gave a definite improvement in the snake, at a slight computational cost. We also tried under sampling this space, to reduce the number of possible alternative positions for a node and so improve computational efficiency, but this gave poorer results.

### *Internal Energy*

A few changes to the method of calculating the two components to the internal energy were tried; this typically involved increasing the neighbourhood for the calculation of the internal energy components. These changes gave no observed improvement for the extra computational cost, so they were dropped.

### *Image Energy*

The image energy of the snake at any given node was changed so that it combines the output of two image processing techniques, and tends to shift nodes towards steep gradients in the image. One of these image processing techniques is a difference of Gaussian operator which produces a black image containing white edges (*zero crossings*). The other operator is a different form of edge operator which produces an edge gradient image.

The zero crossings edge image (based on the Marr-Hildreth edge operator [25]) has the effect of pulling nodes towards zero crossings *and* away from non-zero crossings. The relevant coefficient, *Gamma2*, (taking a nominal value of 0.25) plays a similar role to *Gamma1* in that controls the extent of this movement. Setting *Gamma2* to 0 turns off the effect of zero crossings all together. The zero crossings operator is nominally given a lower Gaussian value, and the upper Gaussian value can be taken as 1.8 times the value of the lower Gaussian.

As the zero crossing image is a binary image (where a point is either *On* or *Off*), there is not the range of normalised values associated with this image as there is with the gradient image. If we were to simply use an integer (Boolean) value for a zero crossing image point, it is possible to produce a fluctuating erratic contour model. We have therefore designated two non-integer values for the two states; *edge\_off\_weight* (taking a nominal value of -0.10) for non-zero crossing points; and *edge\_on\_weight* (taking a nominal value of 0.15) for zero crossing points.

It has been found that the Canny edge operator [26], with a SD of half the sum of the lower and upper Gaussian, acts as a good approximation to the zero crossing operator in most images but with less computational expense.

Four alternative processes are allowed to produce the edge gradient map: the Mero-Vassey [27]; the Sobel edge detector [28]; a bidirectional morphological edge operator [29]; and the Canny edge operator [26]. The user is given control over which operator to use, with the Sobel as the default. Experiments show that the Sobel always out performed the Mero-Vassey, while the extra computational cost of the morphological operator was rewarded with better gradient information in gray-matter areas. For some areas in some images (notably grey-matter areas in CT images), none of the operators provided reliable information, and that it normally paid to reduce the gradient coefficient (*Gamma1*).

Improvements in computational efficiency were found by using a region of interest (ROI) for both image energy operators. The ROI is specified automatically to encompass the contour plus a border large enough to allow a considerable movement in the snake (during a global shift - see below) and in any change in morphology due to node migration.

While we shall come to the fifth edge operator (or more correctly, the maximum likelihood edge detector MLD) at the end of this section (and in the section on future work); the testbed used for the development of the active contour model (and indeed the final Advanced Segmentation Module available in the Radiology Department at the Queen Elizabeth Hospital) does allow the use of this operator. When this is selected, no gradient image is used for local node migration as the MLD is one dimensional operator, but the Canny gradient image is used in the global shift algorithm (see below).

### *Constraint Energy*

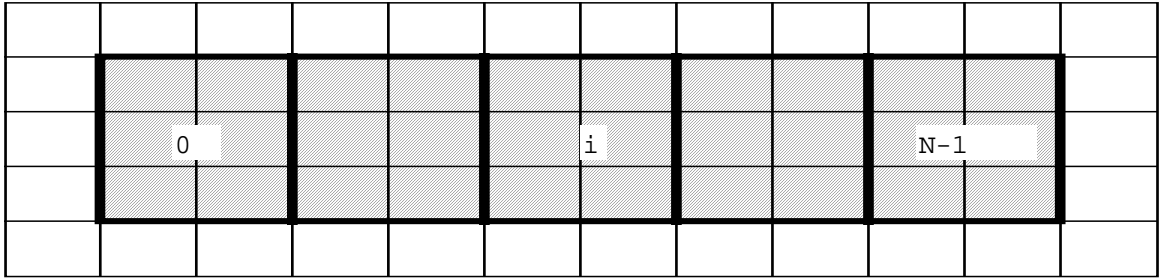
The external constraint forces affecting the active contour model are attractor and repulsor forces from points in the image specified through user interaction. Attractor points are used to pull the contour towards a specific area, while repulsor points are used to ensure that the contour does not easily move towards a specific area. These two types of external force are not necessary for many features, and the active contour model will run without them; the provision of a means for specifying them allows a degree of flexibility in the use of the active contour model for finding accurate contours in noisy or busy image areas.

Repulsor points are used to stop the contour moving towards the areas where are placed; typically image areas of great contrast that are dragging the contour away from its desired shape. Repulsor points affect the movement of nodes in inverse ratio to the (normalised) distance between contour nodes and each repulsor point. A coefficient (*Corepulse*) controls the severity of this effect. As this coefficient is increased (from its nominal value of 0.15) the repulse energy is increased; conversely reducing this coefficient to 0 completely negates the effect of all the repulsor points. There is no limit to the number of repulsor points that can be used. We have found that when attempting to find highly convoluted brain structures (such as the sulci); setting up a central axis of repulsor points in the seed contour (with a low CoRepulse coefficient of less than 0.25) can be beneficial and tends to stop narrow contours crossing themselves. This will be discussed further in the section on future work.

Attractor points are used to coerce the contour into moving towards the areas where are placed; typically image areas of poor contrast that are failing to drag the contour into its desired shape. Attractor points affect the movement of nodes in inverse ratio to the (normalised) distance between contour nodes and each attractor point. A coefficient (*Coattract*) controls the severity of this effect. As this coefficient is increased (from its nominal value of 0.15) the attraction energy is increased; conversely reducing this coefficient to 0 completely negates the effect of all the attractor points. There is no limit to the number of attractor points that can be used.

### *Dipoles and maximum likelihood detectors*

Dipoles are one dimensional edge detectors devised at the Wolfson Image Analysis Unit in Manchester [30]. They work by moving a window along a profile (an arbitrarily orientated line with a sampling width). Image informatics relating to the contents of the window give rise to informatics about the likelihood of an edge at that point along the profile. Maximum likelihood detectors are a development of this idea [31], but require no moveable sampling window. Instead, a moving divisor travels along the profile, and statistically based information about the likelihood of an edge at the divisors current position are amassed.



*Figure 3 : A N Sized Dipole Placed On An Image Section*

Figure 3 shows a  $N$  sized MLD profile placed over a section of image. The size of the operator (here  $N = 5$ ) is superimposed over an arbitrarily orientated profile of width  $W$  and length  $D$ . Typically the profile is orientated according to some higher order constraints; in the snake we use the perpendicular to a tangent at a contour controlling node. We arrange that  $W$  takes an odd integer value, while  $D$  is related to the local sample space parameter (actually  $D = 4 * local$ ). While we can specify the number of sampling spaces in the MLD ( $N$ ), we have to ensure that the sampling space is great enough to allow meaningful statistics to be gathered (the standard deviation of the image pixel values in each sampling space). If this value is not great enough (we use a minimum value of 10 pixels per MLD sampling space, although there is room for experiment here),  $W$  is increased (in steps of 2 to remain odd). The MLD then produces maximum likelihood statistics for each sampling space. This it does by evaluating the standard deviation for the MLD as a whole; and the sds for each windows either side of the  $i$ th sampling space, from which the maximum likelihood can be calculated as follows:-

$$\text{Maximum Likelihood, } w_i = \frac{SD_{\text{profile}}^{N/2}}{[SD_A^{i/2} * SD_A^{(N-i)/2}]}$$

This means that no statistics are gathered for the zero and  $(N-1)$  positions as there exists no window on one side for these positions.

The MLD operator (when used in the snake works as follows):-

1. Call MLD operator with current node  $CN$ .
- i) initialise  $W$  as 3/4 distance from previous to next node, and increase by 1 if even value.
- ii) initialise minimum sampling space  $MinS$  to 10.
- iii) set  $N$  to be twice  $local$  parameter plus 1.

