In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

USING ACTIVE CONTOURS FOR SEGMENTATION OF MIDDLE-EAR IMAGES

McGill University,
Montréal, Québec

January 2003

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

Master of Engineering

© Joubin Hatamzadeh-Tabrizi, 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisisitons et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-88357-4
Our file Notre référence
ISBN: 0-612-88357-4

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

ACKNOWLEDGEMENTS

I would like to thank all the people without whom I could never have accomplished this task. I wish to thank my supervisor, Professor W. Robert J. Funnell, for his advice, support and patience; my parents Doretta and Parviz, for their everlasting support, understanding and encouragements; and Goli, for her support and love throughout my study. I would like to thank my colleagues for their friendship and help.

We wish to thank L.-P. Hamelin, F. Labonté and B. Pelletier, for implementing the first version of the Oxiana programme. This work was supported by the Natural Sciences and Engineering Research Council (Canada) and the Canadian Institutes of Health Research. We thank C. C. Northrop and S. R. Levine from The Temporal Bone Foundation, Boston, for providing us with the histological data, and M. Henson & O.W. Henson, Jr., (UNC-Chapel Hill) and the Center for In Vivo Microscopy (CIVM), Duke University, for providing the MRM data.

ABSTRACT

Image segmentation, or the extraction of the boundaries of objects, is one of the most important problems in computer vision and image processing. As a high-level technique for boundary identification, active contours are used extensively for segmentation purposes.

Two different active contour approaches, *i.e.*, parametric active contours and discrete dynamic contours, were used and compared for the segmentation of middle-ear images. We used histological and Magnetic Resonance Microscopy (MRM) image datasets for our experiments.

Parametric and discrete dynamic contours show similar boundary identification results for the histological and MRM datasets. Gradient, Gradient Vector Flow (GVF), and the gradient plus pressure were used as the external force. The gradient has the disadvantage of having a restricted capture range. Two solutions for improving the capture range, gradient vector flow and pressure force, were compared. Although GVF provides a good capture range, it sometimes wrongly identifies the low-contrast boundaries. It was also found that GVF may wrongly identify the boundaries of close neighbouring structures. As an alternative, pressure forces have shown promising results for histological and MRM middle-ear images. For the same initial contours, a larger number of iterations is required for the parametric contours to converge to the boundary than with the discrete dynamic contours, when the gradient is used as the external force. However, when using GVF and gradient plus pressure, parametric active contours require a smaller number of iterations for active contour convergence, compared with the discrete dynamic approach.

The use of open contours was demonstrated for shared boundaries and thin structures, in addition to the usual closed contours.

To my parents, Doretta and Parviz

RÉSUMÉ

La segmentation d'image ou l'extraction des contours d'un objet est une des tâches les plus importantes en traitement des images. Les contours actifs font partie des techniques de haut niveau fréquemment utilisées dans ce but.

Dans ce mémoire, nous comparons deux approches différentes, fondées sur les contours actifs, pour la segmentation de l'oreille moyenne dans des données d'images histologiques et des données d'images de microscopie par resonance magnétique.

Ces deux approches, les contours actifs paramétriques et les contours dynamiques discrets, fournissent des résultats semblables. Différentes forces externes sont testées : le gradient, le flux de vecteurs gradients et le gradient plus la pression. L'utilisation du gradient ne permet qu'une plage de capture limitée. Pour remédier à ce défaut, nous comparons les approches utilisant le flux de vecteurs gradients et utilisant la pression. Bien que le flux de vecteurs gradients possède une bonne plage de capture, il est peu efficace sur les contours de faible contraste et sur les contours d'objets voisins. Au contraire, les forces de pression fournissent des résultats encourageants sur nos images histologiques et nos images de microscopie par résonance magnétique de l'oreille moyenne. Étant donnés des contours initiaux identiques, un plus grand nombre d'itérations est requis pour les contours actifs paramétriques par rapport aux contours dynamiques discrets, dans le cas où le gradient est utilisé comme force. Dans le cas où le flux de vecteurs gradients ou le gradient plus la pression sont utilisés, on observe le phénomène inverse.

Pour la segmentation des structures fines et des contours communs à deux structures, nous utilisons des contours ouverts, et dans les autres cas, nous utilisons des contours fermés.

Table of contents

LIST OF FIGURES

Figure **Page**

1. INTRODUCTION

Image processing and computer vision have become important within the past two decades. Image processing and computer vision techniques aim to improve the pictorial information for human interpretation and the processing of image data for autonomous machine perception.

Image segmentation, or identification of the boundaries of objects in images, is one of the most important problems in computer vision and image processing. The applications of segmentation techniques range from medicine (*e.g.*, locating a lesion) to industry (*e.g.*, robotic vision) and the military (*e.g.*, target detection). With medical imaging playing an increasingly prominent role in the diagnosis and treatment of disease, segmentation techniques have been applied for extracting clinically useful information about anatomic structures through modalities such as x-ray CT (Computer Tomography), MRI (Magnetic Resonance Imaging), PET (Position Emission Tomography), ultrasound and other modalities (Stytz *et al.*, 1991; Ayache, 1995; Bizais *et al.*, 1995; McInerney & Terzopoulos, 1996; Yezzi *et al.*, 1997). The segmentation of the images can be performed either manually or using image processing and computer vision techniques. In manual segmentation a skilled operator, using a computer mouse or track ball, traces the structures of interest on each slice of the image dataset. Manual segmentation benefits from anatomical knowledge that the user employs for boundary identification of the objects. Manual segmentation, however, suffers from several drawbacks, such as the difficulty in achieving reproducible results, and the facts that it is time consuming and open to operator bias (McInerney & Terzopoulos, 1996). The computer-vision based type of segmentation relies more on the mathematical concepts that are applied for boundary identification.

A wide variety of mathematical and computational approaches has been proposed for solving segmentation problems. Segmentation techniques are classified into low-level and high-level techniques. Low-level techniques, such as traditional edge detectors, region growing and mathematical morphology, use only image information. These techniques are generally computationally fast and may be simple but they require a considerable amount of expert interactive guidance. Furthermore, automating these model-free approaches is difficult because of the shape complexity and variability within and across anatomical

structures. In general, the under-constrained nature of the segmentation problem limits the efficacy of approaches that consider local information only. Noise and other image artefacts can cause incorrect regions or boundary discontinuities in objects identified by these methods (McInerney & Terzopoulos, 1996).

Active contours, or snakes (Kass *et al.*, 1986), are high-level techniques that overcome many of the limitations of low-level image-processing techniques by applying information about the boundaries as part of an optimisation procedure. Active contours are energy-minimising contours that are generally controlled by two energy terms. There are force terms associated with the energy terms, *i.e.*, internal and external forces. The internal force is computed based on the local shape of the contour and preserves the smoothness of the contour. The external energy that drives the active contour to the boundary is based on the image information (*e.g.*, any conventional edge detection technique). The interactions of the force terms cause the active contour to evolve from an initial position (*e.g.*, drawn by the user with a mouse) and it converges to the optimal position, *i.e.*, on the structure boundary, where forces balance one another.

Active contours are used extensively for segmentation and a number of alternative approaches have been proposed, such as geometric deformable models (Miller, 1990), geometric active contours (Caselles *et al.*, 1995 & Malladi *et al.*, 1995) and discrete dynamic contours (Lobregt & Viergever, 1995). Active contours have been applied to many medical imaging modalities, including x-ray, angiography, MRI, PET and ultrasound (McInerney & Terzopoulos, 1996). "Active contours have been used to segment, visualise, track and quantify a variety of anatomic structures, ranging from macroscopic to microscopic scale, including the brain, heart, face, kidney, lungs, stomach, liver, skull, vertebra, arteries, neurons, chromosomes and objects such as brain tumours and a foetus" (McInerney & Terzopoulos, 1996). For instance, Leymaire (1990) used active contours for tracking the deformation and locomotion of cells in the plane. Cohen (1991) used them on a set of successive cross sections of ultrasound and magnetic resonance images, leading to 3D object reconstruction. Ranganath (1995) utilised active contours for contour extraction in cardiac MRI images. Lobregt & Viergever (1995) used active contours for segmentation of femurs from x-ray CT images, brain tumours from MRI images and blood vessels from ultrasound images. Davatzikos *et al.* (1995) & Atkins *et al.* (1998) used active contours for segmentation of the brain from MRI images. Wang *et al.* (1998) applied active contours for segmentation

of MRI images of heart ventricles, blood vessels, brain, and bone. Valdés *et al.* (2000) performed trachea segmentation for the respiratory system using CT images.

Because of the extensive successful applications of active contours in the medical imaging field, we were interested in applying active contours for segmentation of middle-ear images. Middle-ear images include tiny objects such as ligaments, muscles and bones that require precise boundary identification prior to labelling, registration, 3-D reconstruction and modelling. We used grey-scaled histological and magnetic resonance microscopy (MRM) datasets of the middle ear for our experiments. Two different active-contour approaches were used and compared for segmentation of middle-ear images. These techniques include the traditional active contours (parametric contours) (Kass *et al.*, 1986) and discrete dynamic contours (Lobregt & Viergever, 1995). The reason we chose discrete dynamic contours was because of the simplicity of the implementation, and the fact that Lobregt & Viergever (1995) claimed that the discrete dynamic contours do not exhibit problems such as shrinking and clustering that exist with traditional parametric contours. We were interested to test the discrete dynamic contours with our datasets. Although the gradient is usually a successful method when used for computation of external forces, it has a limited capture range. We compared the gradient with two other techniques, *i.e.*, pressure (or balloon) force (Cohen, 1991) and gradient vector flow (Xu & Prince, 1997), that have been suggested to improve the capture range of the gradient. We did experiments with both closed and open contours. In closed contours the starting and ending vertices are connected while in open contours they are unconnected.

Chapter Two introduces conventional edge-detection and edge-enhancement techniques as low-level techniques. Some of these techniques are referred to in the later chapters. In Chapter Three, a history of active contours is followed by the definition and discussion of some different approaches. Chapter Four describes the implementation of the two active-contour approaches, *i.e.*, parametric and discrete dynamic, that we used for our experiments. Chapter Five introduces the materials, *i.e.*, the image datasets, that we used for our experiments; and Chapter Six presents the results of the comparisons of the active-contour approaches, using gradient, gradient vector flow and pressure force, on the image datasets. The conclusions and future work are discussed in Chapter Seven.

2. CONVENTIONAL EDGE-DETECTION AND EDGE-ENHANCEMENT TECHNIQUES

In digital image processing, a grey-level image is considered as a 2-D array, *e.g.*, $I(x, y)$, composed of picture elements (pixels), each having a magnitude and a spatial location represented by x and y coordinate components. An edge is defined by changes of grey level between neighbouring pixels. Depending of the rate of change (abrupt or slow) of grey level between neighbouring pixels, edges can be called strong or weak.

Edge detectors are applied to an image to discriminate the edge pixels from non-edge pixels. The result will be an edge map, which gives the data for tracing the boundaries of the regions of interest in an image. Edge enhancers are used to intensify the edges but they do not discriminate between edge and non-edge pixels. The gradient and the Laplacian of Gaussian (LOG) operators are edge enhancers, and thresholding and the Canny operator are edge detectors.

In the following sections we review these techniques. Some of them are later used in Chapter Three for active contours.

2.1 Gradient

Using partial derivatives, the gradient of a scalar field (a function) generates a field of vectors. The gradient of an image gives the rates of change of grey level per unit distance in the directions of the coordinate axes. The gradient of the image is written as:

$$\nabla I = \frac{\partial I}{\partial x}\mathbf{x} + \frac{\partial I}{\partial y}\mathbf{y} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \qquad (2.1)$$

where $G_x = \frac{\partial I}{\partial x}$ and $G_y = \frac{\partial I}{\partial y}$ are the partial derivatives of the image with respect to the coordinate components in the directions of the coordinates axes. The magnitude of the gradient is an important quantity in edge detection. It can be computed as:

$$mag(\nabla I) = [G^2{}_x + G^2{}_y]^{1/2}. \qquad (2.2)$$

The direction of the gradient vector is another important quantity. This can be represented as:

$$\theta(\nabla I) = \tan^{-1}\frac{G_y}{G_x} \qquad (2.3)$$

where the angle is measured with respect to the x axis at each location.

