A Simple 2D Active Contour Model to Segment Non-Convex Objects in 3D Images

Benoit Godbout*, Claude Kauffmann, Jacques A. de Guise

Laboratoire d'Imagerie en Orthopédie, Hôpital Notre-Dame
1560 Sherbrooke Est [DeSève Y-1614] (Montréal), H2L-4M1

LIVIA, École de Technologie Supérieure
1100 Notre-Dame ouest (Montréal) H3C-1K3

Abstract

In order to help the segmentation of 3D images like those obtained from medical CT or MRI, a new simplified active contour model is presented. First, the 'snake's' [1] internal energy minimization concept is replaced by a simple filtering operation using an idealized filter. This has the effects of reducing the contour shrinking problem and simplifying the complex and often empirical definition of the parameter set inherent to procedure initialization.

Starting from a 3D data set, 3D edge information can be used to induce a more appropriate contour point displacement by taking into account the object's surface existing between adjacent slices. This will allow automatic contour splitting and merging when non-convex objects present important topological changes from slice to slice.

A 3D semi-automatic segmentation tool was developed to test this new method using 3D synthetic data and real CT images of a human spine. User defined parameters are limited to the contour sampling distance and the maximum allowable curvature. Results show that complex shapes can easily be segmented in a 3D data set using only few approximate initial 2D contours.

1. Introduction

Three-dimensional imaging systems such as high resolution CT or MRI often provide 2D digital images or 'slices' that can be stacked as a digital volume data set. This information can be processed slice by slice or as a whole using line and surface segmentation procedures to construct polygonal surface models. Such models are very helpful to represent, manipulate and analyze specific biological structures with 3D computer graphics engines. However, the segmentation process is very difficult for complex structures such as the human spine and no completely automatic and precise procedure is actually satisfactory, especially if it is to be used in a medical environment. Our goal is to produce a simple semi-automatic tool to help the segmentation process, using a 'slice by slice' scheme but with the possibility of adding local 3D information available from neighboring slices.

Many reasons make active contour methods suitable for computer assisted segmentation:

- high precision can be achieved from a coarse manually defined initial contour;
- contours can be defined explicitly to be closed around some objects or regions;
- users can easily edit the results;
- final contours on a slice can be used as approximate contours on neighboring slices to minimize user interaction [2,3];
- compared to global methods, the local nature of the search let us focus on specific objects to facilitate registration and reduce computation time.

Active surfaces [4] could effectively help segment 3D structures but working with 3D meshes in a 3D image space makes user interaction very difficult. This work presents a new simplified 2D active contour model based on a 3D surface following scheme coupled with a simplified regulation procedure. These simplified active contours have also been successfully applied to 2D radiographic image segmentation by Kauffmann & al. [5].

The paper is organized as follows: the original 'snake' model is presented in section 2. We aim to solve some of the problems associated with this method with a new simplified regulation procedure defined in section 3. The 3D surface following process used to control the image driven contour point displacement is

* e-mail : godbout@livia.etsmtl.ca

explained in section 4. Semi-automatic segmentation, contour sampling and contour merging are discussed in section 5. Section 6 presents results obtained on synthetic and real CT images of human vertebrae. A brief discussion follows in the 7th and final section.

2. Active Contour Models

Active contours models proposed by Kass & *al.* [1] also termed 'snakes', are a sophisticated approach to contour extraction and image interpretation. The technique is built upon an energy minimization framework. The global snake energy is defined by:

$$E_{snake}(s) = \alpha(s) \cdot \left| \frac{\partial v(s)}{\partial s} \right|^2 + \beta(s) \cdot \left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2 + E_{image}(v(s)) \quad (1)$$

with v(s) = (x(s), y(s)), where x(s), y(s) are x, y coordinates along the s ε [0,1] contour normalized arc length.

In equation (1), E_{image} is the image energy term which is constructed to attract the snake to desired feature points in the image, such as edges, lines, terminations, etc. This provides *a posteriori* information.