- iv) set profile size **D** to 8 times **local** parameter.
  - v) set window sample space **WSS** to be **D** over **N**.
  - vi) make **MLD** structure
  - vii) run **MLD** and clean up unwanted information.
  - viii) return **MLD** structure to calling routine
2. Make **MLD** structure
- i) while **W\*WSS** less than **MinS** increment **W** by 2.
  - ii) set up memory areas for information gathering
  - iii) sample image over real coordinate profile to produce a local sampled image of size **W** by **D**.
  - iv) return **MLD** structure
3. Run **MLD** operator
- i) for each **MLD** sampling window, calculate:-
    - a) number of pixel points
    - b) sum of pixel values
    - c) sum of the squares of the pixel values
  - ii) using all these statistics calculate *sd* for **MLD** as a whole.
  - ii) step through **MLD** sampling space ( $i \leq i < N-2$ ), and
    - a) calculate  $sd_{A_i}$  over **MLD** space 0 to  $i-1$
    - b) calculate  $sd_{B_i}$  over **MLD** space  $i+1$  to  $N-1$
    - c) calculate maximum likelihood  $w_i$
  - iv)  $w_0$  and  $w_{N-1}$  are set to zero
  - v) return informatics

Back in the snake algorithm, a points position along the profile gives rise to an index into the  $w$  statistics, and this value can then be used as though it is a gradient image value.

### *Global Shift*

One change that we have adopted, is the use of a global energy correspondence in the algorithm. On entry to the active contour model algorithm, the position of the contour is shifted over a  $n*n$  discrete grid with no change in its morphology, in an attempt to find a lower energy position for a contour of the given shape. This it does by summing the internal and image energy for all nodes, and then normalising; this gives a default energy for the snake. The global shift algorithm makes no use of higher order constraints such as repulsor and attractor points. As the entire snake is shifted ( $n$  is defined using the *globals* parameter), the (image) energy is re-evaluated at this new position and the snake shifted in its location if a lower default energy is found at that location.

It should be noted that the global shift algorithm uses the gradient operator nominated by the user, unless that operator happens to be the maximum likelihood edge detector. When this is the chosen edge operator for the active contour minimisation routine, the Canny is used for the global shift as the **MLD** is too computational expensive.

### *General Behaviour Parameters*

There are a number of parameters and user defined actions that affect the behaviour of the active contour model.

The *Iterations* parameter controls the maximum number of overall contour energy minimisations that will take place on each call to the minimisation routine; giving it a value of 0 will not cause any changes to be made. If the contour has not changed after an iteration (that is, there has been no change in the position of any contour defining node) the active contour module returns

The *window* parameter controls the sample size of the profiles at each node; giving it a value of 0 will result in the contour failing to move due to local deformation, irrespective of any energy function parameters, as no alternative points for any node will be considered. A too large value may result in the contour making dramatic and erratic shapes in its morphology.

A further feature is the possibility of *fixing* a node; this causes the selection point in the contour to be made static, and it will not change position irrespective of any other change in the contour.

### *Node Cleaning*

Several algorithms have been investigated for the automatic addition and deletion of nodes. In the original algorithm, the spacing of nodes was controlled through the use of two parameters (*Minus* and *Maxus*). The user can also cause the behaviour of the contour model to change by increasing or decreasing the number of nodes involved in specifying the contour; too few nodes will result in a poor approximation to any desired contour. The active contour model testbed (and the SAMMIE Advanced Segmentation modules) allow the user to add, delete or move nodes manually through the use of the contour editing facilities.

The simplest node cleaning algorithm is that of specifying a minimum and maximum (straight line) distance between nodes (using the *Minus* and *Maxus* parameters). If the distance between a node and the following node is less than *Minus*, the node is removed. If the distance between a node and the following node is greater than *Maxus*, a node is inserted halfway between them. However, this does not allow for any subtlety in the control over the placement of nodes. While this simple spacing requirement ensured that the nodes defining the shape and location of a contour were neither too sparse or over-close, it failed to allow for nodes to be close on contour sections of high curvature, and more sparse on areas of small curvature. Total removal of this algorithm caused the contour to become erratic in its shape, with irregular placement of nodes, particularly after a number of minimisation iterations.

The second algorithm we allow for the deletion (only) of nodes (ThetaCheck in the active contour testbed) looks to see if the direction of the contour changes drastically in going from a node's predecessor to itself and then to its successor. This test is done on nodes that are spaced no more than *Maxus* apart. This is meant to try and keep a smooth overall appearance to the contour. Obviously if the nodes are very widely spaced out (ie greater than *Maxus*) than the first separation only algorithm should insert nodes, which can then be tested using this algorithm.

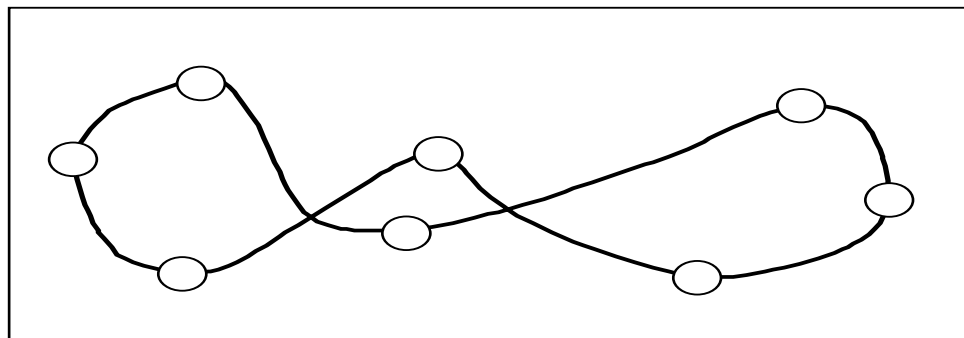


Figure 4. *The Unwanted Crossing Contour*

There is a variation on this algorithm (*Spike Remove*) that deletes closely spaced nodes if there is a drastic change in the distance from the contour centre of gravity in a neighbourhood of three successive nodes.

A further problem is that of controlling and in fact disallowing those deformations which cross themselves (see figure 4). In the current algorithm, we have to check that favouring a new position for a contour defining node does not lead to the generation of a self-crossing contour. Further work on the means of defining the nature of active contour models and controlling the displacement of nodes through energy minimisation (or convergence) may present a means whereby such deformations are impossible because of the very nature of the active contour model (for example sprung radial and SAT contours). Using the active contour model within the Brainworks advanced segmentation tool, we can clean these type of contours using digital morphological [32] techniques, but this is not a totally satisfactory solution.

### *Local Coefficient Adaption*

We have tried, with little success, to allow the various energy coefficients to vary locally in an attempt to relax the snake at the controlling nodes. Unfortunately, as the active contour model uses an energy minimisation technique, this meant that the algorithm for local coefficient adaption simply changed the coefficients to produce a local energy minima; the resultant snake morphology did not generally fit with user expectations. However, if the expected snake energy for a particular object contour, and more specifically for local areas on those contours, could be modelled, a model driven convergence rather than minimisation would be possible. This would perhaps produce a better class of contours for any particular object. Indeed, while experimenting with this algorithm the desired contour was often displayed but as energy minimisation continued the contour diverged from the expected (hoped for) contour. If a database of contours, for objects to be found in the MR and CT imaged brains (across all relevant slices), could be produced, these would provide a training data set for energy statistics and a model-driven energy convergence algorithm with local coefficient adaption may work to high degree of satisfaction.

### *Local Coefficients*

The energy coefficients used in calculating global and local energies are typically fixed to default values, or changed by the user to suit a particular contour. However what may be sensible coefficients for one area of the contour, may be poor for a further sections. For example consider the thalamus dex (a grey-matter object). The right side of this object abuts the high contrast left ventricle and is well defined; suitable for high coefficients for the image energy components. The left side is very much poorer in its definition, and ought to have much lower coefficients for the image energy component and so reduce their effect. We did try local adaption (see above) to solve this problem, but this was ineffective. As a result, we devised a method for allowing the user to specify local coefficients, and so locally override the general energy coefficients in force for the contour as a whole.

---

---

## 6. Future Work

There are a number of interesting avenues of research that we have so far been unable to spend time on, but which may prove to be beneficial for this or some other image domain. I have grouped these below under the appropriate headings, and with the exception of the work on dipoles and maximum likelihood detectors, it will be impossible to further this work from within the confines of the SAMMIE project

### *Dipoles and maximum likelihood detectors*

Maximum likelihood detectors were introduced above, and while they appear to detect edges much more reliably than the other edge operators tested, and have some improvement on the efficiency of the snake for certain objects, we have experienced problems in their use, particularly with sections of closed contours adjoining and containing high curvature contour sections. Further work is really required here.

### *Radial Snakes*

One change to the active contour models that has been thought about, is that of changing their specification for closed contours so that they are specified in terms of radial axes (much like a deformed and deformable bicycle wheel). For the sake of brevity, these shall be referred to as radial snakes (see figure 5).