The gradient has three properties that are of concern in image processing. First, if there is little or no change of grey level in neighbouring pixels (*e.g.,* homogeneous regions) the gradient for that location will be zero or a small value close to zero. Second, at a location on an edge (boundary) where an abrupt change of grey level exists, the gradient of the image at that location will have a large magnitude depending on the rate of grey-level change. Third, according to the gradient theorem (Kreyszig, 1993), the gradient of an image at a certain point (*e.g.,* pixel) on a constant surface (*e.g.,* an edge) is a vector normal to that surface at that point.

In digital image processing, the gradient is computed using spatial filters that are

convolved with the images. The filters are implemented to act as partial derivatives, in the form of pairs of masks, at every pixel location. Each pair is used to calculate the partial derivatives in two orthogonal directions. The mask is centred and superimposed on each pixel, covering some neighbouring pixels depending on its size. The products of mask weightings and the corresponding pixels are then summed up and the result is considered as the intensity of the pixel corresponding to the centre of the mask. The filters have the property that the sum of the weights is equal to zero, so they satisfy the gradient property of giving a zero value for homogeneous regions. Below, examples of such masks are provided:

$$\begin{bmatrix} -1 & 0 & +1 \end{bmatrix} \qquad \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix}$$

Horizontal and vertical gradient operators

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Diagonal pair Roberts operators

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Horizontal and vertical Sobel operators

Figure 2.1 shows an image. Figure 2.2 shows the image filtered by the above horizontal and vertical gradient operator, Robert operators and Sobel operator.

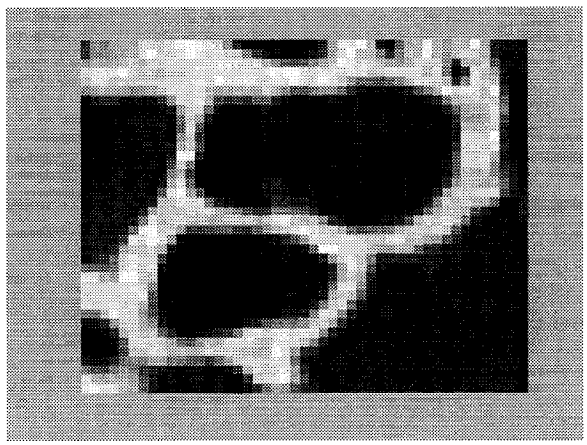


Figure 2.1: An MRM image used for testing gradient operators.

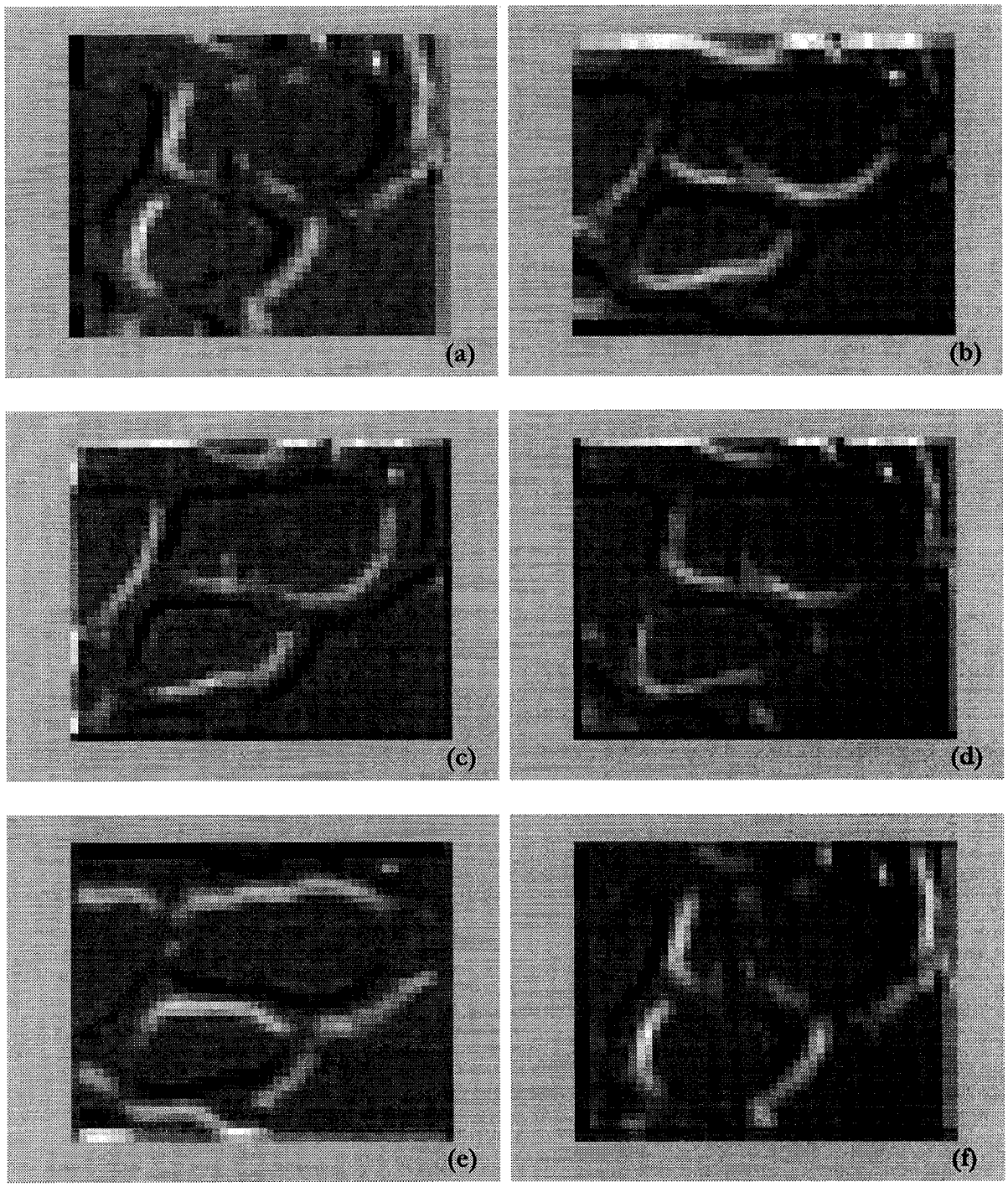


Figure 2.2: Examples of spatial filtering using gradient operators on the image from Figure 2.1: (a) & (b) by horizontal and vertical gradient operators, (c) & (d) by a diagonal pair Roberts operator, and (e) & (f) by horizontal and vertical Sobel operators, respectively.

Figure 2.3 shows an example of the magnitude of the gradient produced by applying Sobel operators to a given image (Figure 2.1).

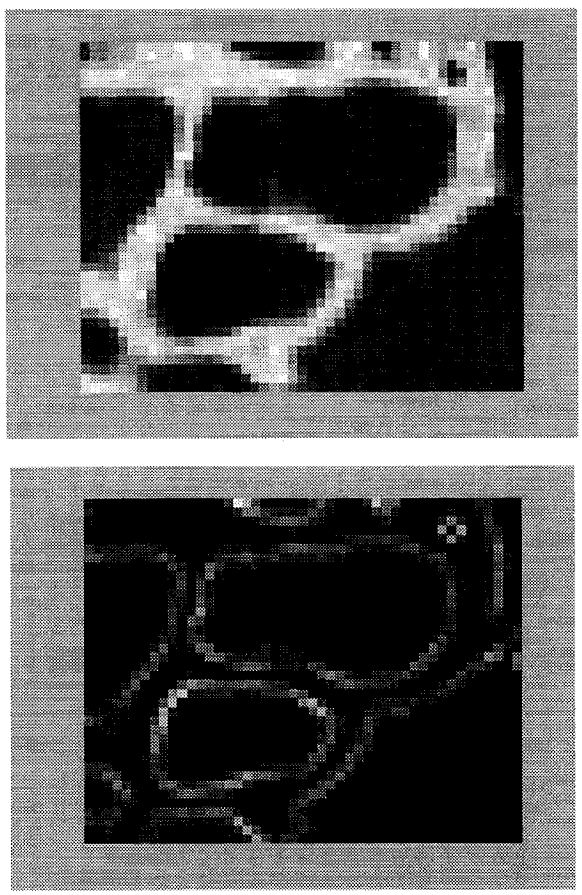


Figure 2.3: Example of the magnitude of the gradient of an image: original image (top), and the magnitude of the gradient of the image (bottom).

2.2 Laplacian Operator

Similar to the gradient operators, the Laplacian operator is another edge enhancer based on the derivatives of the image. The gradient operator uses the first spatial derivative of an image, while the Laplacian operator (∇^2) is based on the second derivative of an image:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2}\mathbf{x} + \frac{\partial^2 I}{\partial y^2}\mathbf{y}. \qquad (2.4)$$

As in the gradient operator, the Laplacian ($\nabla^2 I$) computes the partial derivatives of image values in each location (pixel) with respect to the x and y coordinate axes.

Noise contains a wide range of frequencies: low and high frequencies. The derivative-based operators (the gradient and Laplacian operators) enhance high frequencies that can be edge points or high frequency noise. Intensifying noise is an undesired result. To deal with this issue, Marr & Hildreth (1980) proposed the Laplacian of Gaussian (LOG) operator. This operator is a combination of a 2-D Gaussian kernel and the Laplacian. The 2-D Gaussian (bell-shaped) kernel has a smoothing effect on the image since it is a low-pass filter and has the characteristic of averaging among the neighbouring pixels within an image. By smoothing an image, the Gaussian removes noise. Then, the Laplacian (*i.e.*, the second derivative) will high-pass filter the image and intensify edges. However, this technique does not guarantee that the noise is completely removed and that only the edges are intensified. Figure 2.4 shows a Laplacian of Gaussian kernel.

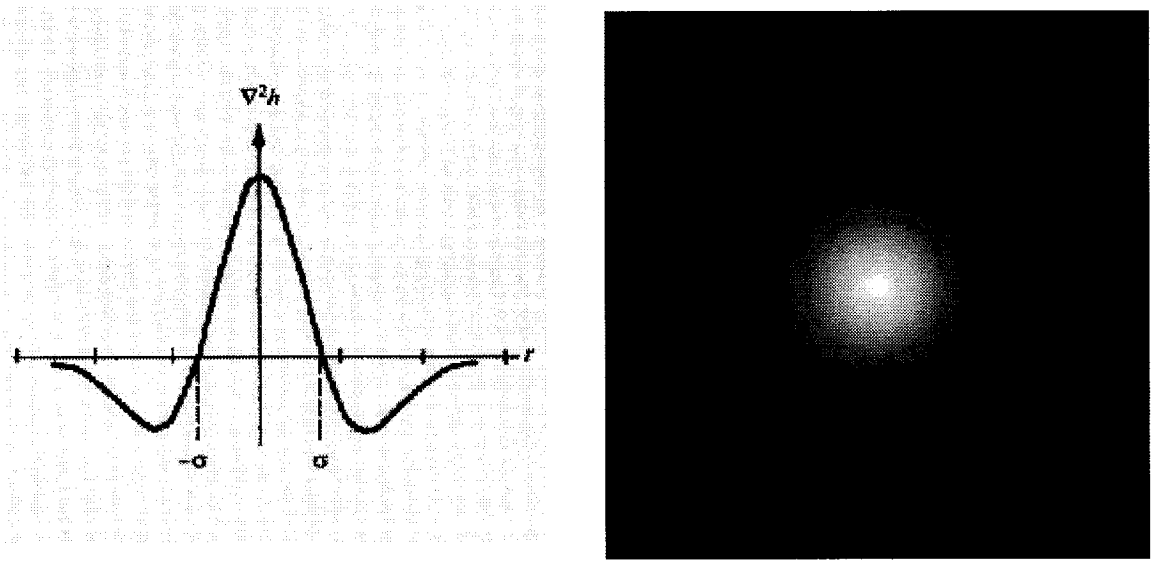


Figure 2.4: Laplacian of Gaussian function: (right) cross section, (left) intensity function image.