The other terms represent the internal energy (E_{int}) which controls the natural behavior of the snake. The first order term, weighted by α(s), makes the contour behave elastically, while the second order curvature term, weighted by β(s), makes it resistant to bending. This provides the contour regularization and is associated with the *a priori* constraints. After discretizing the snake curve $v_i = (x_i, y_i)$ and approximating derivatives with finite differences, equation (1) is solved with an iterative process:

$$\begin{cases} x_t = (A+\gamma I)^{-1}(x_{t-1} - f_x(x_{t-1}, y_{t-1})) \\ y_t = (A+\gamma I)^{-1}(y_{t-1} - f_y(x_{t-1}, y_{t-1})) \end{cases} \quad (2)$$

where the damping parameter γ determines the process convergence rate. Equation (2) can be considered as a composition of snake attraction to the image features and shape smoothing imposed by the stiffness matrix *A*.

We showed in (1) and (2) that the active contour model's behavior is governed by its parameters (α, β, γ). Variations in these parameter values give rise to different solutions with the same initialization contour. Their choice and fine tuning to encourage a desired solution is difficult and can not be done easily in an intuitive way. Another problem is the sensibility to initial position. These problems are highly amplified when noise increases in the images. Poor contrast is also a problem in that minima may not be prominent enough to halt the snake's natural contracting forces, and consequently the snake fails to be attracted by the required feature. Methods were proposed [2,4] to use an additional normal force to inflate the contour. However, the contraction force remains, which complicates the determination of the normal force parameter.

Dynamic programming [6] has been demonstrated as a universal optimization technique and presents the advantage to include hard constrains but the computational costs are high, especially when the search zone gets bigger. The greedy algorithm [7] is also based on a neighborhood search. The local nature of this approach implies poor performance in the presence of noise and does not guarantee a local minimum solution.

The simplified active contour model was developed to simplify the parameter set and reduce the contraction effect while minimizing calculations. It will be introduced in the next section.

3. Simplified Active Contours

We propose a simplified active contour method to achieve a compromise between the dynamic programming and 'snake' solutions. Much like the 'snake', it uses a two step procedure. First, each point is moved toward the best local edge found in an application defined search zone. In this case, the surface following scheme described in section 4 is used. Secondly, the contour is regularized by a numeric filtering process.

Equation (1) shows that minimizing internal energy gives a smooth curve. One can see that, in equation (2), the regulation matrix $(A+\lambda I)^{-1}$ acts just like a discrete low-pass filter applied to contour coordinates. Those observations give us the motivation to analyze the performance of such filtering in the frequency domain.

If we consider a closed contour made of equally spaced discrete points, where *d* is the distance between two points and *R* the local curvature radius at this point, the corresponding frequency is given by *d/R*. The amplitude response at this frequency should correspond to the contraction ratio when the contour is filtered. We present on Figure 1 the magnitude-frequency response of the filter defined by the 'snake' regulation matrix.

Different values of α and β are used to illustrate how the contour can behave elastically or be resistant to bending. One can note that in all cases a short pass-band and a wide and smooth transition-band characterize the filter. This highlights the fact that it is very difficult to obtain a filter with a desired pass-band

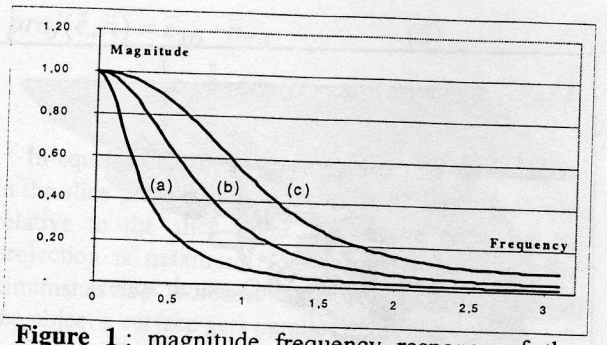

Figure 1 : magnitude frequency response of the 'snake' equivalent filter from equation (2). Parameter values are: (a) $\alpha = 12$, $\beta = 1$; (b) $\alpha = 2$, $\beta = 1$; (c) $\alpha = 0.5$, $\beta = 0.5$.

