

Medical Image Segmentation Using Topologically Adaptable Snakes

Tim McInerney and Demetri Terzopoulos

Dept. of Computer Science, University of Toronto, Toronto, ON, Canada

Abstract. This paper presents a technique for the segmentation of anatomic structures in medical images using a topologically adaptable snakes model. The model is set in the framework of domain subdivision using simplicial decomposition. This framework allows the model to maintain all of the strengths associated with traditional snakes while overcoming many of their limitations. The model can flow into complex shapes, even shapes with significant protrusions or branches, and topological changes are easily sensed and handled. Multiple instances of the model can be dynamically created, can seamlessly split or merge, or can simply and quickly detect and avoid collisions. Finally, the model can be easily and dynamically converted to and from the traditional parametric snakes model representation. We apply a 2D model to segment structures from medical images with complex shapes and topologies, such as arterial “trees”, that cannot easily be segmented with traditional deformable models.

1 Introduction

The segmentation and quantification of anatomic structures is an essential stage in the analysis and interpretation of medical images. Image interpretation tasks such as registration, quantitative analysis, labeling, and motion tracking require anatomic structures in the original image to be reduced to a compact, geometric representation of their shape. Recently, segmentation techniques which combine a local edge extraction operation with the use of active contour models, or snakes [4], to perform a global region extraction have achieved considerable success (see, eg., [3, 2, 1, 6]). These models simulate elastic material which can dynamically conform to object shapes in response to internal forces, external image forces and user-provided constraints. The result is an elegant method of linking sparse or noisy local edge information into a coherent free-form object model.

The application of active contour models to extract regions of interest is, however, not without limitations. The models were designed as interactive models and are sensitive to their initial conditions. Consequently, they must be placed close to the preferred shape to guarantee good performance. The internal energy constraints of the models can limit their geometric flexibility and prevent a model from representing long tube-like shapes or shapes with significant protrusions or bifurcations. Furthermore, the topology of the structure of interest must be known in advance since classical snakes models are parametric and are incapable of topological transformations without additional machinery.

Several researchers have attempted to address some of these limitations. Terzopoulos [7] and Cohen [3] used an internal “inflation” force to expand the deformable surface or snake past spurious edges towards the real edges of the structure, making the model less sensitive to initial conditions. Samadani [6] used a heuristic technique based on deformation energies to split and merge active contours. Similar split and merge operations were done interactively in [2]. More recently, Malladi *et al.* [5] developed a topology independent active contour scheme based on the modeling of propagating fronts with curvature dependent speeds, where the propagating front is viewed as an evolving level set of some implicitly defined function.

Most active contour models are parametric models whose parameterization is pre-specified and does not change automatically throughout the deformation process. If the topology of an object is fixed and known a priori, such models are preferable in that they provide the greatest constraint. Implicit models on the other hand, such as the formulation used in [5], provide topological and geometric flexibility through their level sets. They are best suited to the recovery of objects with complex shapes and unknown topologies. Unfortunately, implicit models are not as convenient as parametric models in terms of mathematical formulation, for shape analysis and visualization, and for user interaction.

This paper presents a new snakes model formulation that is set in the framework of domain subdivision using simplicial decomposition. This framework allows the model to maintain the traditional properties associated with snakes, while overcoming the limitations described above. We develop a parametric model that has the power of an implicit formulation by using a superposed simplicial grid to quickly and efficiently reparameterize the model during the deformation process, allowing it to flow into complex shapes and dynamically sense and change its topology as necessary. Multiple instances of the model can be dynamically created or destroyed, can seamlessly split or merge, or can simply and quickly detect and avoid intersections with other models. The result is a simple, convenient and flexible model that considerably extends the geometric and topological adaptability of snakes.

We apply our topologically adaptable snakes model to segment structures from medical images with complex shapes and topologies that cannot easily be segmented with traditional snakes models. In this paper, we consider the 2D case only, although the model is readily extensible to three dimensions.