The LOG operator can be defined as:

$$\nabla^2 h(x,y) = \frac{(x^2+y^2)}{4\sigma^4} \cdot \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

where (2.6)

$$h(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

where σ is the standard deviation of a 2D Gaussian kernel, *i.e.*, $h(x,y)$, and $\nabla^2 h(x,y)$ represents the Laplacian of the Gaussian kernel. The Laplacian of Gaussian can be implemented as a spatial filter that is convolved with an image. As seen in Figure 2.5, the spatial filter must be designed to assign a high positive weighting in the centre and negative values, with a smaller absolute value than the centre weighting, surrounding the centre weighting, and zero for other weightings, for example:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

LOG operators

The standard deviation of the Gaussian kernel determines the smoothness of the LOG filter. By increasing the standard deviation the filtered image becomes smoother. Choosing the value for σ depends on the amount of noise in the image. Figure 2.5 illustrates the result of an LOG operator on an image.

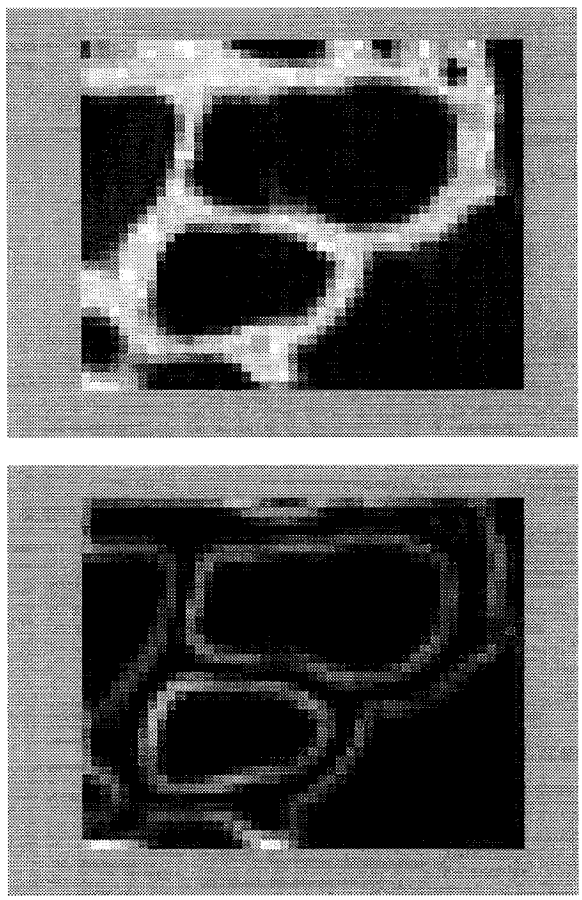


Figure 2.5: Example of an image filtered by an LOG operator: original image (top), the results of applying an LOG operator on the image (bottom).

2.3 Thresholding

Thresholding is a simple technique for edge detection and image segmentation. In this technique, first a threshold value (T) is chosen in the grey-level range of the image and pixels over and under the threshold value are marked differently. Thresholding can be defined as follows:

$$\begin{aligned} I(x,y) > T &\rightarrow I_T(x,y) = 1; \\ I(x,y) \leq T &\rightarrow I_T(x,y) = 0; \end{aligned} \tag{2.6}$$

where $I_T(x,y)$ is the image after thresholding. Hence, all pixels with intensities larger than T are set to a certain number, *e.g.*, 1, and the rest are set to zero. Consequently, the result of thresholding is a binary image.

Thresholding can be single-level, two-level or multi-level. In single-level thresholding only a single thresholding value is chosen, as described above. In the two-level method, two threshold values, *i.e.*, low (T_{low}) and high (T_{high}) threshold values are chosen. Pixels with intensities lying in this range are considered as edge points and are set to a certain value, *e.g.*, 1, and the rest will be set to 0. Two-level thresholding can be represented as follows:

$$\begin{aligned} &I(x,y) > T_{high} \rightarrow I_T(x,y) = 0; \\ &T_{low} \leq I(x,y) \leq T_{high} \rightarrow I_T(x,y) = 1; \\ &I(x,y) < T_{low} \rightarrow I_T(x,y) = 0. \end{aligned} \tag{2.7}$$

Choosing thresholds values is very important since edge points having intensities even slightly lower or higher than a low or high threshold may be excluded (set to zero) , which results in an edge with missing parts. The threshold values are not easily chosen, and they are mostly found based on trial and error. Figure 2.6 shows examples of single-level and two-level thresholding. Thresholding does not always give a good result. An example of edges with missing parts is shown in Figure 2.6 (bottom).

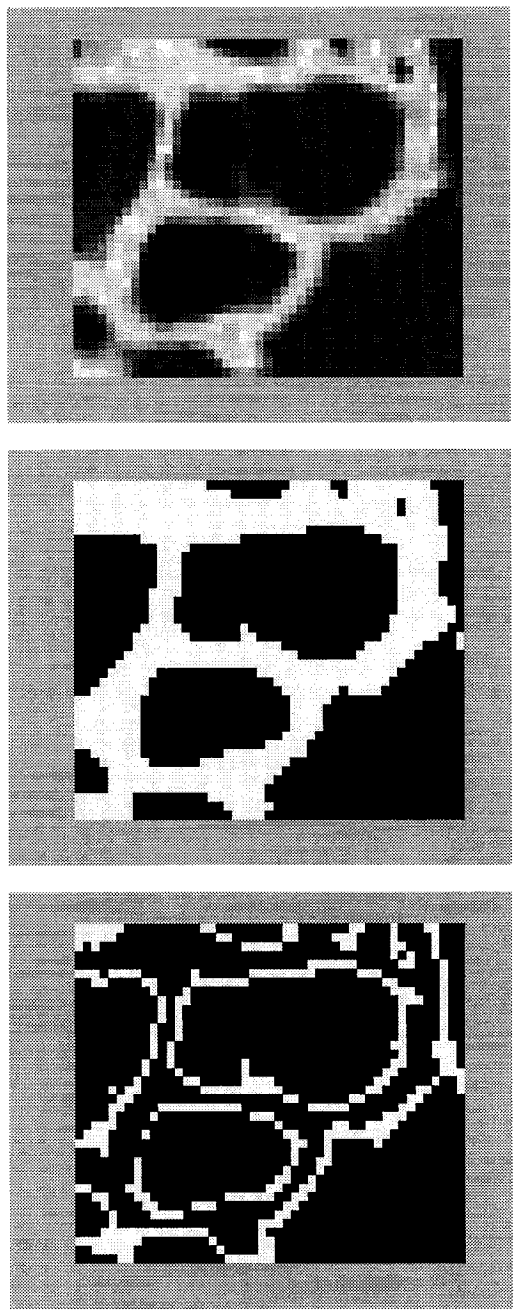


Figure 2.6: Examples of single-level and two-level thresholding: original image (top), and the image after single-level (middle) and two-level (bottom) thresholding. Edges missing parts are seen in the bottom image.

2.4 Canny Operator

It is important that edges that occur in the image should not be missed, and that the edge points should be well localised. That is, the distance between the points marked by the detector and the "centre" of the true edge should be minimized (Canny, 1986).

Low-level techniques such as gradient, thresholding and LOG do not always give good results, especially in case of noisy images or images with a wide range of pixel intensities on edges. Edges with parts missing often happen with edges having a wide range of intensities when using low-level edge-detection techniques.

Canny (1986) proposed a more powerful edge detector than the above-mentioned low-level edge detectors. One aim of this operator is to provide the points marked as edge points that are the centre of the true edge (Canny, 1986).

The Canny operator consists of three processes. First, a low-level edge-enhancer is applied for noise reduction and edge enhancement. Second, two-level thresholding is used to discriminate between edge and non-edge pixels in order to produce a continuous edge outline. Afterwards, the edge-thinning process can be applied to reduce the width of edges to one pixel that is at the centre of the edge.

Since in any image some amount of noise may exist, it is very important to reduce the noise. To do this, an LOG filter is applied to the image for noise reduction. The LOG filter also intensifies the edge points. The next step is an adaptive two-level thresholding process with hysterisis. The purpose of using hysteresis is to ensure that edges are not broken up into multiple edge fragments.

The final step is the edge-thinning process. Based on the gradient magnitude of the thresholded image, the edge-thinning process searches for the edge points in the direction of the gradient, that is, normal to the edge at the edge points. Then, in this process, the centre pixel or pixels of the edge in the direction of the gradient are set to 1 and the rest of the pixels to zero. Consequently, the result will be a binary image containing the edges or the outline of the regions. Figure 2.7 shows an edge map produced by the Canny operator.

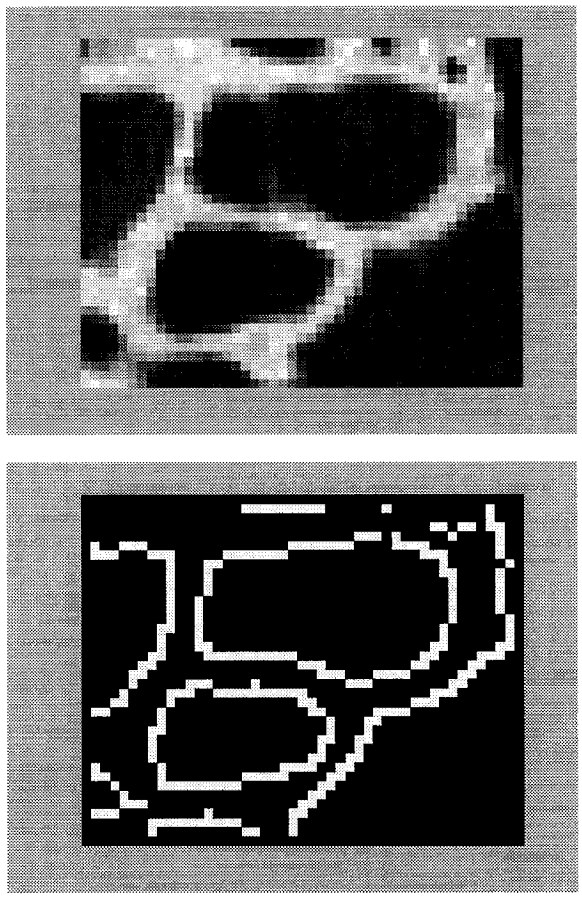


Figure 2.7: Example of edge map using the Canny operator: the original image (top), the result of the Canny operator on the image (bottom).

3. ACTIVE CONTOURS

3.1 Introduction and history

Low-level segmentation techniques serve to simply analyse the image by drastically reducing the amount of data to be processed. However, by reducing the data some necessary image information may be lost. In addition, given a large amount of noise or low contrast, the image information by itself may not be sufficient to result in a successful segmentation. As an alternative, high-level techniques are used, although they are more sophisticated and computationally more expensive than the low-level techniques. Active contours, also known as snakes or deformable contour models (Kass *et al.*, 1986), have proven to be an effective method in line and edge detection, segmentation, shape modelling and motion tracking (Kass *et al.*, 1986).

Active contours were originally proposed by Kass *et al.* (1986). Active contours are an example of a general technique of matching a deformable model to an image boundary by means of energy minimisation (Kass *et al.*, 1986). Energy minimising models have a rich history in computer vision going back at least to the 1970s (McInerney & Terzopoulos, 1996):

> The name "deformable models" stems primarily from the use of elasticity theory at the physical level, generally within a Lagrangian dynamics setting. The physical interpretation views deformable models as elastic bodies which respond naturally to applied forces and constraints. In the Lagrangian setting, the deformation energy gives rise to elastic forces internal to the model. Taking a physics-based view of classical optimal approximation, external potential energy functions are defined in terms of the data of interest to which the model is to be fitted. These potential energies give rise to external forces, which deform the model such that it fits the data. The mathematical foundations of deformable models represent the confluence of geometry, physics, and approximation theory. Geometry serves to represent object shape, physics imposes constraints on how the shape may vary over space and time, and optimal approximation theory provides the formal underpinnings of mechanics for fitting the models to measured data.