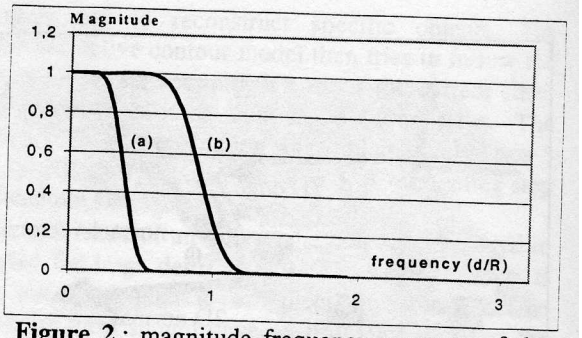

Figure 2 : magnitude frequency response of the filter defined by equation (3). Parameter values are: (a) d=3, R=8; (b) d=3, R=3.

and a sharp cut-off frequency. Therefore, we can search for a low-pass filter that overcomes these problems.

The proposed solution is to design a numeric filter that extends the unitary gain pass-band without sacrificing the transition band. This should reduce the shrinking problem for low curvature parts of the contour while preserving good contour continuity properties. We use a classical idealized low-pass filter defined by a (sinx)/x shape multiplied by a Hamming window function. The discrete coefficients of the numeric filter are given by equation (3).

$$\begin{cases} H(k) = \dfrac{\sin(w_c k)}{w_c k} \cdot \left(0.46 + 0.54 \cdot \cos\left(\dfrac{2 \cdot \pi \cdot k}{L} \right) \right) \\ where : w_c = \dfrac{d}{R} \quad and \quad -L < k < L \end{cases} \tag{3}$$

The filter size parameter L determines the slope of the transition band. To reduce user interaction, we use $L = 10*R/d$, which gives satisfying results. The magnitude-frequency response of this filter is shown in Figure 2. The cutoff frequency w_c can simply be chosen in a way that portions of contour with a certain minimal curvature radius R can be preserved.

By directly using a discrete filter, the concept of the snake model can be simplified. The contour points still move independently according to a translation scheme but the resulting curve is now only considered as a noisy signal that must be filtered to respect some maximal curvature constraints. With this simplification, the parameter set has now gained a physical meaning. R is a minimal allowable curvature radius and d is the sampling distance of the discrete curve. Both values can be expressed in pixels. Also, the excessive contraction problem for high curvature radius portions should be considerably reduced.

The regulation is only the second step in the iterative active contour solution. First, the points must move toward significant features on the image. The next section introduces our surface following translation scheme.

4. Surface Following

Even if 2D contours are used, 3D information is present in the form of explicit 3D surface edges forming the external boundaries of a given object. Relevant 2D contours in a slice can be defined as the intersection of the object's closed surface with the slice plane. Segmentation is the search for all visible 2D surface sections in slices. The role of the translation scheme is to guide individual contour points toward those object features.

In order to detect potential surface edges, the Zucker-Hummel [8,9] operator is used to compute 3D local edge vectors $\vec{e}_{3D}$. In the center slice, classic 3x3x3 directional masks are used. When edge values are needed in adjacent slices, a linear interpolation is included in modified 3x3x2 masks. This is equivalent to using the halfway slice as a neighbor. It increases the z axis accuracy of the method.

All active contour methods need an adequate initial contour to find the right solution. To minimize user interaction, some authors [3,4] use copies of accurate final contours from an adjacent slice as initial contours for the current one. These can themselves be optimized and serve as initial contours in the subsequent slice. For this to work, it must be assumed that the deformation between visible object portions on neighboring slices from volumetric data is bounded. For complex non-cylindrical objects, the assumption doesn't hold. Object sections can be subject to large deformations or to topological changes.