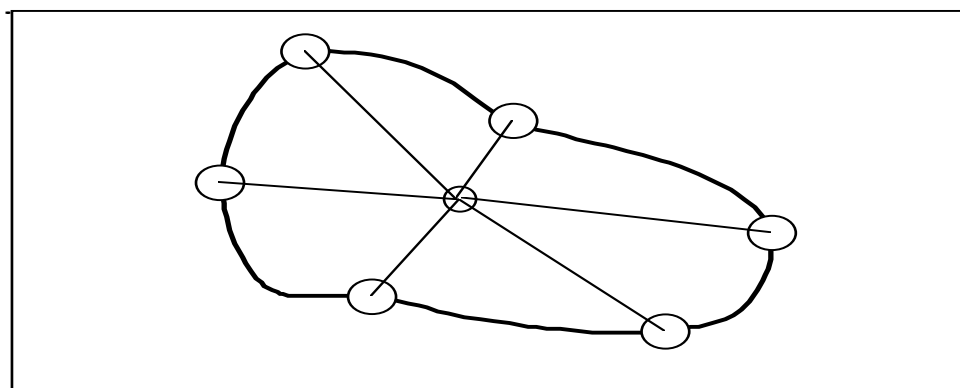


Figure 5. Radial Snake

This would fit well with the profile-based sampling space and the use of the maximum likelihood edge detector. It would allow us to specify that no contour could be deformed in such a way as to allow a zero length radial axis. It may also help in the removing or controlling the generation of noisy deformed contours as discussed in the section on node cleaning above. It would not, however, stop the generation of self-crossing contours, and may actually cause problems where elongated and highly convoluted contours (particularly those demonstrating both convexity and concavity) were required.

### *Sprung SATs*

Another thought, that arose in experiments where the snake collapsed upon itself (or even crossing itself), was that of modelling the contour using some medial axis or skeletonised axial transform, with perpendiculars linked from the SAT to contour controlling nodes using springs. The springs could be compressed or expanded, and so altering their sprung and elastic energies. This would require a more sophisticated energy model than the one presently used. The control of contour deformation using such a model would be appropriate for elongated and convoluted contours displaying high degrees of concavity and convexity (see figure 6); such as the objects delineated at the surface of the brain (for example *gyrus frontalis medialis*).

We have approximated such a model, through using a medial line of repulsor points; this does require some experimentation with the placement and number of repulsor points and the value of the repulsor coefficient.

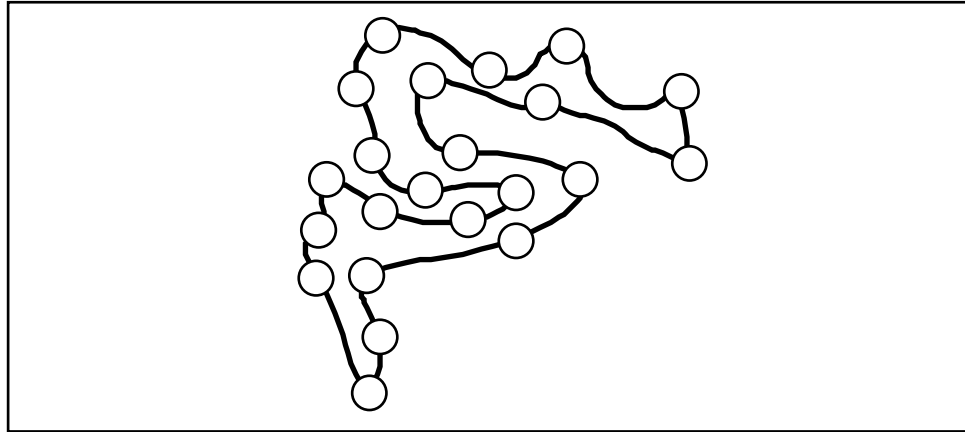


Figure 6. Highly convoluted contour with concavities and convexities.

### *Deformable models and atlases*

The Brainworks suite produced by the SAMMIE project contains two and three-dimensional atlases of the brain. We have run experiments in mapping the 2D atlas to particular data sets and then using these contours as seeds to semi-automatic object delineation methods, including the active contour model. There is much more work to be done here, particularly if a semi-automated system is to be used within the Brainworks suite.

### *Model-Driven Snakes*

Related to the last section on advances to the basic model, is area of research touched on in the section on local coefficient adaption. If we were to model the active contour model energy and parameters for contours fitted to objects in particular slices, the active contour model algorithm could be driven using these parameters until convergence on the modelled energy; which may not necessarily be a minimal energy.

### *Three Dimensional Snakes*

One final area of research that may be particularly suitable for MR and CT imaged brains is that of three dimensional snakes. If we consider a number of successive data slices, containing different views of the same object, we can place a series of contours in each of these two dimensional slices. The deformation of the contours may then be controlled through a set of contours that are placed in the Z dimension, connecting nearest Z axis neighbours in successive slices. We then have a series of bubbles connected vertically through a further set of bubbles (see figure 7).

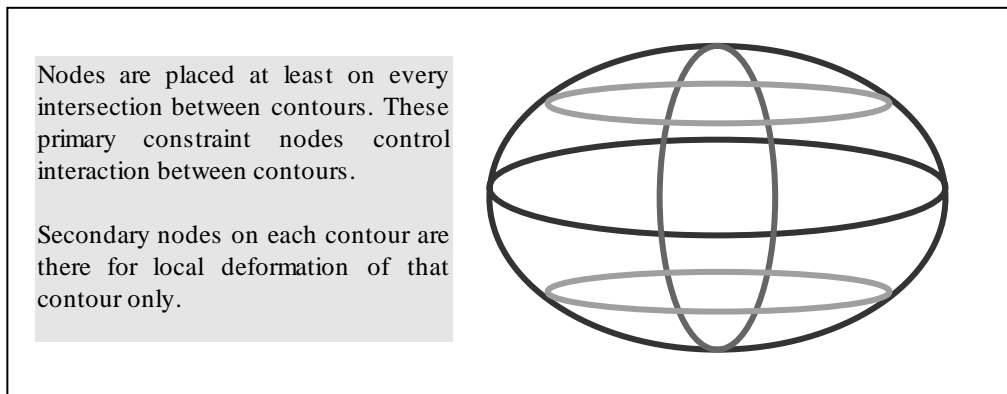


Figure 7. A Simple Three-Dimensional Snake

## 7. Conclusion

It should be stated that although the active contour model will deform to provide good matches to the required object; providing a very poor approximation to the sought object will result in a poor contour model being produced. This is particularly so where the outline includes disparate high contrast features. Even so, the active contour model may be improved through the use of the disparate features developed for the SAMMIE software, such as repulse and attractor nodes.

The active contour model is still essentially stupid; and may work with greater efficacy if higher order constraints (even in a semi-automated system) were made available to control its seeding and nature of deformation. Such constraints are present in the 2D, 3D and statistical Brainworks atlases. However, the nature of the SAMMIE project has not really allowed us the time to research and develop these sufficiently for them to be of any real practical use. The type of knowledge-based system outlined in [10] would be useful, when used in conjunction with these atlases and the active contour model, to provide intelligent object demarcation aides for clinicians. An alternative is to use a (stand-alone) knowledge based scheduler that organises the required components from the Brainworks environment (e.g. symDAB, 2D atlas, stattool and Hand/Advanced Segmentation Tool).