"The notion of snake for active contours was inspired by the way living snakes slither while minimising their energy" (Kass *et al.*, 1986).

The original parametric active contours (Kass *et al.*, 1986) are elastic curves or splines defined in an image domain and have a dynamic behaviour that evolves from an initial position in the image to converge to the boundaries of the objects. The initial position is provided either by interactive action of the user or by a higher-level process (Kass *et al.*, 1986). Based on the Lagrangian formulation of motion, the dynamic behaviour of the elastic curve is associated with the energy functional of the curve. The energy functional is composed of two energy terms, *i.e.*, the internal and the external energy terms. The internal energy term is based on the curve itself. The external energy terms arise from the image information. There are force terms that are associated with the energy terms, *i.e.*, the internal and external forces. The internal force term preserves the smoothness and is composed of two force constraints: tension and rigidity. The external force drives the active contour toward the boundary and is derived from image information, *e.g.*, the gradient of the image. The interaction of the force terms causes the active contour to evolve and the evolution process is terminated when the terms balance each other where the energy functional is minimised (*i.e.*, on the object boundary).

The application of traditional active contours to identify boundaries of the objects is, however, not without limitations. One problem lies with the external forces that have a limited capture range. This is because the external force is computed based on the conventional edge detection techniques (*e.g.*, a gradient operator). The gradient of the image has a limited capture range and the active contour must be located close to the boundary in order to converge (Xu & Prince, 1997). Therefore, if the active contour is not located close enough to the boundary, it may not converge to the boundary. The internal force not only cannot solve this problem, but it also has a shrinking behaviour that may lead the active contour to implode, or at least slows the convergence of the active contour to the boundary (Cohen, 1991; Xu & Prince, 1997). Choosing appropriate parameters for tension and rigidity constraints is very difficult in practical applications. In addition, the traditional parametric contours are computationally expensive due to the way they are implemented (Menet *et al.*, 1990).

Variant implementation methods such as B-snakes (Menet *et al.*, 1990), Fourier snakes (Staib & Duncan, 1992) and finite-element snakes (Cohen & Cohen, 1993) have been

proposed in an effort to improve aspects of the original implementation of parametric approach (*e.g.,* to decrease initialisation sensitivity, to simplify the computation, and to get rid of the internal force parameter). In all these methods a curve is considered in segments. For instance, in B-snakes and finite-element snakes, the curve is segmented into polynomial basis functions, and in the Fourier snakes the curve is segmented into a set of trigonometrical basis functions.

Two other alternative approaches have also been proposed for active contours: geometric deformable models (Miller, 1990) and discrete dynamic contours (Lobregt & Viergever, 1995). Although the names of these approaches may seem different from the parametric approach, they basically use the same principles as the traditional parametric contours. In all these approaches the curve is considered as a set of points or vertices which are connected by straight lines or edge segments. All the active contours have a dynamic behaviour that leads the active contour to evolve from an initial position to a final position (where a boundary is located). The dynamic behaviour of the active contour is based on the interaction of a set of constraints or forces that is intended to minimise a cost function, energy functional or a force term. There are constraint terms, comparable to the internal force, that are defined based on the active contour and are responsible for preserving its shape and smoothness. Also, there is an image constraint which is usually based on the gradient of the image. The major differences among all these active-contour methods are in the way the internal forces, or the equivalent constraint terms, are viewed. For instance, in the traditional parametric contours the components of the internal force, the tension and rigidity, are derived from the locations of and the distances between the vertices of the contour. In geometric deformable models (Miller, 1990) the internal constraints are based on the angles between the adjoining edge segments and their lengths, which are again derived from the locations of the neighbouring vertices. In discrete dynamic contours (Lobregt & Viergever, 1995) the internal force is based on the angles between adjoining edges and the normal vectors to the vertices, which are once more computed based on the locations of vertices.

"Since in the active-contour approaches the external force is based on the gradient of the image, these active contours still suffer from a limited capture range for the external force. Also, when an image has a complex background, the active contour may get confused and finding the correct object boundary from the gradient magnitude only is not easy" (Xu &

Prince, 1997). Various methods such as pressure force or balloon (Cohen, 1991), attraction potential forces (Cohen & Cohen, 1993) and gradient flow vector (GVF) (Xu & Prince, 1997) have been suggested to significantly increase the external force capture range. These methods are described in the following sections.

As a different approach, geometric active contours (Caselles *et al.*, 1995; Malladi *et al.*, 1995) have been proposed based on the curve-evolution theory and the level-set method. Based on this theory a curve (active contour) is shrinking in perimeter in order to minimise its length. The minimisation of the curve is driven by the interaction of functions defined based on the curve and a function computed based on the gradient of the image. Using the level-set method makes it possible for the active contours to simultaneously converge to several objects.

In section 3.2, we only review and discuss the traditional parametric contours, the internal and external forces and the constraints that are applied to improve the capture range of the gradient. In sections 3.3 to 3.5, the geometric deformable models by Miller (1990), geometric active contours by Caselles *et al.*, (1995) & Malladi *et al.*, (1995), and discrete dynamic contours (Lobregt & Viergever, 1995) will be presented in more detail. We used the last approach for our experiments that are reported in Chapter 6 and it is discussed in more detail.

3.2 Parametric contours

In the parametric approach, an active contour is represented as a curve or spline, *i.e.*, $\mathrm{v}(s) = (x(s), y(s))$, including vertices. x and y represent the coordinates of the vertices and are functions of the normalised arc length $0 \le s \le 1$. The active contour has a dynamic behaviour that deforms from an initial position and hopefully converges to the boundary of the object. An energy functional (*i.e.*, E^*_{snake}) composed of energy terms defines this behaviour:

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))ds = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))ds \qquad (3.1)$$

where E_{int} and E_{ext} denote the internal and external energy terms associated with the active contour, respectively. The internal energy term is computed based on the local shape of the contour and preserves the continuity and the smoothness of the active contour. The external energy term is computed based on the image information and it drives the active contour to the boundary. In the parametric approach, the general-form energy functional of the active contour, *i.e.*, Equation 3.1, can be represented as:

$$E^*_{snake} = \int_0^1 [\tfrac{1}{2}(\alpha \mid \mathbf{v}'(s) \mid^2 + \beta \mid \mathbf{v}''(s) \mid^2) + E_{ext}(\mathbf{v}(s)]ds\,. \qquad (3.2)$$

Parameters α and β are the coefficients of the internal energy term and represent tension and rigidity, respectively.

The deformation process of the active contour is driven by minimisation of the energy functional. An active contour that minimises the energy functional (Equation 3.2) must satisfy the Euler equation (Kass *et al.*, 1986):

$$-\frac{\partial}{\partial s}(\alpha\frac{\partial \mathbf{v}(s)}{\partial s})+\frac{\partial^2}{\partial s^2}(\beta\frac{\partial^2 \mathbf{v}(s)}{\partial s^2})+\nabla E_{ext}=0$$

or (3.3)

$$\alpha\mathbf{v}''(s)-\beta\mathbf{v}''''(s)-\nabla E_{ext}=0.$$

Equation 3.3 can be also viewed as a force balance so that the energies are associated with the forces (Xu & Prince, 1997):

$$F_{int}+F_{ext}=0$$

where (3.4)

$$F_{int}=\alpha\,\mathrm{v}''(s)-\beta\,\mathrm{v}''''(s)$$
$$F_{ext}=-\nabla E_{ext}.$$

F_{int} and F_{ext} denote the internal and external forces, respectively. In this way the deformation process can be explained based on the interaction of the force terms and the deformation stops when the forces balance each other, *i.e.*, on the location of the boundary.

As suggested in some publications (*e.g.*, Xu & Prince, 1997), we prefer to use the force balance equation to explain the behaviour of the active contour, so it is simpler to understand the concept.

The external force itself can be the image force, using only image information, or may include additional constraints such as a pressure force or a user-defined constraint force. We describe the forces associated with the energy terms in detail in the following sections.

3.2.1 Internal force

The internal energy term and the associated force, *i.e.*, the internal force, preserve the smoothness and continuity of the active contour. According to Equation 3.4, the internal force is composed of the second and forth derivatives of the contour which are weighted by

α and β parameters, respectively. The second-order right-hand term, *i.e.*, $\alpha \mathrm{v}''(s)$, causes the active contour to behave like a membrane to resist stretching, and the fourth-order term, *i.e.*, $\beta \mathrm{v}''''(s)$, causes the active contour to act like a thin plate to resist bending. We refer to α and β as tension and rigidity parameters, respectively. The tension keeps the active contour contracted and the rigidity keeps it smooth.

The active contour has inherently a tendency to shrink, which is related to the tension force. To explain this, the active contour can be considered as an ideal rubber band, with zero initial length and linear behaviour, to represent the tension force. If such a rubber band with any length other than zero (an active contour) is only affected by tension, it shrinks to a point. In the presence of only rigidity, by contrast, the active contour tends to be like a wire and cannot converge to the sharp corners. Consequently, various values for α and β produce different results for an active contour, with the behaviour ranging from a rubber band to a rigid wire. The examples of different combinations of tension and rigidity parameters are provided in Chapter Six, Section 6.2.1.

3.2.2 Image force

While the internal force is responsible for preserving the shape of the active contour, the external force drives the active contour to the boundary. The external force is primarily the image force, computed based on the image itself, and a constraint force can optionally be added. The external force is represented as:

$$F_{ext} = F_{image} + F_{constraint} \qquad (3.5)$$

where F_{image} and $F_{constraint}$ are the image and constraint forces, respectively. One reason for applying a constraint force is the capture-range limitation that the image force may exhibit. The constraint force is used to provide a high-level guidance to the active contour, to expedite the active contour convergence to the boundary and improve the capture range of the image force (Feng & Gelenbe, 1998).

According to Equation 3.4, the external force is the negative of the gradient of the image. In order to intensify the edges, prior to computing the gradient of the image, an edge detector or enhancer can be applied to the image. We compared the performances of the

edge-detection and edge-enhancement techniques in Chapter 2 as the external force for parametric active contours on different regions of an MRM image. These techniques were thresholding, LOG, gradient and Canny operator. The criterion for choosing the edge-detector or edge-enhancement technique was the simplicity of computation. Based on the literature and our own results, we chose the gradient operator as an edge detector. The reasons for this choice are as follows:

- Thresholding is the least sophisticated technique in terms of computation among the mentioned techniques. The result of this technique is a binary image. Therefore, some grey-scale information which may be related to the boundary is lost by binarising an image. Apart from this issue, choosing a proper threshold value is based on trial and error and can differ from one image to another, or from one image slice to another in a single volume. In addition, in some cases, as shown in Figure 2.6, thresholding may result in edges with gaps.

- Although the Canny operator is known as a successful edge detector, this technique has some disadvantages. Firstly, it is the most complicated technique among the above-mentioned edge detectors. Secondly, because of its thresholding process it produces a binary image, which is undesired. And finally, the edge-thinning process in the Canny operator may change the natural shape of the boundaries and may lead the active contour not to precisely identify the boundary.

- Similar to thresholding, choosing the appropriate value for the standard deviation of the LOG operator may vary from one image to another, depending on the noise content and image sharpness. This parameter is found by trial and error. We prefer to use LOG only as a pre-processing technique for enhancement of noisy images.

- The gradient operator is less complicated than the Canny operator, and in contrast to thresholding and LOG, there is no need for choosing any parameter (*e.g.*, threshold or standard deviation).

3.2.3 Constraint force

Using image information, the image force drives the active contour towards the boundary. This force can be computed based on any conventional edge detector and enhancer (*e.g.*, thresholding, LOG, the gradient or Canny operator) to detect the objects' boundaries.

Since the image force computations are generally based on the gradient magnitude of the image, the active contour is attracted to contours with large gradients, *i.e.*, strong edges. A disadvantage of the gradient technique is the limited capture range that it produces for the image force (Cohen, 1991; Xu & Prince, 1997). The limitation exists since the gradient technique results in a noticeable magnitude where there is an abrupt change in grey-scale (*e.g.*, an edge pixel) but inside a homogeneous region the result of the gradient operator is zero or a small value close to zero. As a result, the active contour must be located close to the boundary in order to converge to it.