For our purpose, the contour copying process is viewed as a 1 voxel (volume pixel) translation of a surface section in the z direction. Normally, going from the initial to the final position, a continuous 3D surface

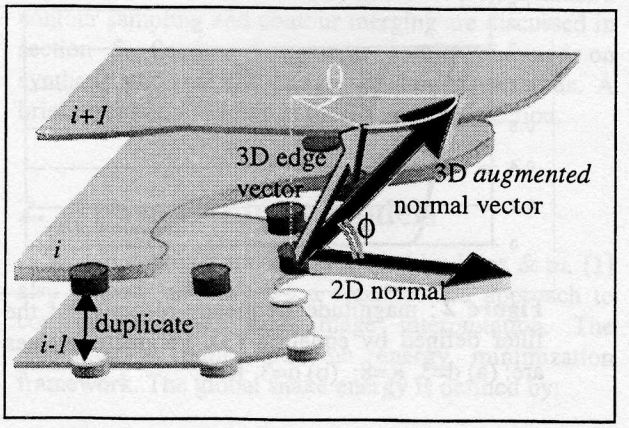

Figure 3 : Surface following scheme (1). The initial contour is duplicated from the slice below. The 3D *augmented* normal vector is computed from the 2D normal and the local 3D edge vector.

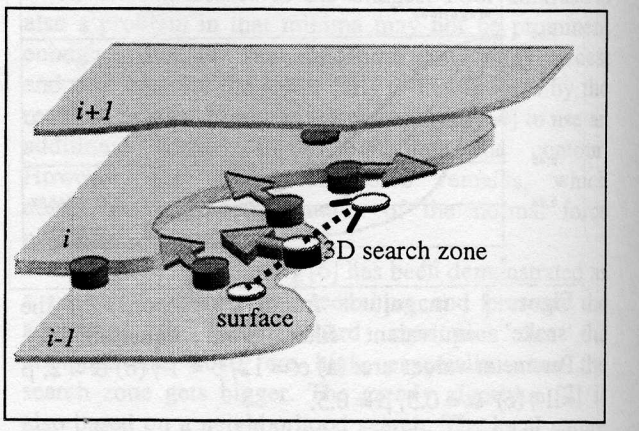

Figure 4 : Surface following scheme (2). The best edge is found in a 3D search zone in the *augmented* normal direction and the point moves accordingly. The resulting contour is then filtered.

should be present in between both slices. The resulting contour is constrained to stay on its own slice plane but can find its way to the current object section by following the nearby surface. If the surface is non-convex, it will stay closer than one voxel throughout the deformation process. These factors dictate the configuration of our 3D search zone.

The search for the closest surface edge is done in a 3D neighborhood but it must induce a two-dimensional translation in the slice plane. In order to do so,

- voxels in the direction of the 3D normal vector to the surface are considered to be in the exterior region;
- voxels in the opposite direction are considered in the interior region;
- a contour point moves in the contour's normal direction if it finds the object's surface on the exterior;
- it moves in the opposite direction if the surface is found inside the contour;
- the point stays in place if it is already on the best surface edge.

Because 2D contours are used, the 3D normal vector to the surface is unknown. To approximate it, we define an *augmented* normal vector $\vec{n}_{3D}$ as the normal vector to the curve $\vec{n}_{2D}$, rotated out of the plane until it gets the same angle ϕ as the local edge vector. Since the point is close to the object surface, the assumption that the orientation of the current edge vector is close to the orientation of the nearby surface should hold. Figure 3 demonstrates the principle.

$$\vec{n}_{3D} = \frac{(n_x, n_y, \tan(\phi))}{\left\|(n_x, n_y, \tan(\phi))\right\|} \qquad (4)$$

where

$$\tan(\phi) = \frac{e_z}{\left\|\vec{e}_{xy}\right\|}$$

To keep computing time to a minimum, the search zone is limited to 3 positions: the current one and the positions of the 2 neighbor voxels in the *augmented* normal direction of the curve. This search zone will be adapted to the surface edge orientation as long as the contour stays close to the surface. It should be true for surface following, but not when the contour gets further than 1 voxel, in a 3D sense, from the object. Again if the best edge is inside, the contour deforms inwards and if it is outside, it deforms outwards. A contour point stops moving when it becomes the best local surface edge. This can be seen in Figure 4.