---

## 8. References

- [17] Anthony, D., Hines, E., Taylor, D. and Barham, J. (1989) An Investigation Into The Use Of Neural Networks For An Expert System In Nuclear Medicine Image Analysis. IEE 3rd International Conference On Image Processing and its Applications, July 1989, pp 338-342.
- [14] Banks, G., Vries, J.K. and McLinden, S. (1987) Radiologic Automated Diagnosis (RAD). Computer Methods and Progress in Biomedicine, 25, pp 157-168.
- [9] Brummer, M.E., Mersereau, R.M., Eisner, R.L. and Lewine, R.R.J. (1993) Automatic detection of brain contours in MRI data sets. IEEE Trans. Medical Imaging, 12(2), pp 153-166.
- [26] Canny, J.F. (1983) Finding edges and lines in images. MIT Technical report AI-TR-720.
- [14] Chen, S-Y., Lin, W-C. and Chen, C-T. (1989) An expert vision system for medical image segmenatation. SPIE 1092: Medical Imaging III: Image Processing, pp 162-172.
- [23] Cootes, T.F. and Taylor, C.J. (1992) Active Shape Models - 'Smart Snakes', Procs. BMVC 1992, Leeds, pp 266-275.
- [1] Davis, D.N., Natarajan, K. and Claridge, E. (1994) AIM Project A2032, Advanced Segmentation, (Deliverable 31).
- [2] Davis, D.N., Natarajan, K. and Claridge, E. (1994) AIM Project A2032, User Guide To Advanced Segmentation, (Companion to Deliverable 31).
- [13] Dhawan, A.P., Baxi, H. and Ranganath, M.V. (1988). Knowledge-based low-level image analysis for computer vision systems. SPIE 937. Applications of Artificial Intelligence VI, pp 2-9.
- [7] Gerhardt, P. and Frommhold, W. (1988). Atlas of Anatomic Correlations in CT and MRI. Thieme Medical Publishers Inc., New York.
- [30] Graham, J. and Taylor, C.J. (1988). Boundary Cue Operators for Model Based Image Processing. Proceedings of the 4th Alvey Vision Conference, University Of Manchester, September 1988, pp. 59-64.
- [4] Hounsfield, G.N. (1973). Computerized transverse axial scanning (tomography): Part 1. Description of system. British Journal Of Radiology, 46, 1016-1022, 1973.
- [22] Karaolini, P., Sullivan, G.D. and Baker, K.D. (1992) Active Contours using Finite Elements to Control Local Scale., Procs. BMVC 1992, Leeds, pp 481-487.
- [3] Kass, M., Witkin, A. and Terzopoulos. (1987) Snakes: Active contour models, International Journal of Computer Vision, 1(4), pp 133-144.
- [32] Kong, T.Y. and Rosenfeld, A. (1989) Digital Topology: Introduction and survey. CVGIP 48, pp 357-393.

- [8] Kretschmann, H.-J., and Weinrich, W. (1986). *Neuroanatomy and Cranial Computed Tomography*. Thieme Medical Publishers Inc., New York.
- [29] Lee, J.S.J., Haralick, R.M. and Shapiro, L.G. Morphologic Edge Detection, (1986) IEEE, pp 369-373.
- [11] Li, C., Goldgof, D.B. and Hall, L.O. (1993) Knowledge-based classification and tissue labeling of MR images of human brain. *IEEE Trans. Medical Imaging*, 12(4), pp 740-750.
- [5] Mansfield, P. and Morris, P.G. (1982). *NMR imaging in Biomedicine*, 155-201, Academic Press, New York, 1982.
- [24] Marr, D. and Hildreth, E. (1980) Theory of Edge Detection. *Proceedings of the Royal Society, B* 207, pp. 187-217.
- [27] Mero, L. and Vassey, Z. (1975) A simplified and fast version of the Heuckel operator for finding optimal edges in pictures. *Proceedings 4th ICAI, Georgia, USSR*, pp 650-655.
- [10] Natarajan, K. (1993) *A Knowledge Based System for Radiological Image Recognition*. PhD Thesis, Computer Science, University Of Birmingham.
- [31] Pycock, D., Ping, Z. The maximum Likelihood Detector. *BMV Medical Vision Seminar*, Nov 1994.
- [17] Robb, R.A., Heffernan, P.B., Camp, J.J. and Hanson, D.P. (1987) A workstation for multi-dimensional display and analysis of biomedical images. *Computer Methods and Programs in Biomedicine* 25, pp 169-184.
- [21] Sonka, M., Hlavac, V. and Boyle, R. (1993) *Image Processing, Analysis and Machine Vision*. Chapman and Hall.
- [19] Terzopoulos, D., Witkin, A. and Kass, M. (1987) Symmetry-seeking models for 3D object reconstruction. *Proceedings of First International Conference on Computer Vision, London*, pp 269-276.
- [16] Tomita, F. (1988) Interactive and automatic image recognition system. *Machine Vision and Applications* 1, pp 59-69.
- [12] Vernazza, G.L., Serpico, S.B. and Dellepiane, S.G. (1987) A knowledge-based system for biomedical image processing and recognition. *IEEE Trans. Circuits and Systems, CAS-34*(11), pp. 1399-1416.
- [24] Williams, D.J. and Shah, M. (1992) A Fast Algorithm for Active Contours and Curvature Estimation. *CVGIP*, 55 (1), pp 14-26.
- [20] Witkin, A., Terzopoulos, D. and Kass, M. (1987) Signal matching through scale space. *International Journal of Computer Vision*, 1(2), pp 133-144.
- [6] Young, S.W. (1988). *Magnetic Resonance Imaging: Basic Principles (Second Edition)*. Raven Press, New York.

## Appendix A. Using The Snake Testbed

The active contour model is available as part of the Advanced Segmentation Tool module within the Hand segmentation Tool, and also in its own testbed. Both these tools can be found in the vision section on the World Wide Web pages for the Department Of Computer Science pages (in directories *HandSeg* and *Snake* respectively). The Advanced Segmentation Tool has its own technical and user manuals [1,2]. The following presents details on how to use the *Snake* testbed.

Moving into the Snake WWW directory, the commands *source configure* and *Snake* (assuming environmental variables for DISPLAY etc are correctly set) will result in the four windows (shown in figures 8, 9, 10, 11) being displayed. If an image is given as an in-line parameter only three windows are displayed; the section on in-line parameters gives further details on how to do this. The following paragraphs detail the functionality associated with each of these windows.

### *Main Control Panel*



Figure 8. Top Control Panel For The Active Contour Model Testbed

The main control frame (titled Active Contour Model - see figure 8) contains nine buttons. These are:-

#### *Quit*

Pressing this button causes a pop up, asking for confirmation of the intention to quit the tool, to be displayed. The user can either cancel at this point, causing no effect on the tool, or to quit, closing down the tool.

#### *File*

This is used in combination with an in-line parameter specifying a file containing a snake contour. If such a parameter has been given, pressing this button results in the contour being displayed in the image display window.

#### *Bubble*

This button causes the drawn contour displayed over the image (as in figure 11) to be passed to the active contour model, with parameters set as defined in the parameter selection panel. It is meant to be used with closed contours.

#### *Snake*

This button causes the drawn contour displayed over the image (as in figure 11) to be passed to the active contour model, with parameters set as defined in the parameter selection panel. It is meant to be used with open (string-like) contours; as these are not currently implemented it actually works in the same manner as the *Bubble* button.

### *Reject Snaked Outline*

On returning from the active contour model, the original contour (with nodes represented as crosses) and a further contour (with no nodes explicitly displayed) are shown. The second of these contours is the result, and can be rejected by pressing this button.

### *Accept Snaked Outline*

On returning from the active contour model, the original contour (with nodes represented as crosses) and a further contour (with no nodes explicitly displayed) are shown. The second of these contours is the result, and can be accepted by pressing this button. It then becomes the seed contour which may be edited or used in further active contour experiments.

### *Disp edge*

Pressing this button causes the current edge (as produced by the Sobel or whatever edge operator the user has chosen) map to be displayed in the image display window. There is no edge image associated with the use of the dipole however. To observe the effect of using the dipole, the user must go to the window labelled *Outline* (see below).

### *Disp zero*

Pressing this button causes the zero crossing map to be displayed in the image display window. For most images, the canny approximation to the zero crossings is used, and in these cases, the Canny output image is displayed.

### *Disp image*

Pressing this button causes the source image to be displayed in the image display window; this is typically used after displaying one of the edge or zero crossing maps in order to revert to the default display of the source image.

## *Image Selection Panel*

---

**Image name :**

<pre>/thalamus/AIM/dnd/Snake/Images/ceph1 /thalamus/AIM/dnd/Snake/Images/QE_MR.7 /thalamus/AIM/dnd/Snake/Images/QE_MR.1 /thalamus/AIM/dnd/Snake/Images/QE_CT.10</pre>
---

**Dir :** /thalamus/AIM/dnd/Snake/Images

**Filename :** QE\_MR.1

**CT\_ZERO** 976

**CT\_RANGE** 127

Figure 9. Image Selection Panel For The Active Contour Model Testbed

The image selection panel (with the frame label *Images Available*) is displayed even if an image is given as a in-line parameter to the Snake program; unless the in-line parameter *-noisp* is given. The image selection panel (see figure 9) contains one column of selectable text, a text field detailing the image directory, a text field detailing the name of the currently selected image and two numeric fields for controlling the windowing parameters for CT images, and a one function button.

The scrollable text contains the names of the default images available for use in the active contour testbed; typically a selection of MR, CT and lateral cephalogram images. Any image name given as an in-line parameter is also placed in this list. If a large number of images are available this window becomes scrollable. Clicking on any row, results in that image being displayed in the image display window.