Kass *et al.* (1986) introduced "springs" and "volcanos" as user-defined constraints that can be applied to an iterating active contour using a users' interface programme. The user-defined constraints are used to push the vertices (selected by the user) relatively close to the boundary so that the active contour can converge to the boundary.

A "spring" is modelled as a linear spring with a fixed point and a vertex to which it is connected, positioned by the user on the image. The user also chooses the vertex. The spring pulls the vertex towards itself with a force proportional to the distance between the spring point and the vertex:

$$\mathrm{F}_{spring,i} = k(s - v_i) \qquad (3.6)$$

where s and v are positions of the spring and the vertex, respectively, and k is a weighting factor.

As a constraint term, a volcano creates a pushing-out force that is applied to all vertices from a point c inside the region of interest. At each vertex this force is inversely proportional to the distance between point c and the vertex. In the case that a vertex is exactly on point c, this force is considered to be zero in order to prevent the division by zero. A volcano is defined as:

$$\mathrm{F}_{volcano,i} = \frac{1}{r}(v_i - c) \qquad (3.7)$$

where v and c are the positions of a vertex and of point c, respectively. r is the distance between point c and the vertex.

A pressure force or balloon model (Cohen, 1991) was proposed as a constraint force to improve the capture range of the gradient. The pressure force is independent of the image information and defines the direction of the active contour deformation by inflating or deflating it. Using this technique, the external force is composed of the image and the pressure forces:

$$F_{ext} = k_{pressure}\mathbf{n}(s) - k\frac{F_{image}}{\left\|F_{image}\right\|} \qquad (3.8)$$

where $k_{pressure}$ is the pressure weighting and its positive or negative sign leads the active contour to inflate and deflate, respectively; $\mathbf{n}(s)$ represents the unit vectors normal to vertices; k is the image force weighting and the image force is the gradient of the image. The normalization to the image force is applied to intensify the boundary, especially when the boundary has low contrast.

If the pressure force acts too strongly, the active contour may overwhelm weak boundaries, especially in the presence of contours with gaps (Xu & Prince, 1997). Therefore, the image force weighting (k) must be larger than the pressure weighting ($k_{pressure}$). This is because the image force should dominate the pressure force and stop the active contour at the location of the boundary; otherwise the pressure force will lead the active contour to pass the boundary.

3.2.4 Alternative image force

Alternative techniques have been proposed to improve the capture range of the image force. An attraction potential force (Cohen & Cohen, 1993) was proposed as the external force. It is the negative gradient of a potential function that is computed using a Euclidean distance map which is based on the image information:

$$F_{ext} = -\nabla P(v). \qquad (3.9)$$

The Euclidean distance map can be computed in different ways, for example (Cohen & Cohen, 1993):

$$P(v) = -e^{d(v)^2} \qquad (3.10)$$

where $d(v)$ is the distance between a point v on the image and the nearest image edge pixel that is detected by an edge detector. $P(v)$ is the potential and the closest edge point has the greatest effect at a position v (Cohen & Cohen, 1993). Similar to the image force produced by the gradient technique, the attraction potential forces also point normal to a boundary concavity and do not lead the active contour to converge to the concavity (Xu & Prince, 1997).

3.2.5 Gradient Vector Flow (GVF)

Xu & Prince found that pressure forces may overwhelm subjective contours, *i.e.*, parts of a boundary with weak or zero contrast. They proposed Gradient Vector Flow (GVF) to improve the capture range of the image force, and found that it could be applied to subjective contours without overwhelming them. GVF involves a vector field derived by solving a vector diffusion equation which diffuses the gradient vectors of a grey-level image. The particular advantage of GVF is the ability to move into boundary concavities (Xu & Prince, 1997).

The GVF field is defined to be the vector field $\mathbf{x}(x, y) = [u(x, y), v(x, y)]$ that minimises the energy functional (ε):

$$\varepsilon = \iint \mu(u^2{}_x + u^2{}_y + v^2{}_x + v^2{}_y) + |\nabla f|^2 |\mathbf{x} - \nabla f|^2 \, dxdy$$
$$f(x, y) = -E_{image}(x, y). \tag{3.11}$$

The subscripts represent partial derivatives with respect to x and y. $f(x, y)$ is the image energy term that can be the gradient of the image. μ is a weighting parameter, also called a regularisation (Xu & Prince, 1997) factor. It is applied to manage the trade off between the two terms of the integrand. When $|\nabla f|$ is small (*e.g.*, in homogenous regions), the energy functional is dominated by the first term (sum of squares of the partial derivatives of the vector field) yielding a slowly varying field. This term is called the smoothing term. However, if $|\nabla f|$ is large, the second term dominates the integrand, and the term is minimised when $\mathbf{x} = \nabla f$. In the latter case, Equation 3.11 keeps the $\mathbf{x}$ nearly equal to the gradient of the image energy term when its gradient is large. Choosing μ depends on the amount of noise in the image. The more noise is in the image, the larger μ should be. This is because in the presence of noise the gradient increases and μ should increase in order to control the trade-off between the first term and the second term in Equation 3.11.

The solution for the minimisation of the energy functional (Equation 3.11) is performed using the Euler equations (Xu & Prince, 1997):

$$\mu\nabla^2 u + (u - f_x)(f_x^{\,2} + f_y^{\,2}) = 0$$
$$\mu\nabla^2 v + (v - f_y)(f_x^{\,2} + f_y^{\,2}) = 0 \qquad (3.12)$$

where ∇^2 represents the Laplacian operator. The above pair of equations can be solved by treating u and v as functions of time:

$$u_t(x, y, t) = \mu\nabla^2 u(x, y, t) - [u(x, y, t) - f_x(x, y, t)] \cdot [(f_x(x, y)^2 + f_y(x, y)^2]$$
$$v_t(x, y, t) = \mu\nabla^2 v(x, y, t) - [v(x, y, t) - f_y(x, y, t)] \cdot [(f_x(x, y)^2 + f_y(x, y)^2] \qquad (3.13)$$

The coupled Equations 3.13 can be solved iteratively, in order to compute the GVF vector field. The GVF fields, *i.e.*, u and v, have converged when the value of the left-hand side of Equation 3.13 remains the same between successive iterations. Since the gradient of the image in the homogeneous regions is nearly zero, the second terms of the pair of equations (Equation 3.13) are zero. In such a case, u and v are determined by Laplace's equation using the Laplacian operator (∇^2) and the resulting gradient vector field is interpolated from the region's boundary, reflecting a sort of competition among the boundary vectors (Xu & Prince, 1997). In other words, GVF is the result of mixing the gradient and Laplacian.

The gradient vectors are normal to the boundary surface, but by combining Laplacian and gradient the result is not the normal vectors to the boundary surface. As a result of this, GVF yields vectors that point into boundary concavities, so that the active contour is driven through the concavities. Figures 3.1 & 3.2 show a quasi-oval object and an object with narrow concavities, respectively. The vector fields produced by the gradient and GVF technique are demonstrated in Figures 3.3 & 3.4.

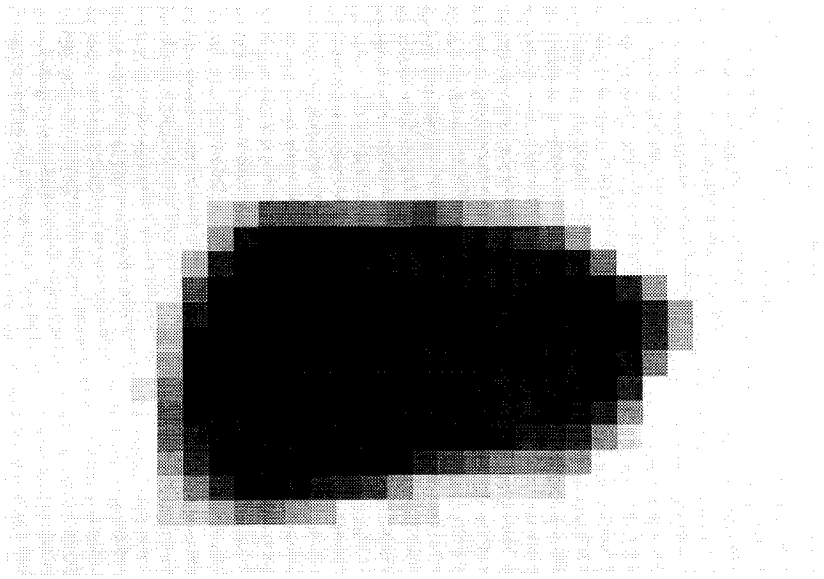


Figure 3.1: An MRM image for testing GVF

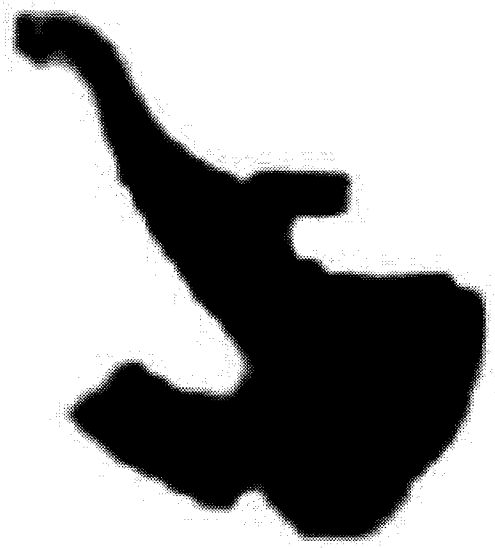


Figure 3.2: An MRM image showing a structure with narrow concavities for testing GVF

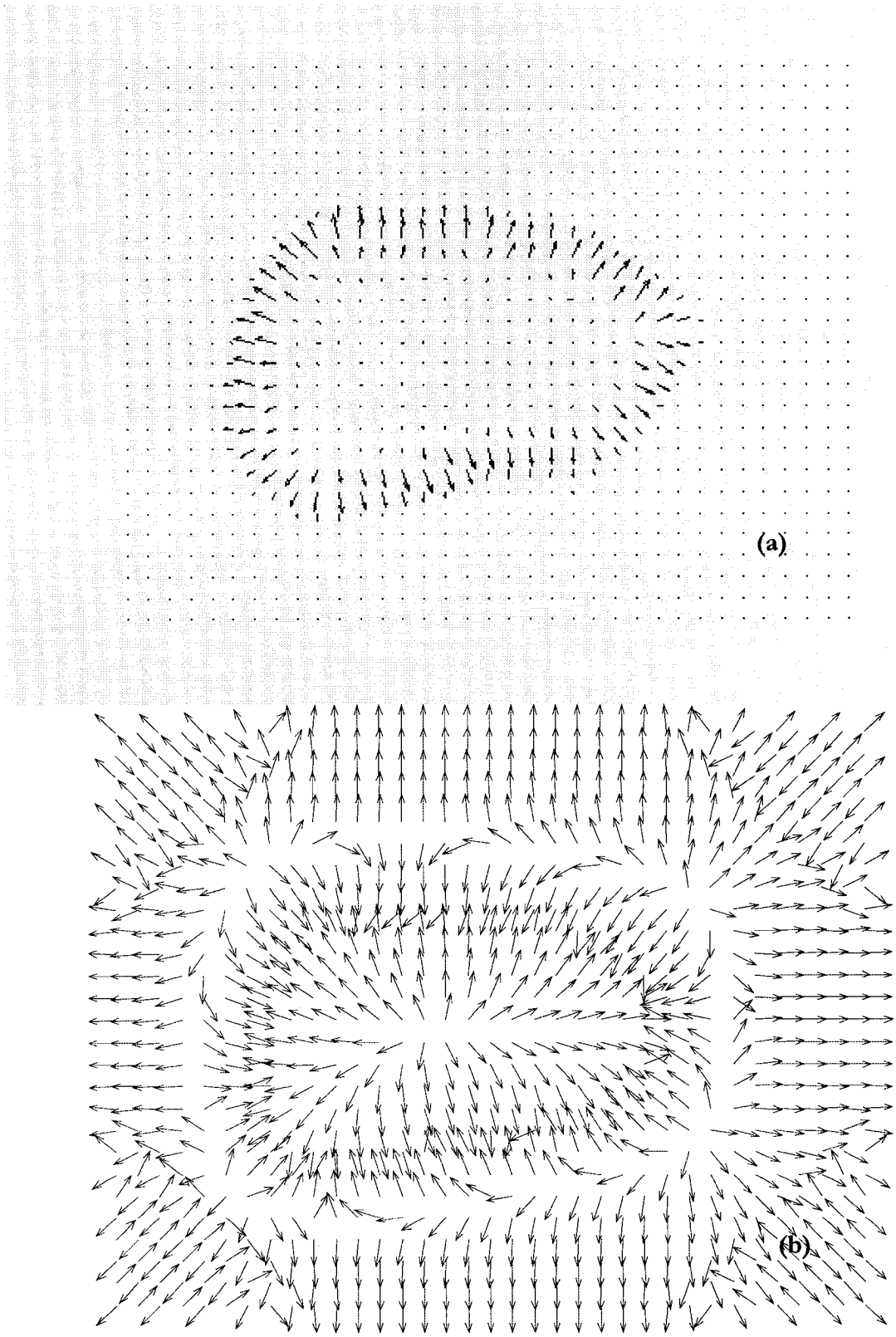


Figure 3.3: Vector fields for object from Figure 3.1: produced by (a) gradient and (b) GVF.