The Zucker-Hummel edge vector length could be used to find the best possible surface in the search zone. However, the direction information from the curve will help align it to image edges. A simple projection to the normal wouldn't do because it would cancel the z component. Instead, a projection of the edge vector on the *augmented* normal vector ensures surface edge quality.

$$proj(\vec{e},\vec{n}) = \vec{e}_{3D} \cdot \vec{n}_{3D} \qquad (5)$$
$$= \left\|\vec{e}_{xy}\right\| \cdot \cos(\theta) \cdot \sin(\phi) + e_z^{\,2} \cdot \cos(\phi)$$

In equation (5), θ is the angle between both vectors in the slice plane and ϕ is the angle of the surface edge relative to the slice plane. It can be seen that the projection is maximum for ϕ equal to 0°. Its effect diminishes as ϕ increases and becomes irrelevant when the object's surface gets parallel to the slice plane.

Figures 3 and 4 illustrate the surface following procedure for one contour point. The contour in the middle slice is copied from the one below. The 3D search zone for a given point is then determined using the normal vector to the curve and the local edge orientation. Since the maximal edge value in the direction of the *augmented* normal vector is below, the point will move in the opposite direction to the normal vector. We also see that the point moved inward (in 2D) after it found the best edge inside (in 3D).

Once the translation phase has been applied to each point, the new curve is filtered to get the final contour for 1 iteration. This process is included in a global 3D segmentation procedure described in the next section.

5. Segmentation Procedure

We developed a 3D image segmentation tool based on this simplified active contour model and surface following scheme. User defined parameters are distance and minimum curvature radius. This is an important improvement in simplicity over other active contour models. Figure 5 resumes the segmentation strategy.

First, the user manually defines the contours of the object in slice near the center of the 3D data set. If different objects are segmented, an object tag is assigned. This tag will be used to allow contour merging and to reconstruct specific objects. The simplified active contour model then tries to follow the closest object surface until it crosses the current slice. This iterative process consists of four steps. The surface following translation was explained in section 4 and the filtering process in section 3. A resampling step and a contour merging step are added.

At both ends of an object, the contour will have to account for large deformations. The object surface is then almost parallel to the slice plane. The surface following term should have no problems to direct the contour toward the right edge even with its limited search zone but regulation works only if the distance between neighboring discrete contour points stays constant.

In the new method, to allow expansion or shrinking of the contour, the contour points are periodically resampled using the approximate resampling method proposed by Lobregt & al. [10]. This step keeps the distance between points within certain limits. Points are added when the distance becomes more than 50 percent over the desired distance. Inversely, points are removed when they get closer than 50 percent of the desired distance. This resampling combined with contour filtering leads to adequate parameterization of the curve.

Concavities or holes in a 3D object will cause the slice contours topology to change. Contours can split or merge. They can also disappear completely. These changes can be easily detected with contour crossing, leading to an appropriate redefinition of the contours. Figure 6 shows experimental results for different cases of contour redefinition.

The iterative process stops when the contour becomes stable. This stability is relative. The fact that discrete positions are used for every calculation induces a small oscillation in the final position. This reduces reproducibility to a 1 voxel error. Errors in 2D position can increase as the 3D object surface becomes parallel