2 Model Implementation

We begin by defining our model as a closed elastic contour (in 2D) consisting of a set of nodes interconnected by adjustable springs [2]. The elastic contour model is a discrete approximation to the traditional snakes model and retains all of the properties of snakes. That is, an “inflation” force pushes the model towards image edges until it is opposed by external image forces, the internal spring forces act as a smoothness constraint, users can interact with the model

using spring forces, and the deformation of the model is governed by discrete Lagrangian equations of motion.

Unlike traditional snakes, the set of nodes and interconnecting springs of our model does not remain constant during its motion. That is, we decompose the image domain into a grid of discrete cells. As the model moves under the influence of external and internal forces, we reparameterize the model with a new set of nodes and springs by efficiently computing the intersection points of the model with the superposed grid. By reparameterizing the model at each iteration of the evolutionary process, we create a simple, elegant and automatic model subdivision technique as well as an unambiguous framework for topological transformations. This allows the model to be relatively independent of its initial placement and “flow” into complex shapes with complex topologies in a stable manner. Furthermore, conversion to and from a traditional parametric snakes model representation is simply a matter of discarding or superposing the grid at any time during the evolutionary process.

2.1 Discrete Snake Model

We define a discrete snake as a set of N nodes indexed by $i = 1, \dots, N$. We associate with these nodes time varying positions $\mathbf{x}_i(t) = [x_i(t), y_i(t)]$ and a mass m_i along with compression forces which make the snake act like a series of unilateral springs that resist compression, rigidity forces which make the snake act like a thin wire that resists bending, and external forces that act in the image plane. We connect the nodes in series using nonlinear springs. This forms a discrete dynamic system whose behavior is governed by the set of ordinary differential equations of motion

$$m_i \ddot{\mathbf{x}}_i + \gamma_i \dot{\mathbf{x}}_i + \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i = \mathbf{f}_i \quad , \quad (1)$$

where $\ddot{\mathbf{x}}_i$ is the acceleration of node i , $\dot{\mathbf{x}}_i$ is its velocity, m_i is its mass, γ_i is a damping coefficient that controls the rate of dissipation of the kinetic energy of the node and \mathbf{f}_i is an external force that attracts the model toward salient image edges [4]. The force

$$\boldsymbol{\alpha}_i = a_i \epsilon_i \hat{\mathbf{r}}_i - a_{i-1} \epsilon_{i-1} \hat{\mathbf{r}}_{i-1} \quad (2)$$

makes the snake resist expansion or compression, where $\mathbf{r}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and the caret denotes a unit vector, a_i is the spring stiffness and $\epsilon_i = \|\mathbf{r}_i\| - L_i$, where L_i is the spring “rest” length. Since a new set of model nodes and springs is computed during every iteration, we update these rest lengths by setting them equal to the new spring lengths. The “rigidity” forces

$$\boldsymbol{\beta}_i = b_{i+1}(\mathbf{x}_{i+2} - 2\mathbf{x}_{i+1} + \mathbf{x}_i) - 2b_i(\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}) + b_{i-1}(\mathbf{x}_i - 2\mathbf{x}_{i-1} + \mathbf{x}_{i-2}) \quad , \quad (3)$$

act to make the snake resist bending. When computing this force, we “normalize” the spring lengths to account for the uneven node spacing. Finally, an “inflation” force, $\mathbf{h}_i = k\mathbf{n}_i$, is used to push the model towards image edges until it is opposed by external image forces, where k is the force scale factor and \mathbf{n}_i is the

unit normal to the contour at node i . The use of an inflation force essentially eliminates the need for an inertial force term. For this reason, we have simplified the equations of motion, while still preserving useful dynamics, by setting the mass density m_i in equation (1) to zero to obtain a model which has no inertia and which comes to rest as soon as the applied forces vanish or equilibrate. We integrate this first-order dynamic system forward through time using an explicit Euler method.

2.2 Domain Decomposition

The grid of discrete cells used to approximate the deformable contour model is an example of space partitioning by simplicial decomposition. There are two main types of domain decomposition methods: non-simplicial and simplicial. Most nonsimplicial methods employ a regular tessellation of space. These methods are fast and easy to implement but they cannot be used to represent surfaces or contours unambiguously without the use of a disambiguation scheme.