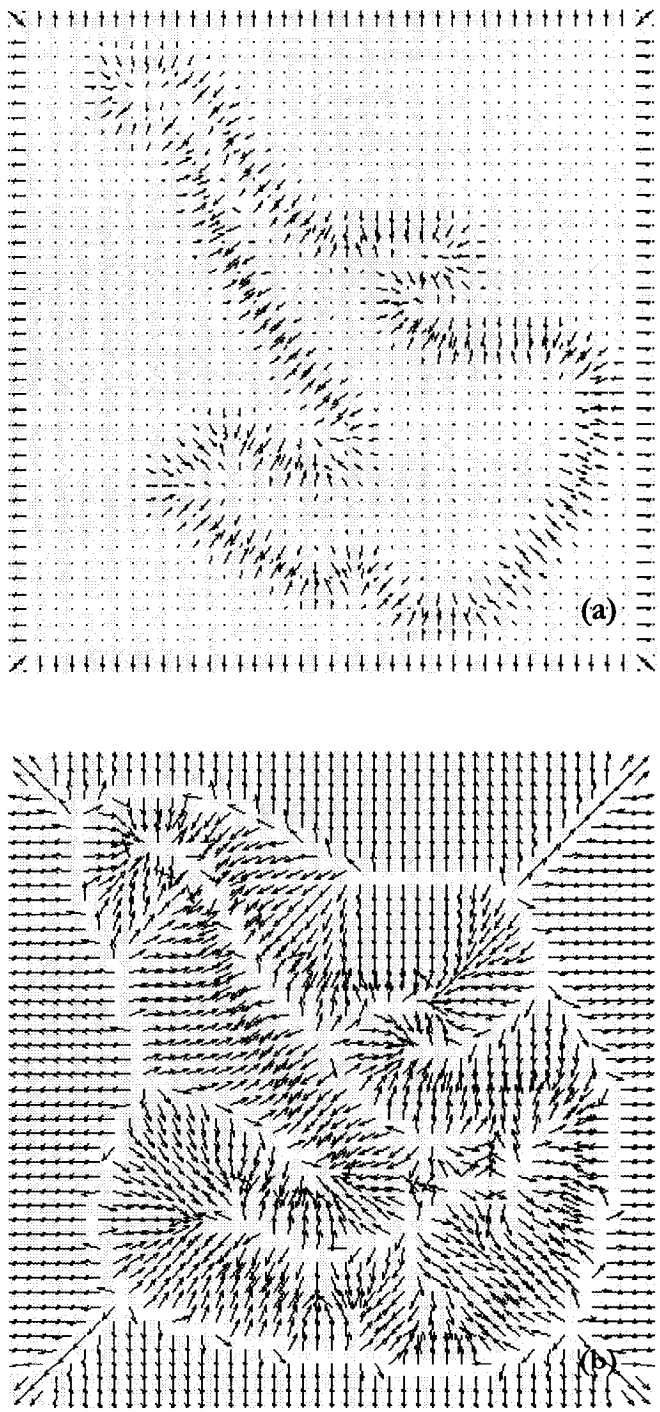


Figure 3.4: Vector fields for object from Figure 3.2: produced by (a) gradient and (b) GVF.

As shown in Figures 3.3 and 3.4, the GVF vectors originate inside the region and point toward the boundary. The boundary is seen where the vectors from inside and outside of the region meet in the opposite direction. Therefore, GVF drives the active contour to the boundary and fastens the active contour on the location of the boundary.

GVF provides a large capture range for the boundary, so the active contour is not required to be located close to the boundary. It also improves the active convergence for the concavities.

3.3 Geometric deformable contours

Miller *et al.* (1990) proposed a geometric deformable model. In their model an active contour is considered as a polygon, consisting of points (vertices) and edge segments connecting the points. As with all active contours, this polygon model deforms until it converges to the boundary of an object. The deformation process is performed by minimising a cost function that defines the displacement of the vertices. The cost function is a sum of constraint terms based on the shape of the polygon shape (angle and distance constraints), the image (image constraints) and pressure (pressure constraint). The constraint equation is defined as follows:

$$C_i(x,y,d,\alpha) = a_0\tau(x,y) + a_1 D(x,y) + a_2\Lambda(d) + a_3\theta(\alpha) \qquad (3.14)$$

where $C_i(x,y,d,\alpha)$ is the cost function associated with the position of the vertices; d and α denote the distance between the two neighbouring vertices and the angle between adjoining edge segments, respectively; $\tau(x,y)$ is the image constraint term; $D(x,y)$ is the the pressure constraint term; $\Lambda(d)$ is the distance constraint term; $\theta(\alpha)$ is the angle constraint term; and a_0, a_1, a_2 & a_3 are weightings. We briefly explain these constraint terms:

The image constraint term, $\tau(x,y)$, is a thresholded image and it can be an edge map produced by any edge detector.

The pressure constraint term, $D(x,y)$, controls the direction of deformation by inflating or deflating the polygon model. This constraint term is a monotonically increasing or decreasing function of the distance between a vertex and a reference point. In the decreasing form the pressure constraint term is represented as:

$$D(x,y) = \ln\frac{M}{|(x,y)-(x_0,y_0)|} \qquad (3.15)$$

where M is the diameter of the image, (x,y) is the position of the vertex and (x_0,y_0) is

the position of a reference point that is assigned inside the object of interest. In this form the pressure constraint term deflates the polygon. In the increasing form the pressure constraint term $\frac{1}{D(x,y)}$ is applied to the constraint equation, and it inflates the polygon.

The distance constraint term, *i.e.*, $\Lambda(d)$, maintains the distribution of the vertices, based on the distance between the neighbouring vertices. To do this, the mean distance (μ) between neighbouring vertices is calculated:

$$\mu = \frac{1}{n}\sum_{i=k-n}^{k-1}\left|P_{i+1}(x,y) - P_i(x,y)\right| + \frac{1}{n}\sum_{i=k+1}^{k+n}\left|P_i(x,y) - P_{i-1}(x,y)\right| \qquad (3.16)$$

where n is the number of neighbours to each side of the vertex k, P_i representing the position of vertex i. Furthermore, the minimum (*i.e.*, $d_{\min}$) and maximum (*i.e.*, $d_{\max}$) distances are defined 20% below and above μ, respectively. $\Lambda(d)$ is defined based on the minimum and maximum distance in such a way that, if the length of each edge segment (d_{edge}) adjoining a vertex is between these two distances, the distance constraint term will be zero. However, if the edge length is smaller than the minimum distance, the distance constraint will be equal to the difference between the minimum distance and the edge length, and for edge lengths larger than the maximum distance the distance constraint will be the difference of the maximum and the edge length.

The angle constraint term, *i.e.*, $\theta(\alpha)$, maintains the shape of the polygon model. First, the initial internal angle (α) between two adjoining edges is calculated for each vertex. The minimum angle (*i.e.*, $\alpha_{\min}$) and maximum angle (*i.e.*, $\alpha_{\max}$) are computed as 15% below and above α. During the deformation process, if an angle is between the maximum and minimum values, the angle constraint is set to zero; otherwise, if the angle is smaller than the minimum parameter the angle constraint is equal to the difference between the angle and the minimum value. For an angle larger than the maximum angle, the angle constraint is set to the difference between the angle and the maximum value.

The deformation process is carried out by minimizing the cost function for each vertex. To minimise the cost function, each vertex moves in the direction of the gradient of the cost function (*i.e.,* $\partial C_i/\partial x$ and $\partial C_i/\partial y$). The minimization process is performed iteratively until the cost function for each vertex reaches zero or a certain small value.

Geometric deformable models are computationally less expensive that traditional active contours (Miller, 1990). In geometric deformable models the constraint equation is evaluated only for the vertex position, not for the trajectory of the connecting edge segments. This makes the model discrete; therefore, the length of the connecting edges defines its resolution (Lobregt & Viergever, 1995).

3.4 Geometric active contours

Geometric models of active contours were proposed by Caselles *et al.* (1993) and Malladi *et al.* (1995). These models are based on the theory of curve evolution and geometric flows. In these active contour models, the curve is propagating (deforming) by means of a velocity that contains two terms, one related to the shape of the curve and the other to the image (Caselles *et al.*, 1995).

Malladi *et al.* (1995) used the level-set method for the geometric active contours. The level-set method was proposed by Osher & Sethian (1988). The curve evolution is implemented by embedding the curve $C(s)$ in a surface function $\Phi(x, y, t)$. Specifically, at $t = 0$, the curve is the level set given by $\Phi(x, y, t = 0) = 0$. The curve evolves as the surface evolves over time. When the evolution of $\Phi(x, y, t)$ stops, for example at $\Phi(x, y, t = T)$, the evolved curve can be obtained from the level set $\Phi(x, y, t = T) = 0$.

By using the level-set method, the geometric active contours have the advantage over other active contours that they can automatically handle topological changes (*e.g.*, the splitting and merging of curves during evolution). Therefore these active contours can simultaneously detect several objects.

Caselles *et al.* (1995) showed that a particular case of the classical energy-minimising active contours is equivalent to finding a geodesic curve (minimal distance path between given points) in a Reimannian space with a metric derived from the image. This geodesic active contour includes a new component in the curve velocity, based on the image information, that improves the geometric active contour model. The new velocity component allows accurate tracking of boundaries even with small gaps.

3.5 Discrete dynamic contours

Discrete dynamic contours were proposed by Lobregt and Viergever (1995), inspired by the geometrically deformable model (Miller *et al.*, 1990). Adopting the basic structure of the model that vertices are connected by edge segments, discrete dynamic contours depend on the distance between a vertex and its neighbours, and the estimation of local curvature.

The dynamic behaviour of the contour model is defined based on a force equation, which is computed for each vertex:

$$\mathrm{F}_{total,i} = w_{ext}\,\mathrm{F}_{ext,i} + w_{int}\,\mathrm{F}_{int,i} + w_{damp}\,\mathrm{F}_{damp,i} \qquad (3.17)$$

where $\mathrm{F}_{total,i}$ is the total force term; $\mathrm{F}_{ext,i}$, $\mathrm{F}_{int,i}$ and $\mathrm{F}_{damp,i}$ are the external, internal and damping force terms, respectively; and w_{ext}, w_{int} and w_{damp} are the external, internal and damping weightings, respectively. The internal force is computed based on the local shape of the contour, the external force is based on the image information, and the damping force is used to improve the stability of the dynamic process of the active contour. First the dynamic behaviour of the active contour is described and later we discuss the force terms.