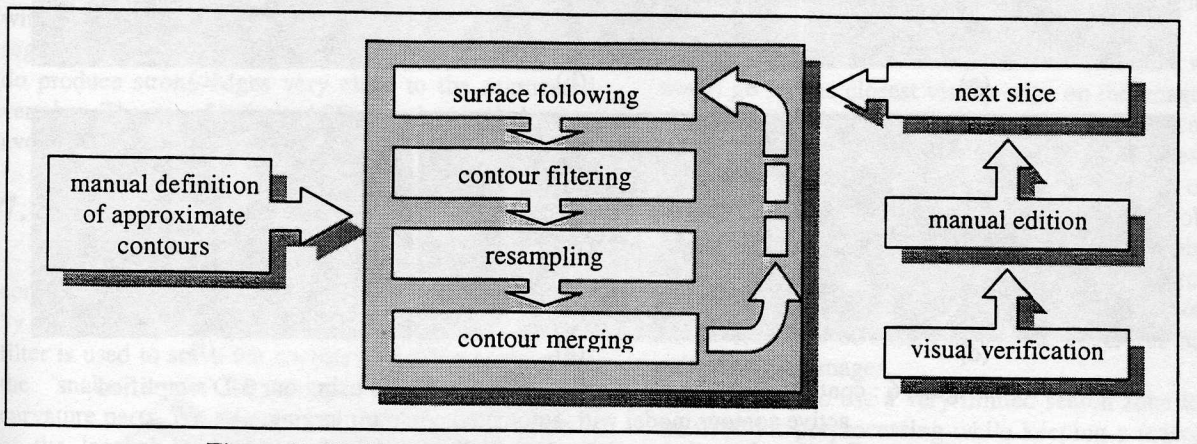

Figure 5 : 3D image segmentation strategy with simplified active contours

to the plane. Appropriate image filtering and a final step with linear edge interpolation would probably reduce this effect.

Final contours in a slice can be manually edited to correct segmentation errors. They are then automatically copied into the next slice to be used as initial contours. The process is repeated again and again until all object contours are found.

6. Results

We tested a C++ implementation of the segmentation procedure on various synthetic and clinical images. Key results are presented here. The first example shown in figure 6 consists of 5 noisy (s/n = 2) synthetic images merged into a 3D image block. The first and last images are duplicated to allow the use of a 3D operator. These images are designed to simulate topological changes that can occur when segmenting CT images of human vertebrae. We use the segmentation procedure described in section 5 with the mean distance between points set to 3 pixels and the minimum curvature radius set to 4 pixels. This is a relatively soft contour. Even if no sharp corners are present, we find that soft contours can best expand in noisy images. Coarser contours would have a great difficulty to get into concavities. Anyhow, our tests tend to prove that simplified active contours are stable on any image edges as long as the contour respects the curvature radius constraint.

Human interaction is limited to initial contour definition in the first image as shown in figure 6-(a). For the first image, the contours had to be defined very close to the object but after that, all relevant contours were found in subsequent images starting with the final contour from the previous image. We see that even with the limited search area and the absence of any filtering, the contour could account for large deformations of the visible edges by following the underlying surface. For a 2D method, a much larger search zone or appropriate filtering would have been needed, thus adding to the numerical complexity of the method.

Different topological changes were handled successfully. Contours have merged from 6-(c) to 6-(d) to form an internal and external contour. Then they crossed again and became one contour with a concavity in 6-(e). This contour then crossed itself to get back as two separate contours. This tends to demonstrate that the surface following method effectively forces contour to cross each other when a topological change occurs.

With these noisy images, some small bubble like segmentation errors are visible on parts of the resulting contours. These errors can be explained in part by the resampling approximation for the contour. In these bubbles, points are compressed by the translation scheme but not enough to remove a point. The net effect is a local lowering of the allowable curvature radius.

The diminution of the convergence rate as the noise increases in the images is another important drawback. Contour regulation is designed to preserve the local deformation tendencies. These tendencies can get lost as the noise level increases. Proper image filtering