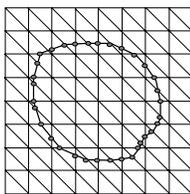


Fig. 1. Simplicial approximation of a contour model using a Freudenthal triangulation. The model nodes (intersection points) are marked.

Simplicial methods, on the other hand, are theoretically sound because they rely on classical results from algebraic topology. In a simplicial decomposition, space is partitioned into cells defined by open simplicies (an n -simplex is the simplest geometrical object of dimension n). A simplicial cell decomposition is also called a triangulation. The simplest triangulation of Euclidean space \mathcal{R}^n is the Coxeter-Freudenthal triangulation (Fig. 1). It is constructed by subdividing space using a uniform cubic grid and the triangulation is obtained by subdividing each cube in $n!$ simplicies.

Simplicial decompositions provide an unambiguous framework for the creation of local polygonal approximations of the contour or surface model. The set of simplicies (or triangles in 2D) of the grid that intersect the contour (the boundary triangles) form a two dimensional combinatorial manifold that has as its dual a one dimensional manifold that approximates the contour. The one dimensional manifold is constructed from the intersection of the true contour with the edges of each boundary triangle such that a line segment approximates the contour inside this triangle (Fig. 1) and such that the contour intersects each boundary triangle in 2 distinct points, each one located on a different edge. The set of all these line segments constitute the combinatorial manifold that approximates the true contour.

The cells of the triangulation can be classified in relation to the partitioning of space by the closed contour model by testing the “sign” of the cell vertices during the each time step. If the signs are the same for all vertices, the cell must be totally inside or outside the contour. If the signs are different, the cell must intersect the contour (Fig. 2).

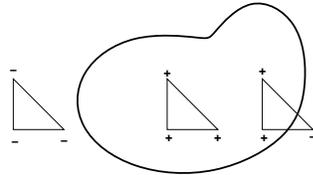


Fig. 2. Cell classification.

2.3 Topological Transformations

When a deformable model collides with itself or with another model, or when a model breaks into two or more parts, a topological transformation must take place. In order to effect consistent topological changes, consistent decisions must be made about disconnecting and reconnecting model nodes. The simplicial grid provides us with an unambiguous framework from which to make these decisions. Each boundary triangle can contain only one line segment to approximate a closed snake in that triangle. This line segment must intersect the triangle on two distinct edges. Furthermore, each vertex of a boundary triangle can be unambiguously classified as inside or outside the snake. When a snake collides with itself, or when two or more snakes collide, there are some boundary triangles that will contain two or more line segments. We then choose two line segment end-points on different edges of these boundary triangles and connect them to form a new line segment. The two points are chosen such that they are the closest points to the outside vertices of the triangle and such that the line segment joining them separates the inside and outside vertices (Fig. 3). Any unused node points are discarded. With this simple strategy, topological transformations are handled automatically, consistently and robustly.

To determine inside and outside vertices of a boundary triangle, we use a simple, efficient, and robust ray casting technique. That is, we count the number of intersections that a ray cast from a cell vertex along its grid row makes with the enclosing snake. An odd number of intersections indicates that the vertex is inside the snake. Counting the number of intersections is simply a matter of a look-up into an active edge-list table, constructed during each time step when a snake is projected onto the grid.

Once the topological transformations have taken place, we can run through the list of nodes generated by the procedure above and perform contour tracings via the grid cells, marking off all nodes visited during the traversals. In this fashion we find all new snakes generated by the topological transformation phase

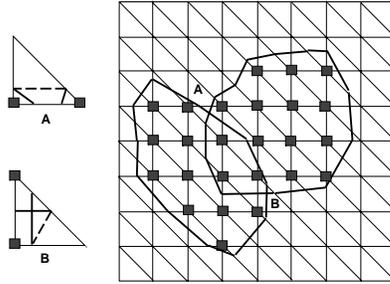


Fig. 3. Intersection of two models with “inside” grid cell vertices marked. Model nodes in triangles A and B are reconnected as shown.

and assign each a unique identifier. The result is that at any time during the evolutionary process, we can track, control, and interact with each model created.