The text immediately below this scrollable list gives the name of the default image directory. To define an alternative image directory must be done before starting the *Snake* program. There are two ways of specifying an alternative image directory: either the configure file must be edited and the pathname associated with the variable *ACM\_IMAGE\_DIR* changed; or after running the *source configure* command, use the *setenv* command to associate a different pathname with this environmental variable. The latter option is the one to do if using the *Snake* program from the WWW directory. The next field contains the name of the currently loaded image.

The two numeric text fields are used to control the windowing of CT images; the windowing of MR images is handled automatically. The default parameters are usually sufficient for grey matter object delineation.

However, for certain structures and for images sourced from differing imaging apparatus other values may be required. The *CT\_ZERO* field defines the windowing zero value, while the *CT\_RANGE* defines the range of pixel values that are to be fitted between black and white. The return key must be pressed for these values to be accepted. The function button (*CT\_load*) causes the CT images present in the scrollable window (plus any further image files loaded) to be loaded using these changed windowing parameters.

In all cases, the loading of an image file replaces an image of exactly the same full path specification in the list of available images.

## *Snake Parameter and Contour Control Panel*

The snake parameter and contour control panel (see figure 10) contains three sets of controls. The first set (of 12 function buttons) are used in defining and editing contours. The next set contains 13 numeric text fields and a mutually exclusive selection list that define parameters of use in the active contour model. The final set consists of four function buttons. These are detailed below under headings related to the text associated with the button or numeric text field.

It will be noticed that some fields and buttons are only functional if a contour has been drawn; whilst others are only functional if no seed contour currently exists. Many of the buttons do not function until an image has been loaded (see above). Like most of the snake testbed, these buttons are mouse driven. While the following descriptions describe the button functionality through the use of the left mouse button, it is possible in some drawing function modes to use the middle button. As different function buttons are selected, one or two text fields are placed on the bottom of the image display window frame. These detail the current functionality of the left and right mouse buttons.

### *Auto ROI*

Pressing this button allows the user to generate a regular polygon as a ROI (or contour). The user can control the size of this by moving the mouse cursor to a desired location and rubber-banding. Rubber-banding is achieved by pressing the left mouse button and, while keeping this button depressed, moving the mouse cursor until a suitably sized polygon edge is achieved. Releasing the left button then causes the polygon to be drawn. The ROI can be moved (using the *Move ROI* button) to a better position if required.

### *Free hand ROI*

This button allows the user to draw a free hand curve (or ROI) anywhere within the displayed image. After selecting this option, the user can move the mouse cursor to the required contour start place and commence with a free hand drawing of the contour. The drawing is started by depressing the left hand mouse button and by dragging the mouse cursor while keeping the left hand mouse button depressed, the contour grows with every cursor movement. If the drawing mode is in bubble (closed contour) mode, the user should aim to nearly close the contour; releasing the left button causes the contour to be completely closed. At this point a contour for editing or seeding the active contour model is available.

<b>Number of nodes</b>	12	/	5
<b>Auto ROI</b> )	<b>Free hand ROI</b> )	<b>Clear ROI</b> )	
<b>Move ROI</b> )	<b>Move nodes</b> )	<b>Clear node</b> )	
<b>Fix Static</b> )	<b>Add Attract</b> )	<b>Add Repulse</b> )	
<b>Remove Attract</b> )	<b>Remove Repulse</b> )	<b>Locals</b> )	
<b>alpha</b>	1.00		
<b>beta</b>	1.00		
<b>gama_grad</b>	0.75		
<b>gama_zeroX</b>	0.25		
<b>zeroX Off</b>	-0.10		
<b>zeroX On</b>	0.15		
<b>Attract Coefficient</b>	0.15		
<b>Repulse Coefficient</b>	0.15		
<b>Maxus</b>	15		
<b>Minus</b>	6		
<b>Window</b>	3		
<b>Global</b>	5		
<b>Iterations</b>	10		
<b>Edge:</b>	Mero	Sobel	LHS
	Canny	Dipole	
<b>Break Off</b> )	<b>Dipole</b> )	<b>Energy</b> )	<b>Node Check</b> )

Figure 10. Parameter Panel For The Active Contour Model Testbed

If the user requires, it may be moved to a more suitable position, have its node population reduced or make use of any of the functions associated with the testbed.

*Clear ROI*

Clicking on this function button causes a pop up to be displayed asking for confirmation on the deletion of the currently drawn contour. The user can accept this deletion or cancel; the latter choice causes no change.



Figure 11. Image Display Panel For The Active Contour Model Testbed

#### *Move ROI*

Clicking on this function button allows the user to move the currently drawn contour. The user must move the mouse cursor near the curve and depress the left hand mouse button. The contour may then be dragged to a more suitable position; letting go of the left mouse button causes this movement to be accepted.

#### *Move nodes*

Clicking on this function button allows the user to move contour controlling nodes. By placing the mouse cursor near the contour section to be changed, a press and drag operation using the left mouse button causes the node nearest to mouse cursor to be moved to the shown position.

#### *Clear node*

Clicking on this function button allows the user to delete contour controlling nodes. By placing the mouse cursor near the contour section to be changed, a press on the left mouse button causes the node nearest the mouse cursor to be deleted. If the contour is closed, it remains so but with one fewer node.

#### *Fix static*

Clicking on this function button allows the user to select control nodes that are not allowed to move. By placing the mouse cursor near the contour section to be changed, a press on the left mouse button causes the node nearest the mouse cursor to be labelled as *fixed*. This has the effect of over-riding the global shift algorithm and prohibiting the selected node to undergo any local displacement in the active contour model.

#### *Add Attract*

Clicking on this function button allows the user to add attractor nodes and make use of these in the energy minimisation routine. Attractor nodes are placed at the current mouse cursor position by clicking the left mouse button. There is no limit to the number of attractor nodes that may be placed on an image. In the current implementation, attractor nodes affect all contour defining nodes, but in inverse relation to their distance from any attractor nodes. As attractor nodes are added, all current attractor nodes are displayed.

### *Add Repulse*

Clicking on this function button allows the user to add repulsor nodes and make use of these in the energy minimisation routine. Repulsor nodes are placed at the current mouse cursor position by clicking the left mouse button. There is no limit to the number of repulsor nodes that may be placed on an image. In the current implementation, repulsor nodes affect all contour defining nodes, but in inverse relation to their distance from any repulsor nodes. As repulsor nodes are added, all current repulsor nodes are displayed.

### *Remove Attract*

Clicking on this function button allows the user to remove attractor nodes. Clicking the left mouse button causes the attractor node nearest the cursor to be deleted. All remaining attractor nodes are then displayed.

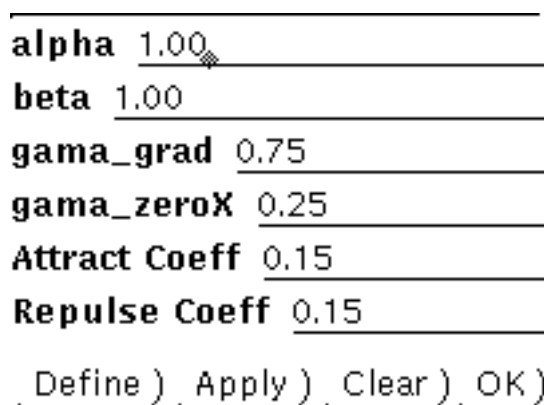
### *Remove Repulse*

Clicking on this function button allows the user to remove repulsor nodes. Clicking the left mouse button causes the repulsor node nearest the cursor to be deleted. All remaining repulsor nodes are then displayed.

### *Locals*

Clicking on this function button allows the user to specify differing energy coefficients for differing contour sections. A small menu containing a 6 numeric text fields and 4 function buttons is displayed (see figure 12) if a contour has been drawn on the image.

The numeric text fields mirror the functionality of the 6 energy coefficient text fields in the Snake Parameter and Contour Control Panel. As the frame is displayed, these take the default global values; the user can then change these to suit his intentions and they will override the global energy coefficients for that section of the contour they apply to.



The image shows a dialog box titled "Local Coefficient Parameter Panel". It contains six text input fields, each with a label and a value: "alpha 1.00", "beta 1.00", "gama\_grad 0.75", "gama\_zeroX 0.25", "Attract Coeff 0.15", and "Repulse Coeff 0.15". Below these fields are four buttons: "Define", "Apply", "Clear", and "OK". Each button has a single underlined letter as a mnemonic.