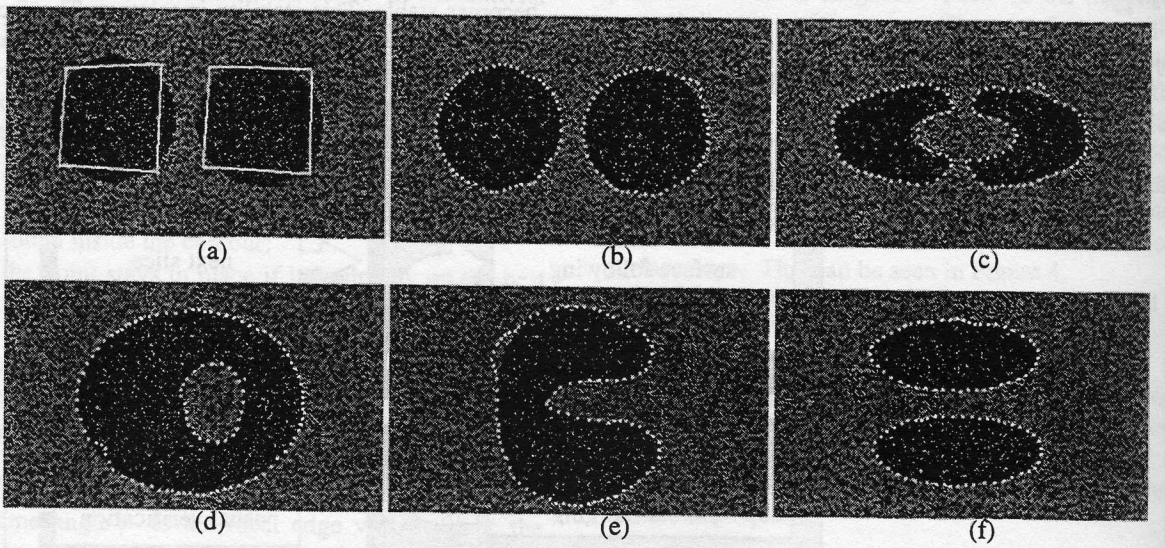

Figure 6 : contour evolution for 5 noisy synthetic images using our 3-D simplified active contour model with automated contour merging.
Contours from the previous slice are used as initial contours.

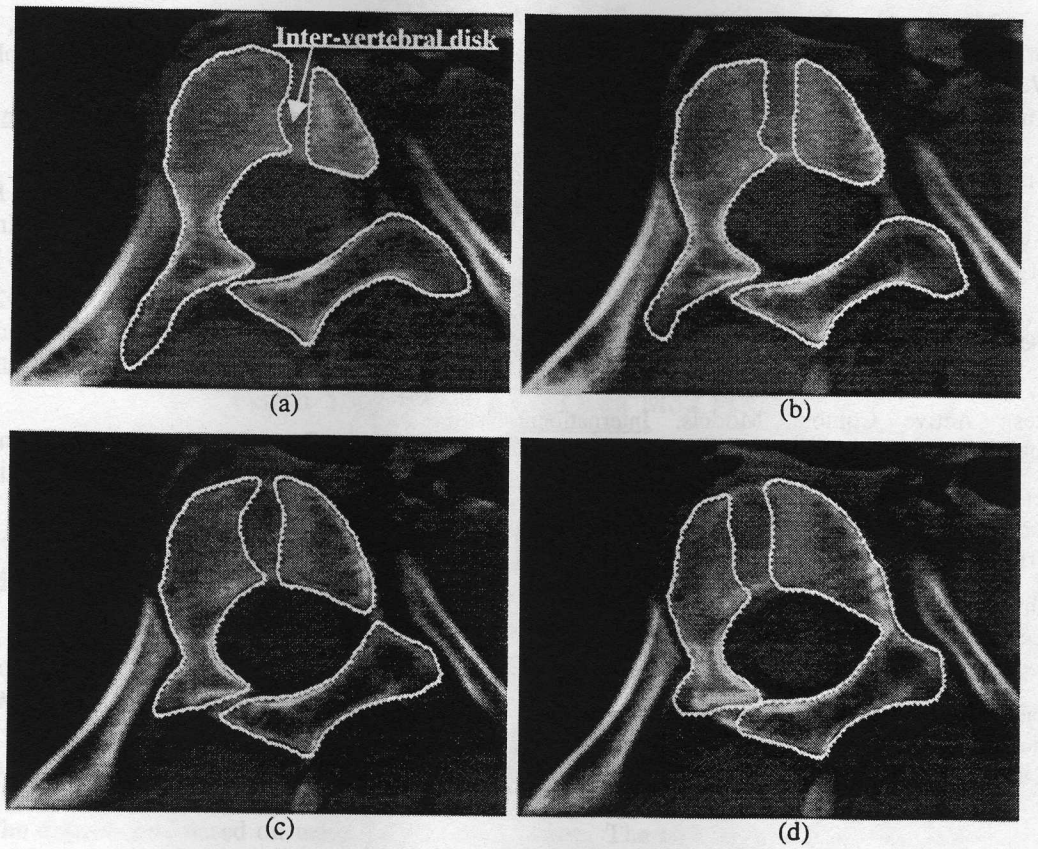

Figure 7 : contour evolution on CT images from scoliotic spine. Contours from the previous image are used as initial contour.

would be an asset to insure convergence.