2.4 Tracking Interior Regions for Model Collision Detection

We can keep track of the interior regions of our closed contour models as they deform. Interior region information can be used to perform efficient collision detections among multiple models or potentially to aid in a region analysis of the structures extracted by the models. We track the interior region of a model using an efficient update mechanism. As a closed snake deforms, new interior regions of the snake are defined, usually encompassing all or part of the old interior. We then use an efficient region filling algorithm to “turn on” all new cell vertices and turn off vertices in regions no longer encompassed by the snake. In practice, the increased (or decreased) area encompassed by the new snake boundary is usually small compared to the current area, meaning that few vertices require “filling” or “emptying”.

When segmenting multiple objects with common or adjacent boundaries using multiple models, an efficient collision detection scheme could be utilized as an extra constraint in the segmentation process. By keeping track of the interior regions of our models, we can perform efficient collision detection between models. Each snake simply fills the grid vertices of its interior with a unique value, effectively giving these vertices an identifier (Fig. 4). As a snake deforms, it may attempt to move into new grid triangles. A simple and efficient check can then be performed to see if any of the vertices of these new grid cells are already filled with a value different from the value of the interior vertices of the current snake. If so, various “avoidance” strategies can be used such as applying a force to the intersecting nodes of the colliding model to push the model out of the already occupied region.

3 Segmentation Results

Currently our topologically adaptable snakes model based segmentation procedure requires the user to draw an initial contour (or multiple contours, if desired)

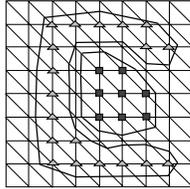


Fig. 4. Collision detection and avoidance using model interior region “identifiers”. A model cannot deform into “territory” already occupied by another model.

within the objects of interest. These contours are then closed and converted to a parametric model representation using the superposed grid. The snakes are then updated during each time step by: calculating the forces on each snake node and updating the node positions using the simplified equation (1), reparameterizing each snake (computing a new set of nodes and springs) by finding the intersection points of the snake with the grid, computing the new spring rest lengths L_i , performing topological transformations within grid cells as needed, and traversing each snake via the grid cells, identifying all new snakes.

3.1 Experiments with Synthetic Data

In the first experiment we demonstrate the “flowing” property of the snake by segmenting a spiral shaped object (Fig. 5a–d). We superposed a 40×40 square cell grid, where each cell is divided into two triangles, onto a 128×128 pixel image. The parameter values for all of the experiments with synthetic data sets are: $\Delta t = 0.002$, $a_i = 10.0$, $b_i = 5.0$, $k = 20.0$, $\kappa = 20.1$ (note: κ is the external image force scale factor):

In the second set of experiments we demonstrate the topological transformation capabilities of the model. In Figure 5e–h, a snake flows around two “holes” in the object, collides with itself and splits into three parts. In Figure 5i–l, several snakes are initialized in the protrusions of the object, flow towards each other, and seamlessly merge. In Figure 5m–p, the snake shrinks, wraps and finally splits to segment each object.

3.2 Experiments with 2D Medical Images

In the first experiment we use several 256×256 image slices of a human vertebra phantom. We overlay these images with a 64×64 square cell grid, with each cell divided into two triangles. The parameter values are: $\Delta t = 0.001$, $a_i = 30.0$, $b_i = 15.0$, $k = 60.0$, $\kappa = 62.0$. Once the snake collides with itself in Figures 6a–d, it automatically splits into three parts, two parts segmenting the inner boundary of the vertebra and one for the outer boundary. In Figures 6e–h, the snake shrinkwraps itself around the objects and splits to segment the different parts.

In the second set of experiments, we apply the model to a 128×128 portion of a retinal image (Figs. 7a–d) to segment the vascular “tree”, a structure with extended branches and bifurcations. Note that the arteries and veins do not