Figure 12. Local Coefficient Parameter Panel

By clicking the *Define* button the user can define a rectangular region (through rubber-banding with the mouse left button). Those sections of the contour (or to be more exact, those contour controlling nodes) within this rectangle will be affected. Once the region of interest rectangle has been defined and the energy coefficients changed to non-default values, pressing the *Apply* button causes the contour data structure to be updated with these local coefficients. There is no limit to the number of ROIs that may be placed over a contour, but the user must be aware that the last apply overrides any earlier ROI that may have been applied to contour controlling

nodes. The *Clear* button is used to clear all rectangular regions of interest and any local coefficients affecting contour controlling nodes. The OK button simply closes down the Local Coefficient Parameter Panel, without affecting any local coefficients.

#### *alpha*

This numeric field text allows the user to specify non-default real number values for the *alpha* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *beta*

This numeric field text allows the user to specify non-default real number values for the *beta* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *gama\_grad*

This numeric field text allows the user to specify non-default real number values for the *gama\_grad* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *gama\_zeroX*

This numeric field text allows the user to specify non-default real number values for the *gama\_zeroX* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *zeroX Off*

This numeric field text allows the user to specify non-default real number values for the *edge\_off\_weight* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *zeroX On*

This numeric field text allows the user to specify non-default real number values for the *edge\_on\_weight* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *Attract Coefficient*

This numeric field text allows the user to specify non-default real number values for the *attract* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *Repulse Coefficient*

This numeric field text allows the user to specify non-default real number values for the *repulse* coefficient as used in the energy minimisation routine. It coerces integer values to real (double) values, and requires the user to press the return key after changing the displayed value.

#### *Maxus*

This numeric field text allows the user to specify non-default integer number values for the *maximum node separation value* as used in the node cleaning routines. It only accepts integer values, and requires the user to press the return key after changing the displayed value.

#### *Minus*

This numeric field text allows the user to specify non-default integer number values for the *minimum node separation value* as used in the node cleaning routines. It only accepts integer values, and requires the user to press the return key after changing the displayed value.

#### *Window*

This numeric field text allows the user to specify non-default integer number values for the *windows value* as used in the local contour deformation routine. It only accepts integer values, and requires the user to press the return key after changing the displayed value.

#### *Global*

This numeric field text allows the user to specify non-default integer number values for the *globals value* as used in the global shift routine. It only accepts integer values, and requires the user to press the return key after changing the displayed value.

#### *Iterations*

This numeric field text allows the user to specify non-default integer number values for the *iterations value* as used in the energy minimisation routine. It only accepts integer values, and requires the user to press the return key after changing the displayed value.

#### *Edge*

This exclusive choice menu allows the user to specify which gradient edge operator is required in the active contour model. There are currently five options: Mero-Vassey; Sobel; the bi-directional morphological edge operator (LHS); the Canny; and the maximum likelihood edge detector (MLD). The default operator is set as the Sobel, but by clicking on one of the other four options that is preferred as the operator to produce the gradient edge map.

#### *Break*

Clicking on this function button ensures that the minimisation routine breaks (stops) after each iteration, and displays the histogram of the energies around all of the contour defining nodes. The user must hit the return key in the shell window running the Testbed for the next iteration to commence.

#### *Dipole*

---

Define) Apply) Clear)  
Display) OK)

**Dipole W:** 5

**Dipole Len:** 20

**Min Sample:** 9

Figure 14. Stand-alone MLD Parameter Panel

Clicking on this function button allows the MLD operator to be used in isolation. A small pop-up window containing a number of 5 function buttons and 3 numeric text fields is displayed (see figure 14).

The *Define* button is used to define the start and end points of the profile. The user after clicking on this function button, moves the mouse cursor to the image window, and selects the start point by depressing the left mouse button. The user can then rubber-band a line and define the end point by releasing the left mouse button.

Clicking the *Apply* button activates and runs the MLD using the displayed profile and parameters defined by the three numeric text fields. The *Clear* button is used to free all data associated with the MLD, including the currently defined profile. The *Display* button is used to display the results of using the MLD operator; it displays the profile line, the point most likely to be an edge and the informatics associated with the MLD operator. The *OK* button causes the MLD parameter frame to disappear. The numeric fields are used to vary three of the MLD parameters; the sampling width, the MLD length (N) and the minimum sampling space for one MLD window.

#### *Energy*

Clicking on this function button causes the histogram of the energies around all of the contour defining nodes to be displayed; this only works on contours that have passed through the active contour model and been accepted.

#### *Nodes*

Clicking on this function button allows the user to select which, if any, node checking algorithm is to be used. A small pop-up window containing a number of function buttons with changeable text fields is displayed (see figure 15). The text field states whether or not a particular algorithm is being used.

The Simple and Augmented Check use the same algorithm; the basic node checking algorithm discussed in section 5. The difference is that Simple checking only happens on entry to the energy minimisation routine while the Augmented check occurs on every iteration of the williams-shah algorithm. The theta node and spike remove checks are as detailed in section 5. The minimum separation check ensures that no node encroaches upon any other.

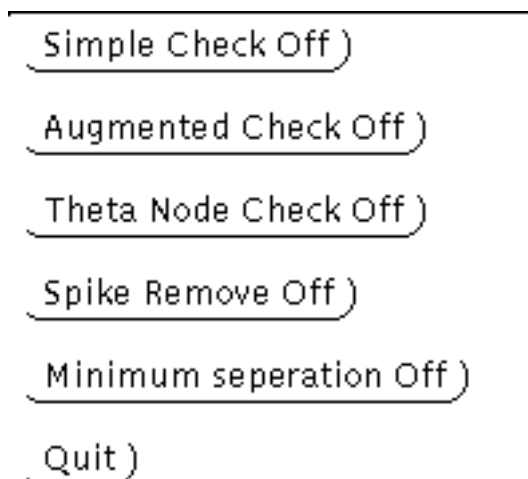


Figure 15. Node Checking Selection Panel

## *General Notes On Using The Testbed*

Any existing contour can be used to initialise the active contour model; the better the seed, the better the result. In any case, this initial contour provides the starting place for the active contour model, which should deform the contour to fit the loosely outlined object.

### *In-Line Parameters*

The active contour model testbed need not be given any parameters; the user is then limited to using images in the directory defined by the environmental variable *ACM\_IMAGE\_DIR*. There are, however, a number of parameters that the system can accept, eg

*Snake -F Snake\_file*

*Snake -I /tmp/My\_Image*

*Snake -w 256 -h 256 -I /tmp/Small\_Image*

*Snake -I /tmp/MRI\_Image -T MR*

There is no set order in which any of these five parameter are given, but the user must be aware of the following.

-F

The next parameter gives either the relative (to *ACM\_HOME*) directory or the absolute pathname of a file containing a snake descriptor to be loaded using the *File* function button.

-I

The next parameter gives either the relative (to *ACM\_HOME*) directory or the absolute pathname of an image file. Images loaded via in-line parameters are assumed to be 512 by 512 8bit data files; unless CT or MR images from the QE which are assumed to be 256 square short integer from the known scanners. Where this is not the case the user must specify one of the following:

-w

The next parameter must be an integer specifying the image width.

-h

The next parameter must be an integer specifying the image height.

-T

The next parameter must be one of the following patterns, which specify what type of Image the -I parameter specifies.

- A      Advanced Systems (Wolfson Image Analysis Unit) Manchester
- M      RSRE-Malvern Image
- B      Straight data stored as bytes (default anyway)
- O      Rasterops Image capture card type

R Sun raster image  
S Straight data but stored as short integer  
CT Queen Elizabeth Medical Centre CT image (advantage scanner)  
MR Queen Elizabeth Medical Centre MR image (seimens scanner)

---

## Appendix B. Using The Snake In The Advanced Segmentation Tool

The active contour model is available as part of the Advanced Segmentation Tool module within the Hand segmentation Tool, and also in its own test bed. Both these tools can be found in the vision section on the World Wide Web pages for the Department Of Computer Science pages (in directories *HandSeg* and *Snake* respectively). The Advanced Segmentation Tool has its own technical and user manuals [1,2]. The following presents details on how to use the latest version of this tool.