According to the force equation (Equation 3.17) the interaction of the force terms results in a cost function (*i.e.*, total force) for each vertex. Similar to traditional parametric contours, in this model the deformation process is performed by minimization of the total force for each vertex. The dynamic behaviour of the active contour is determined by computing an acceleration term for each vertex using the total force:

$$\mathbf{a}_i(t) = \frac{1}{m_i}\mathrm{F}_{total,i}(t) \qquad (3.18)$$

where $\mathbf{a}_i$ is the acceleration vector of vertex i, m_i represents the mass of the vertex and is a scalar, and t represent the state of the contour in an iteration. The velocity term and the position of each vertex are then computed based on the acceleration term:

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \mathbf{a}_i(t)\Delta t \qquad (3.19)$$

and

$$p_i(t+\Delta t) = p_i(t) + \mathbf{v}_i(t)\Delta t \qquad (3.20)$$

where $\mathbf{v}_i$ is the velocity vector of vertex i, Δt represents the incremental time between two iterations, and p_i represent the position of the vertex.

In the following sections we discuss the internal, external and damping forces in detail.

3.5.1 Internal force term

In the discrete-dynamic approach, a contour consists of vertices that are connected by straight-line edge segments. The internal force aims to preserve the contour's smoothness. To do this the internal force should minimise the local curvature for each vertex. The measure of the local curvature for the vertex is the angle between the two adjoining edges at each vertex. The local curvature for each vertex is defined as the difference between the directions of the two edge segments that join at that location. Figure 3.5 shows a contour composed of eleven vertices. Each vertex is represented as V_i, and the previous and next neighbouring vertices are represented as V_{i-1} and V_{i+1}, respectively. Figure 3.6 shows two edge segments and the local curvature vector.

The local curvature is defined as

$$\mathrm{c}_i = \hat{\mathrm{d}}_i - \hat{\mathrm{d}}_{i-1} \qquad (3.21)$$

where c_i represents the local curvature at V_i which is located between the adjoining edge segments d_i and d_{i-1}; $\hat{\mathrm{d}}_i$ and $\hat{\mathrm{d}}_{i-1}$ are the unit vectors which represent the directions of the edge segments. The local curvature has a length which depends only on the angle between the adjoining edge segments at the location of the vertex and is not influenced by the lengths of the two adjoining edge segments.

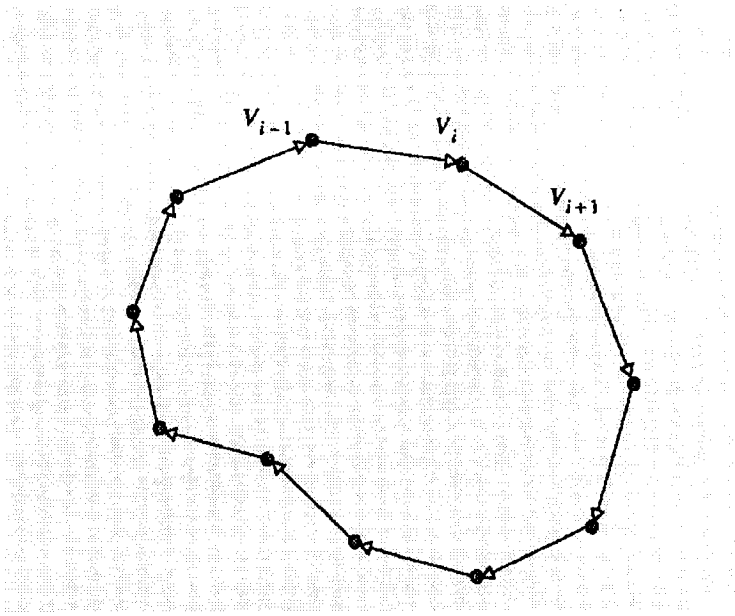


Figure 3.5 A contour consisting of a set of eleven vertices (V) that are connected by edge segments (from Lobregt & Viergever, 1995).

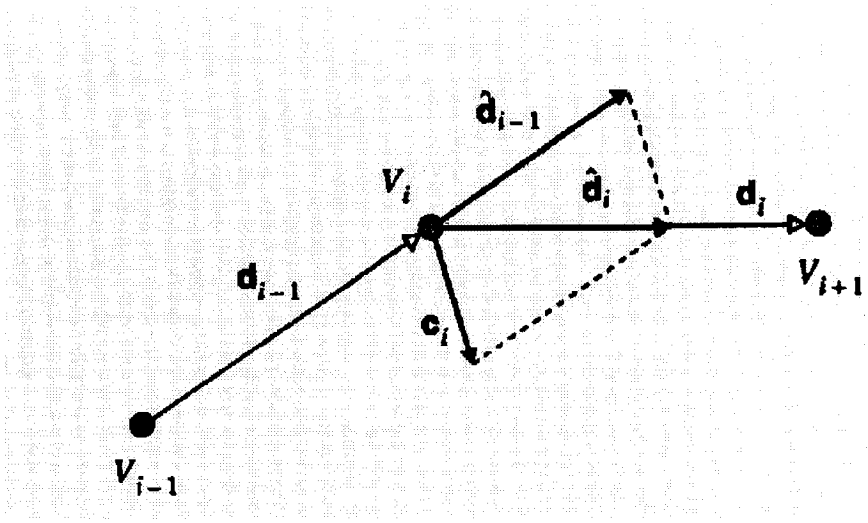


Figure 3.6: Calculation of local curvature. Local curvature (C_i) at the position of a vertex (V_i) is defined as the difference between the directions of the adjoining edges ($\hat{\mathbf{d}}_i$ & $\hat{\mathbf{d}}_{i-1}$) (from Lobregt & Viergever, 1995).

The radial unit vectors are calculated based on the locally tangential unit vectors for each vertex. The tangential unit vector for each vertex is the normalised sum of the unit vectors of the two adjoining edge segments at that location. The radial unit vector for that vertex will be the rotation of the tangential unit vector by 90 degrees. Figure 3.7 shows the tangential ($\hat{t}_i$) and radial ($\hat{r}_i$) unit vectors at the location of vertex V_i.

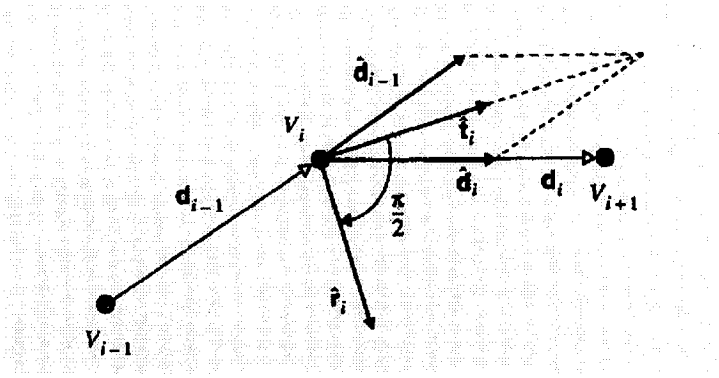


Figure 3.7: Local tangential and radial vectors for a vertex. $\hat{t}_i$, $\hat{r}_i$ and V_i are the local tangential vector, the radial vector and the vertex, respectively (from Lobregt & Viergever, 1995).

Equation 3.22 represents the computation of the locally tangential and radial unit vectors for vertex V_i:

$$\hat{t}_i = \frac{\hat{d}_i + \hat{d}_{i+1}}{\| \hat{d}_i + \hat{d}_{i+1} \|}$$

$$\hat{r}_i = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \hat{t}_i \qquad (3.22)$$

The goal of applying the internal force is to preserve the smoothness of the contour by reducing the local curvature. The calculation of the internal force can be done by considering the interal force for a vertex as only the component of the local curvature vector of the vertex in the direction of the local radial vector, as follows:

$$\mathbf{F}_{int,i} = (\mathbf{c}_i \cdot \hat{\mathbf{r}}_i)\hat{\mathbf{r}}_i . \qquad (3.23)$$

Figure 3.8 shows examples of contours for which the internal force gives a satisfactory result (c) and unsatisfactory results (a & b). The contours are shown on the left-hand side, including the internal force vectors (arrows). On the right side the internal force vectors are shown in r,t coordinates. The vertical (r) and horizontal (t) axes of the coordinates represent the radial and tangential directions, which are perpendicular to each other. As shown in this coordinate system, the internal force vector of each vertex is parallel to the radial unit vector. In contour (a) the internal force vectors attempt to minimise the areas with constant curvature, and this has a shrinking effect and leads the contours to implode to a point. In contour (b) this internal force leads the active contour to accumulate the vertices in the sharp corners. However, the internal force gives a successful result for model (c), minimising the local curvature and straightening the contour.

However, the internal force should reduce the local curvature without affecting areas of constant curvature. For that reason, the lengths of the internal force vectors should be zero for parts of the contour with constant curvature.

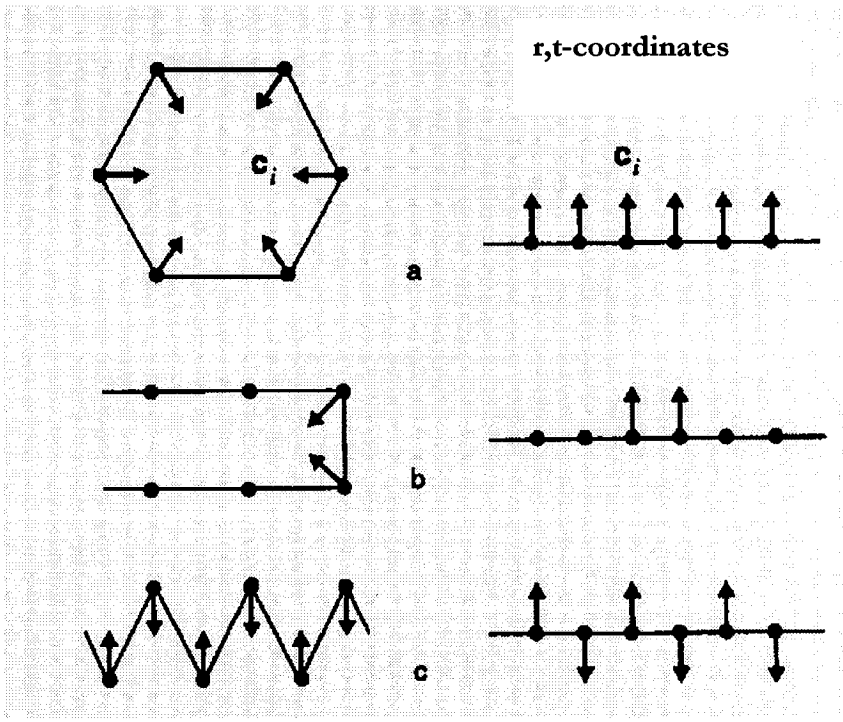


Figure 3.8: Examples of different contours (left). The arrows represent the internal forces. On the right, the internal forces are shown using the locally tangential $\hat{t}_i$ and radial $\hat{r}_i$ coordinates, for each contour (from Lobregt & Viergever, 1995).

In order to prevent the internal force from shrinking areas with constant curvature, one solution (proposed by Lobregt and Viergever) is to convolve the result of Equation 3.23 with a filter (k_i):

$$F_{int,i} = (c_i \cdot \hat{r}_i) * k_i . \qquad (3.24)$$

k_i can be designed in different ways. Lobregt and Viergever used the simple symmetrical filter

$$k_i = \{...,0,0,0,-\tfrac{1}{2},1,-\tfrac{1}{2},0,0,0,...\} \qquad (3.25)$$

so the internal force will be

$$F_{int,i} = [-\tfrac{1}{2}(c_{i-1} \cdot \hat{r}_{i-1}) + (c_i \cdot \hat{r}_i) - \tfrac{1}{2}(c_{i+1} \cdot \hat{r}_{i+1})]\hat{r}_i . \qquad (3.26)$$

In this case the internal force of each vertex is a combination of local curvature of the vertex in its radial direction, combined with half of the internal forces of the neighbouring vertices. The internal force of each vertex is in either the same or the opposite direction as the radial unit vector for the vertex. In this formulation, the local curvature and the direction of the radial unit vectors of the neighbouring vertices affect the internal force of the vertex. The new internal force can solve most of the shrinking effects that exist with the internal force of Equation 3.26. Figure 3.9 shows the internal forces computed with Equation 3.26 for the same contours shown in Figure 3.8. The new internal force computation gives satisfactory results for contours (a), (b) and (c).