The second example shows segmentation of CT images of a scoliotic spine. Because of this sickness, the vertebrae are tilted; the inter-vertebral disc becomes very hard to follow from slice to slice with traditional 2D active contour methods. We see on Figure 7 that the contours followed the disk surface properly. These disks are very difficult to see as 2D features but they become significant when 3D edges are considered.

Finally, in real images, unwanted nearby objects with strong edges can interfere with active contour segmentation. In this specific example, ribs and lungs do produce strong edges very close to the segmented vertebra. The use of a limited 3D search zone helped to avoid these unwanted objects.

7. Discussion

In this paper, we introduced a new simplified active contour model. The internal energy term was replaced by a simple filtering step. An approximation of an ideal filter is used to solve the contour shrinking problem of the 'snake' model for complex shapes with high curvature parts. We also showed that this simplification of the 'snake' leads to more intuitive definition of parameters. Only the maximum curvature radius in pixels and the sampling resolution of the contour must be specified.

The image energy is also replaced by a simple discrete translation step that pushes the contour points toward the best edge in an application defined search zone. To help segment 3D images, a surface following translation scheme was employed. This allows the use of simply defined 2D contours without sacrificing the 3D surface edge information. After looking for the closest surface in a 3D search zone, contour points are moved in the local curve normal direction.

Compared to some other 2D snakes methods, which would go for the closest visible edge on the image, our model will follow the closest surface, which can be anywhere in the 3 adjacent slices. That gives our contour a net tendency to stay on the chosen object rather than going for a better edge on another object. Experimental results shows that even for very oblique surfaces like a vertebral disk, the contour keeps track of the right edge. Furthermore, our simplified contour model seems to be stable on visible edges for binary, noisy and real images.

The ability to use a very limited search zone lowers the need for preprocessing while keeping a reasonable calculation time. This makes our method suitable for a semi-automatic interactive tool. Our first goal was to

develop a fast, simple and efficient method. We are now working on making the method more robust to noise. We also aim to generalize the method to be able to work in any slicing plane, in the quest to reduce and simplify human interaction for segmentation of 3D medical images.

This work is supported in part by NSERC and FCAR organizations.

Reference

1. Kass, M., Witkin, A., Terzopoulos,D. (1988). Snakes : Active Contour Models. International Journal of Computer Vision, 1(4), pp. 321-31.
2. Cohen, L.D. (1991). On Active Contour Models and Balloons. CVGIP: Image Understanding, 53(2), pp. 211-18.
3. Xiaohan, Y., Ylä-Jääski, J. (1995). Interactive Surface Segmentation for Medical Images. Proceedings of SPIE, vol. 2564, pp. 519-27.
4. Cohen, L.D., Cohen, I. (1993). Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. IEEE PAMI, 15(11), pp. 1131-47.
5. Kauffmann, C., Godbout, B, deGuise, J.A. (1998). Simplified Active Contour Model applied to bone structure segmentation in digital radiographs. Proceedings of SPIE : Medical Imaging 1998. San Diego (to appear).
6. Amini, A.A., Weymouth, T.E., Jain, R.C. (1990). Using Dynamic Programming for Solving Varitional Problems in Vision. IEEE PAMI, 12(9), pp. 855-66.
7. Williams, D.J., Mubarak, S. (1992). A Fast Algorithm for Active Contours and Curvature Estimation. CVGIP: Image Understanding, 55(1), pp. 14-26.
8. Benoit, C. (1995). Segmentation d'images tridimentionelles. Master Thesis. École Polytechnique, Montréal, 165 p.
9. Zucker, S.W, Hummel, R.A. (1981). A Three-Dimensional Edge Operator. IEEE PAMI, 3(3), pp. 324-330.
10. Lobregt, S., Viergever, M.A. (1995). A Discrete Dynamic Contour Model. IEEE Trans. on Medical Imaging, 14(1), pp. 12-24.