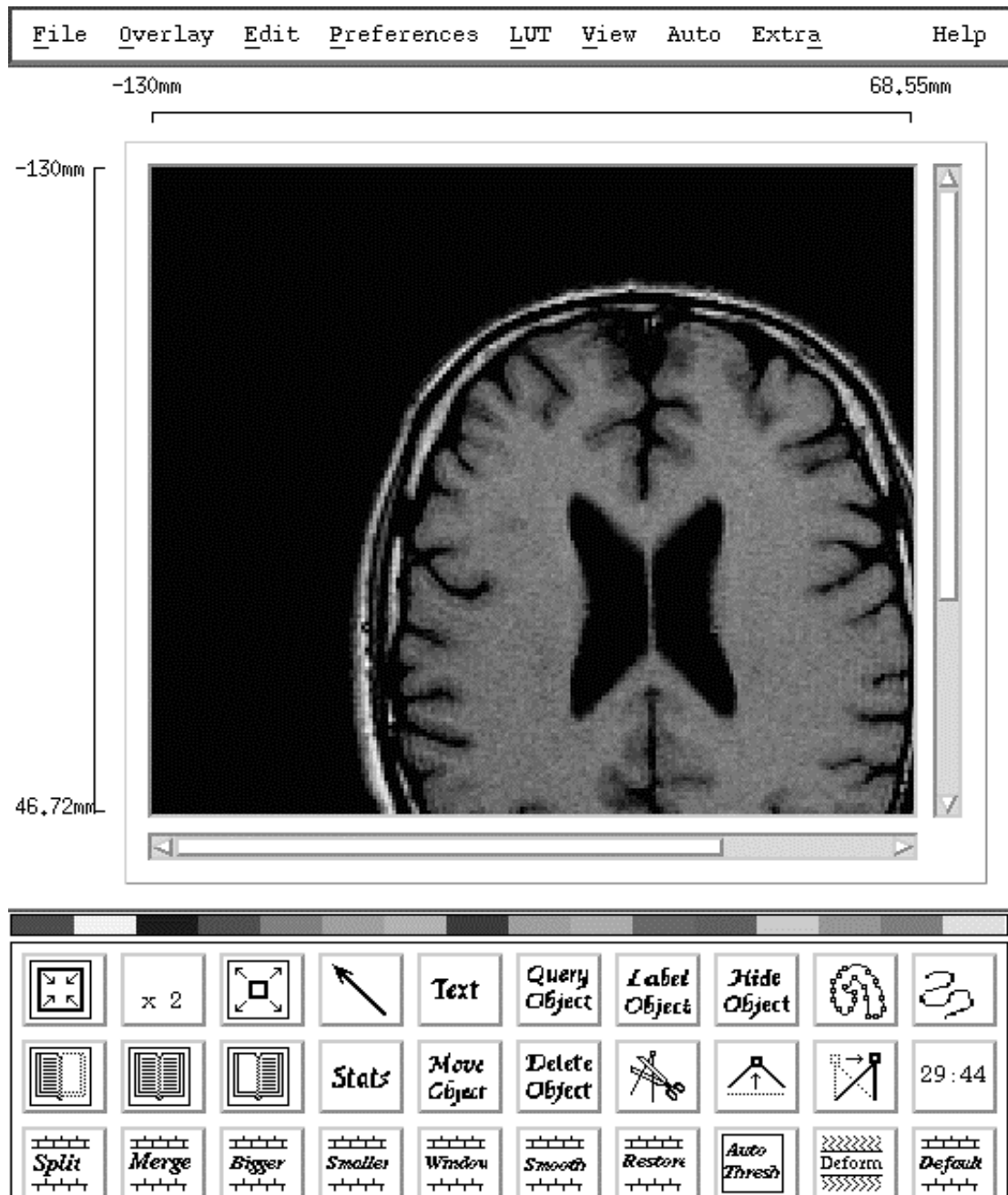


Figure 16. The Hand/Advanced Segmentation Tool Displayed With MR Slice Of The Brain

After starting the Hand/Advanced Segmentation Tool and selected an image data set (see Fig 16), the user can activate the active contour model by selecting the *AutoSnake* option to be found on the pull down menu associated with the *Auto Menu* label (seventh option on the top row menu bar). This causes the Semi-Automatic Methods Pop up to be displayed in Active Contour Model mode (see Fig. 17). The pop up consists of 6 buttons

(*Select Seed, Deform It, Repeat Test, Do Every Outline, Use New Snake, Accept Snake*), 5 panels (*Internal Influences, Image Influences, Constraint Influences, Regional Influences, Get Seed From Image*) and two function buttons at the base of the pop up (*Close, Help*). Associated with each panel are Show and Hide Toggle Buttons; in its default display all these panels are hidden (and the snake is set up to use default parameters). By pressing the appropriate Show toggle button the user can access the parameter control panels for the various function panels. Fig 18 shows all the parameter control panels open.

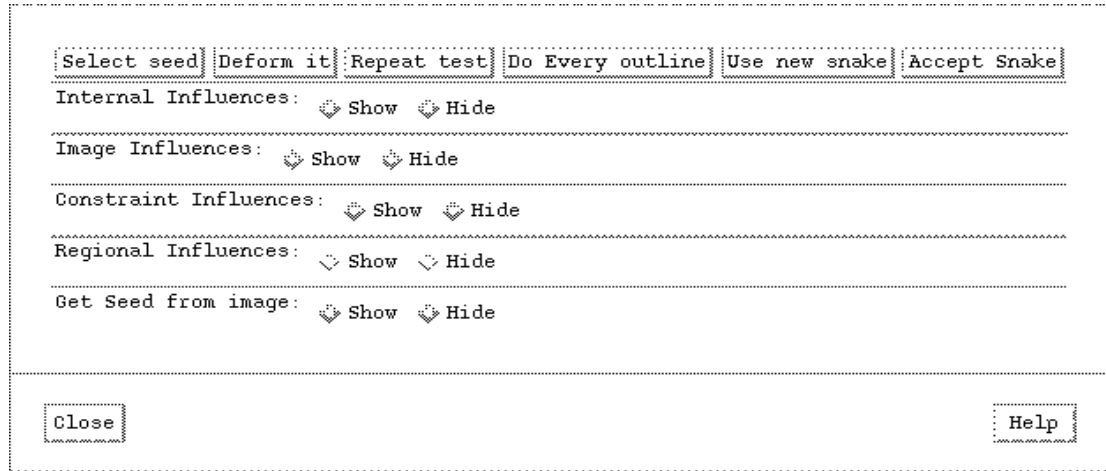


Figure 17. The Active Contour Model Pop Up In Default Compacted Mode.

There now follows a brief description of the function of each of these.

### *Select Seed*

Given that the displayed image has some form of annotation (closed contours) selecting this button allows the user to select desired seed contours for the active contour model. Once a contour has been selected, it flashes briefly in alternate colours. The seed contour may be from any possible source (hand drawn, database, other contour defining method).

### *Deform It*

The active contour model is called with the currently selected seed contour and snake parameters. The user can see the deformation effect during the use of the active contour model; the resultant snake is displayed and can be accepted, used as a further seed contour or rejected (by doing nothing with it). The window the Hand Segmentation Tool called from has a variety of messages printed in it. If no change occurs, a pop up informing the user is displayed.

### *Repeat Test*

This acts like Deform It except that it systematically steps through a range of possibly important snake parameters. This is meant to be used as an aid to finding non-default parameters that suit a particular object in a specific image modality; the developers stress that this button should be used in the normal use of the Advanced Segmentation Tool.

### *Do Every Outline*

This acts like Repeat Test except that it systematically steps through the known set of seed contours drawn in the image of possibly important snake parameters. This is meant to be used as an aid to finding non-default parameters that suit a particular object in a specific image modality; the developers stress that this button should be used in the normal use of the Advanced Segmentation Tool.

### *Use New Snake*

Use resultant contour from Deform It as seed to Deform it without accepting it.