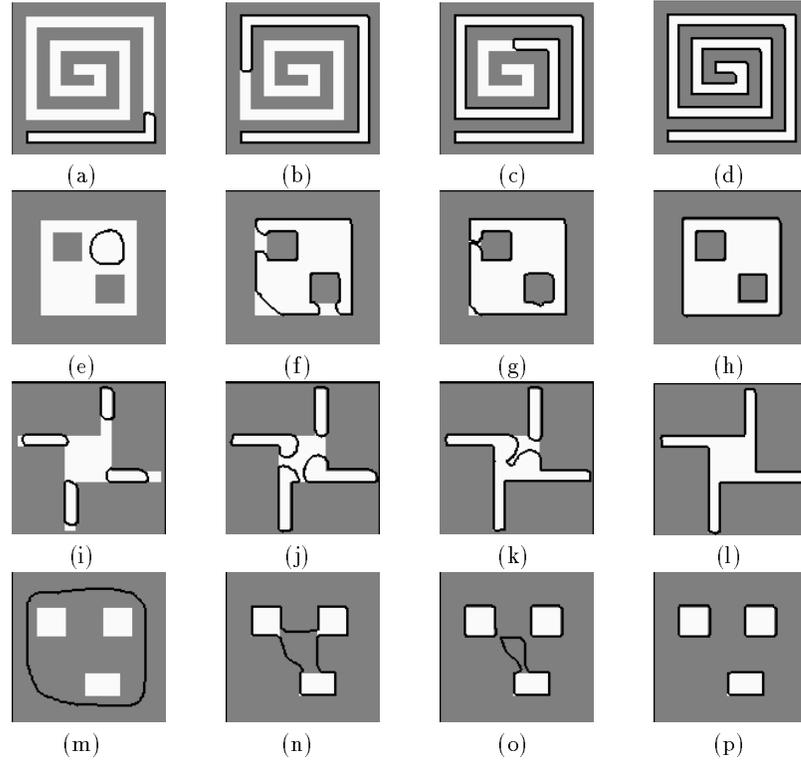


Fig. 5. Segmentation of objects with complex geometries and topologies.

physically intersect as they appear to do in this 2D image projection. Currently, we do not attempt to identify vessel intersections or bifurcations. A 64×64 square cell grid is used and the parameter values are: $\Delta t = 0.001$, $a_i = 30.0$, $b_i = 30.0$, $k = 60.0$, $\kappa = 70.0$. In Figures 7e–h we have initialized several models in various branches, demonstrating the potential for automating and parallelizing the segmentation process.

In the final experiment we demonstrate the use of the model collision detection property. A sagittal slice of an MR brain image was manually segmented into four anatomical regions and the pixels in each region were assigned a constant intensity level. We initialized two snakes in regions 4 and 3 and recovered the shapes of these regions (Fig. 8). We then initialized two snakes in regions 1 and 2 and allowed the snakes to flow, using only a minimum intensity threshold as the external image force. When these snakes attempt to flow into the brighter regions 4 and 3 they collide with the initial snakes and are forced to assume the shape of the common boundary regions (Fig. 8c).

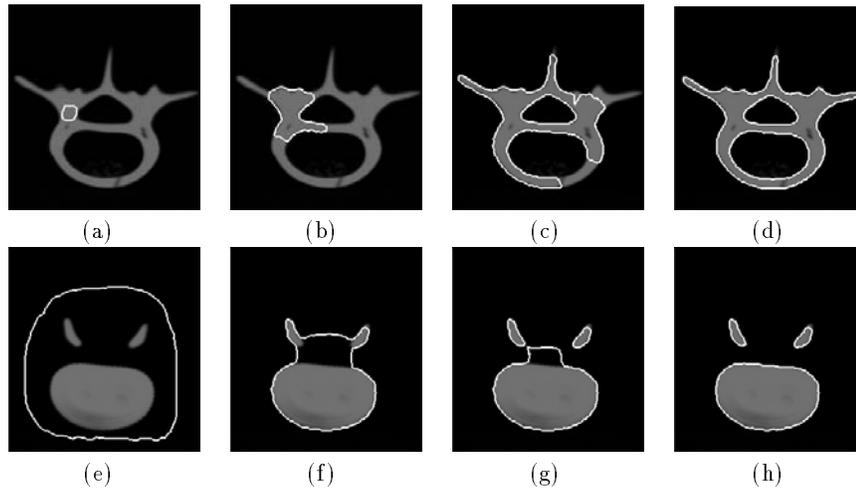


Fig. 6. Segmentation of two cross sectional images of a human vertebra phantom.

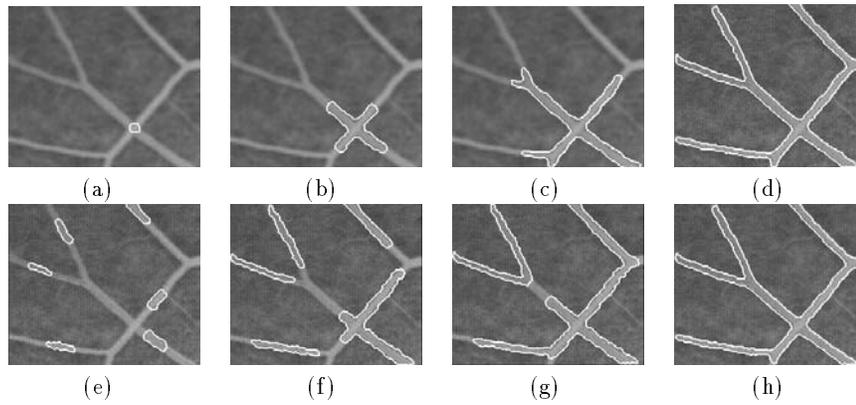


Fig. 7. Segmentation of the blood vessels in angiogram of retina. Top row: using a single snake. Bottom row: using multiple snakes.

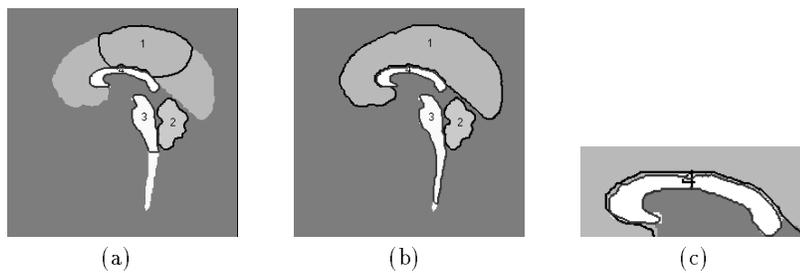


Fig. 8. Segmentation of anatomical regions of the brain using multiple models. The collision detection property was used as an extra constraint when segmenting regions 1 and 3.

4 Conclusion

We have developed a topologically adaptable snakes model that can be used to segment and represent structures of interest in medical images. By combining a domain decomposition technique with snakes, we have considerably extended the snake capabilities and overcome many of the limitations while keeping all of the traditional properties associated with these models. By iteratively reparameterizing the snake using the superposed grid, we have created a simple and efficient automatic subdivision technique as well as an unambiguous framework for topological transformations. Furthermore, the model has all of the functionality of the implicit level set techniques described in [5], but does not require any mathematical machinery beyond the basic formulation of classical snakes. Since the model retains a parametric formulation, users can control and interact with it as they would with traditional snakes. We have applied the model to various 2D synthetic and medical image datasets in order to segment structures of interest that have complicated shapes and topologies. We are currently extending the model to a three dimensional surface representation based on a tetrahedral decomposition of the image domain.

Acknowledgements

The segmented MR brain image was provided courtesy of Stephanie Sandor and Richard Leahy of the Signal and Image Processing Institute, Dept. of Electrical Engineering Systems, University of Southern California. TM is grateful for the financial support of an NSERC postgraduate scholarship. DT is a Fellow of the Canadian Institute for Advanced Research. This work was made possible by the financial support of the Information Technologies Research Center of Ontario.

References

1. M. O. Berger and R. Mohr. Towards Autonomy in Active Contour Models. In *Proc. of Tenth International Conference on Pattern Recognition*, pages 847–851, 1990.
2. I. Carlbom, D. Terzopoulos, and K. Harris. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Transactions on Medical Imaging*, 13(2):351–362, 1994.
3. L.D. Cohen. On active contour models and balloons. In *CVGIP: Image Understanding*, volume 53(2), pages 211–218, March 1991.
4. M. Kass, A. Witkin, and D. Terzopoulo. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
5. R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*. In Press.
6. R. Samadani. Changes in connectivity in active contour models. In *Proceedings of the Workshop on Visual Motion*, pages 337–343, March 1989.
7. D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

This article was processed using the \LaTeX macro package with LLNCS style