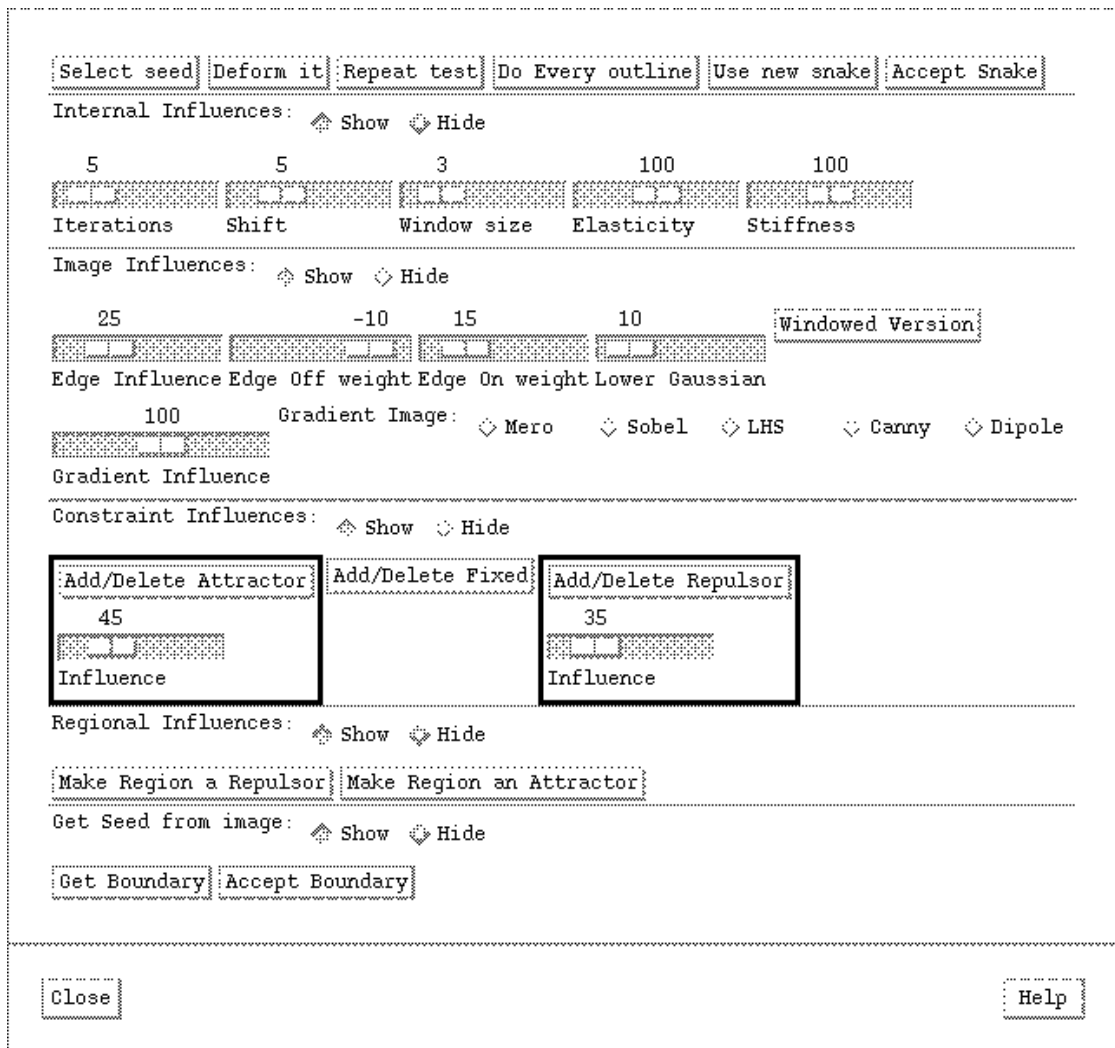


Figure 18. Active Contour Model Control Panel In Full Expanded Mode.

### *Accept Snake*

This causes contour resulting from Deform It or any other call to ACTIVE CONTOUR MODEL to be accepted in the dynamic memory set.

### *Internal Influences*

There are five parameters, controlled by sliders, in this panel which affect the internal energy of the snake.

#### *Iterations*

This parameter controls the (maximum) number of iterations to the energy minimisation function on any call to the ACTIVE CONTOUR MODEL. It takes a value in the range 0 to 50, zero ensures that no change will occur.

#### *Shift*

This parameter controls the size of the square window used in the global shift algorithm. It takes a value in the range 0 to 20; a zero value effectively switches off the global shift algorithm.

#### *Window Size*

This parameter controls the size of the local profiles used in searching for alternative positions for a contour defining control node. It takes a value in the range 0 to 25; a zero value effectively switches off the local deformation algorithm.

### *Elasticity*

This parameter controls the value of the alpha (elasticity) coefficient. It takes a value in the range 0 to 200; the maximum value of 200 actually maps into a coefficient value of 1.0.

### *Stiffness*

This parameter controls the value of the beta (stiffness) coefficient. It takes a value in the range 0 to 200; the maximum value of 200 actually maps into a coefficient value of 1.0.

### *Image Influences*

There are six parameters, controlled by sliders, in this panel which affect the image energy of the snake.

#### *Edge Influence*

This parameter controls the value of the gamma2 (zero crossing) coefficient. It takes a value in the range 0 to 100; the maximum value of 100 actually maps into a coefficient value of 1.0. A value of zero effectively switches off the zero crossing aspect of the snake.

#### *Edge Off*

This parameter controls the value of a non-edge (zero) pixel in the zero crossing edge map. It takes a value in the range -100 to 0; the -100 actually maps onto a value of -1.0.

#### *Edge On*

This parameter controls the value of a edge (non-zero) pixel in the zero crossing edge map. It takes a value in the range 0 to 100; the 100 actually maps onto a value of 1.0.

#### *Lower Gaussian*

This parameter controls the value of the lower (and hence upper) gaussians used in producing a zero crossings edge map; it takes values in the range 0 to 200. A value of 200 maps into a lower gaussian of 10.0; a zero value effectively switches off the zero crossing aspect of the snake.

#### *Gradient Influence*

This parameter controls the value of the gamma1 (gradient edge map) coefficient. It takes a value in the range 0 to 100; the maximum value of 100 actually maps into a coefficient value of 2.0. A value of zero effectively switches off the gradient edge map aspect of the snake.

#### *Gradient Image*

There is a choice of five edge operators (*Mero-Vassey*, *Sobel*, *Morphological Gradient - LHS*, *Canny* and *Maximum Likelihood Detector - Dipole*) from which to produce a gradient edge map; the default choice is the Sobel operator. Choosing the MLD (Dipole) operator means that the Canny edge operator is used in the global shift algorithm (if used).

### *Constraint Influences*

There are three functions, controlled by buttons and sliders, in this panel which affect the constraint energy of the snake.

#### *Add/Delete Attractor*

Clicking on this button allows the user to add or delete attractor nodes. Using the left mouse button causes attractor nodes to added at the current mouse cursor position; the right mouse button causes the attractor nearest to mouse cursor position to be removed.

#### *Influence*

This parameter controls the value of attractor coefficient (CoAttract). It takes a value in the range 0 to 200; the 200 actually maps onto a value of 2.0.

### *Add/Delete Fixed*

Clicking on this button allows the user to add or delete fixed nodes. Using the left mouse button causes contour controlling node nearest the current mouse cursor position to become fixed; the right mouse button causes the fixed node nearest to mouse cursor position to be removed from the list of fixed contour nodes.

### *Add/Delete Repulsor*

Clicking on this button allows the user to add or delete repulsor nodes. Using the left mouse button causes repulsor nodes to added at the current mouse cursor position; the right mouse button causes the repulsor nearest to mouse cursor position to be removed.

### *Influence*

This parameter controls the value of repulsor coefficient (CoRepulse). It takes a value in the range 0 to 200; the 200 actually maps onto a value of 2.0.

### *Regional Influences*

This control panel allows areas (existing contours annotated over the source image) to have an influence on the constraint energy of the snake.

### *Make Region A Repulsor*

Clicking on this button allows the user to select a closed contour (or region), from the set annotating the image, to be used as a set of repulsor nodes. The user can control the influence of repulsion by using the slider associated with the constraint energy panel.

### *Make Region an Attractor*

Clicking on this button allows the user to select a closed contour (or region), from the set annotating the image, to be used as a set of attractor nodes. The user can control the influence of attraction by using the slider associated with the constraint energy panel.

### *Get Seed From Image*

This panel allows the user to control the generation of a seed contour using the zero crossing edge operator. This facility requires the user to follow the following sequence of actions:

1. Define a region of interest (a rectangular box will suffice) within the image using the drawing facilities associated with the Hand Segmentation Tool. Note that this algorithm will actually use a box of twice the width and height of that drawn.
2. Go to the Select Seed button and select this region of interest.
3. Select the appropriate lower gaussian using the slider in the image constraints panel.

### *Get Boundary*

Once this sequence of operations has been performed, pressing on this button causes the zero crossing edge (DOG) operator to be run within the extended region of interest. This results in the display of the major contour in that area. Note that the DOG operator is designed to look for dark objects on a light background; if a light object on a dark background is the object of interest, the user must negate the look up table (LUT) before pressing this button.

### *Accept Boundary*

Pressing this button causes the result of Get Boundary to be added to the current set of contours annotating the source image. The contour can then be used in the same way as any other contour, for example using it as a seed contour for the active contour model

### *Close*

Clicking on this button causes the active contour model dialogue to be closed.

*Help*

Clicking on this causes available help pages relating to the active contour model and the dialogue to be displayed.