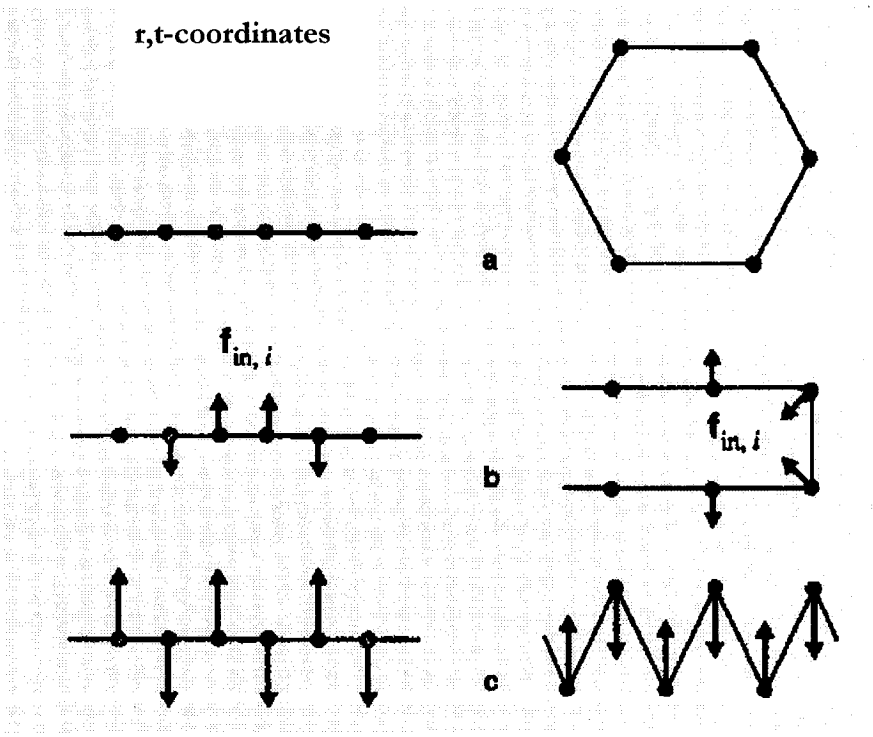


Figure 3.9: Examples of different models (right) that are shown in Figure 3.8. The arrows represent the internal forces. The internal forces are shown in the locally tangential $\hat{t}_i$ and radial $\hat{r}_i$ coordinates for each model (left) (from Lobregt & Viergever, 1995).

For open contours, at the open ends, t_i is defined to be equal to the direction of the first or the last contour segment: $t_1 = d_1$ and $t_n = d_{n-1}$, while the local curvature is set zero for both end positions.

3.5.2 External force term

In discrete-dynamic contours, the external force term is similar to that in other active contours:

$$F_{ext,i} = F_{image,i} + F_{constraint,i} \qquad (3.27)$$

where the image force term can be a gradient of the image, and the external constraint is an optional user-defined external constraint such as springs or volcanoes.

3.5.3 Damping force term

A damping force term is used to improve the stability of the deformation process. This term is used because sometimes the contour may oscillate between two states. For each vertex, the damping force term is a force proportional to the vertex velocity as:

$$F_{damp,i} = -\mathbf{v}_i \qquad (3.28)$$

where $\mathbf{v}_i$ is the velocity of the vertex.

3.5.4 Re-sampling

A re-sampling process is applied to maintain the resolution of the active contour. To do this, a desired distance value between neighbouring vertices, d_{edge}, is defined by the user, then minimum and maximum distances are set:

$$d_{min} = \frac{1}{2} d_{edge}$$
$$d_{max} = \frac{3}{2} d_{edge} \qquad (3.29)$$

In this procedure, if the edge length between two neighbouring vertices is more than d_{max} a new vertex is inserted in between, while two neighbouring vertices are merged if the distance between them is less than d_{min}.

The discrete dynamic contours are conceptually simple and computationally less expensive than the previous active contours, *e.g.*, parametric contours (Kass *et al.*, 1986) and geometric deformable models (Miller *et al.*, 1990). The re-sampling process of the discrete dynamic contour improves the resolution that exists with active contours such as Miller's geometric deformable models. It also reduces the shrinking effect caused by the internal forces of the traditional active contours (Lobregt & Viergever, 1995).

4. IMPLEMENTATION OF ACTIVE CONTOURS

In this chapter we will concentrate on the implementation of two approaches proposed for active contours, *i.e.*, the parametric approach (Kass *et al.*, 1986) and the discrete dynamic approach (Lobregt & Viergever, 1995). We used active contours based on these approaches for our experiments in Chapter Six.

4.1 Parametric contours

In this section we review the discrete (in terms of time and space) implementation for parametric contours which was originally proposed by Kass *et al.* (1986). They applied numerical methods for the implementation of their active contours. A contour consists of n vertices ($\mathbf{v}$) that are connected by straight lines.

In the discrete form, the energy functional of the active contour (Equation 3.1) can be represented as:

$$E^*_{snake} = \sum_{i=1}^{n} [E_{int}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i)]$$

where

$$E_{int} = \alpha_i |\mathbf{v}_i - \mathbf{v}_{i-1}|^2 + \beta_i |\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2 \qquad (4.1)$$

$$E_{ext} = -|\nabla I|$$

where $\mathbf{v}_0 = \mathbf{v}_n$, meaning that the active contour is a closed contour and the first and last vertices are at the same location. The vertices are represented as column vectors, with respect to the coordinate components:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} \quad , \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix}.$$

The corresponding Euler equation for energy minimization of the active contour (Equation 3.3) is represented as (Kass *et al.*, 1986):

$$\begin{aligned} &\alpha_i(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_{i+1}(\mathbf{v}_{i+1} - \mathbf{v}_i) \\ &+ \beta_i[\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i] - 2\beta_i[\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}] + \beta_{i+1}[\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_i] \\ &+ (\frac{\nabla E_{ext}}{\partial x}, \frac{\nabla E_{ext}}{\partial y}) = 0 \end{aligned} \tag{4.2}$$

or, with respect to the coordinate components,

$$\begin{aligned} &\alpha_i(x_i - x_{i-1}) - \alpha_{i+1}(x_{i+1} - x_i) \\ &+ \beta_i[x_{i-2} - 2x_{i-1} + x_i] - 2\beta_i[x_{i-1} - 2x_i + x_{i+1}] + \beta_{i+1}[x_i - 2x_{i+1} + x_i] \\ &+ (\frac{\nabla E_{ext}}{\partial x}) = 0 \\ &\alpha_i(y_i - y_{i-1}) - \alpha_{i+1}(y_{i+1} - y_i) \\ &+ \beta_i[y_{i-2} - 2y_{i-1} + y_i] - 2\beta_i[y_{i-1} - 2y_i + y_{i+1}] + \beta_{i+1}[y_i - 2y_{i+1} + y_i] \\ &+ (\frac{\nabla E_{ext}}{\partial y}) = 0. \end{aligned} \tag{4.3}$$

Equation 4.3 can be written in matrix form as:

$$\begin{aligned} \mathbf{Ax} + \frac{\partial E_{ext}(x,y)}{\partial x} = 0 \\ \mathbf{Ay} + \frac{\partial E_{ext}(x,y)}{\partial y} = 0 \end{aligned} \tag{4.4}$$

where $\mathbf{A}$ is a penta-diagonal banded matrix. The size of this matrix is $n \times n$ (n being the number of vertices). For instance, for an active contour consisting of 10 vertices, matrix $\mathbf{A}$ is as follows:

$$A=\begin{bmatrix} 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\ \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta \end{bmatrix}.$$

To solve Equation 4.4, the right-hand sides of the equations can be set equal to the product of a step size and the negative time derivatives of the left-hand sides (Kass *et al.*, 1986):

$$\begin{aligned} \mathbf{A}\mathbf{x}_t + \frac{\partial E_{ext}(x_{t-1}, y_{t-1})}{\partial x} &= -\gamma(\mathbf{x}_t - \mathbf{x}_{t-1}) \\ \mathbf{A}\mathbf{y}_t + \frac{\partial E_{ext}(x_{t-1}, y_{t-1})}{\partial x} &= -\gamma(\mathbf{y}_t - \mathbf{y}_{t-1}) \end{aligned} \qquad (4.5)$$

where subscript t is the iteration number and γ is the step size. According to Equations 4.6, the balance between the force terms is achieved when the right-hand sides of the equations are zero, which means that the location of the vertices in the successive iterations has not changed. The solution for Equations 4.5 to compute the vertex positions is done by matrix inversion as follows:

$$\mathbf{x}_t = (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{x}_{t-1} - \frac{\partial E_{ext}(x_{t-1}, y_{t-1})}{\partial x})$$
$$\mathbf{y}_t = (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{y}_{t-1} - \frac{\partial E_{ext}(x_{t-1}, y_{t-1})}{\partial y}) \quad (4.6)$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. The $\mathbf{A}$ matrix must be updated after each iteration.

For our experiments with parametric active contours, we used an implementation by Xu & Prince at the Image Analysis and Communications Lab, Johns Hopkins University. The code was written in Matlab (The MathWorks Inc.). Further information about the implementation and the code is provided at http://iacl.ece.jhu.edu/projects.

According to the implementation by Xu & Prince of the parametric active contours, to apply a pressure force to the active contour, as in Equation 3.8, Equation 4.6 can be modified to:

$$\mathbf{x}_t = (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{x}_{t-1} + \kappa_1\mathbf{n}(x_{t-1}) - \kappa\frac{\partial E_{image}(x_{t-1}, y_{t-1})}{\partial x})$$
$$\mathbf{y}_t = (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{y}_{t-1} + \kappa_1\mathbf{n}(y_{t-1}) - \kappa\frac{\partial E_{image}(x_{t-1}, y_{t-1})}{\partial y}) \quad (4.7)$$

where $\mathbf{n}$ represents the normal unit vectors to the vertices, κ is the image force weighting, and κ_1 represents the pressure weighting, which can be a positive or negative value to inflate or deflate the active contour, respectively.

In the implementation by Xu & Prince, a re-sampling process is applied that maintains the resolution of the active contour. After each iteration the distance between two vertices is computed. If it is larger than a defined maximum value (*e.g.*, $1\frac{1}{2}$ pixel) a new vertex is inserted in between, and if the distance is less than a defined minimum value (*e.g.*, $\frac{1}{2}$ pixel) the vertices are merged.

4.5 Discrete dynamic contours

We developed a semi-automatic computer programme, Oxiana, that implements the discrete dynamic contours of Lobregt & Viergever (1995). This programme was initially written by Hamelin, Labonté & Pelletier (1999) in our lab. Oxiana is written in C. It uses GIMP, the GNU Image Manipulation Programme, for graphics and general data structure implementations. The GIMP programme includes the GLADE toolkit, which is used to build the graphical user interface. The GLADE toolkit uses GTK and GDK, multi-platform toolkits for creating graphical user interfaces, and the Glib library, containing libraries for the graphics, and for general data structure implementations. Further information about GIMP and GLADE is available at http://www.gimp.org. Information about GTK, GDK and the Glib library is available at http://developer.gnome.org/doc/API/.

The user can either draw an initial contour using the mouse, or load a previously saved contour. To draw an initial contour using the mouse, the user clicks in the region of interest and Oxiana connects the successive vertices by lines. It connects the first and last points in order to draw a closed contour. Figure 4.1 shows the sequence of drawing a closed contour. The user interface makes it possible for the user to edit the location of each vertex through the Vertex List. Also through the Edit Vertex window, the user can set the selected vertex to be either normal or anchored. The user can add a spring point that is applied to the selected vertex. The volcano has not been yet added to the interface, although the code for volcanoes is available in the programme. This was because we prefered to use pressure forces instead of volcanos, since applying pressure forces requires less user interaction and is easier than using volcanoes. Figure 4.2 shows the Vertex List and edit menu for the vertices of a current active contour.

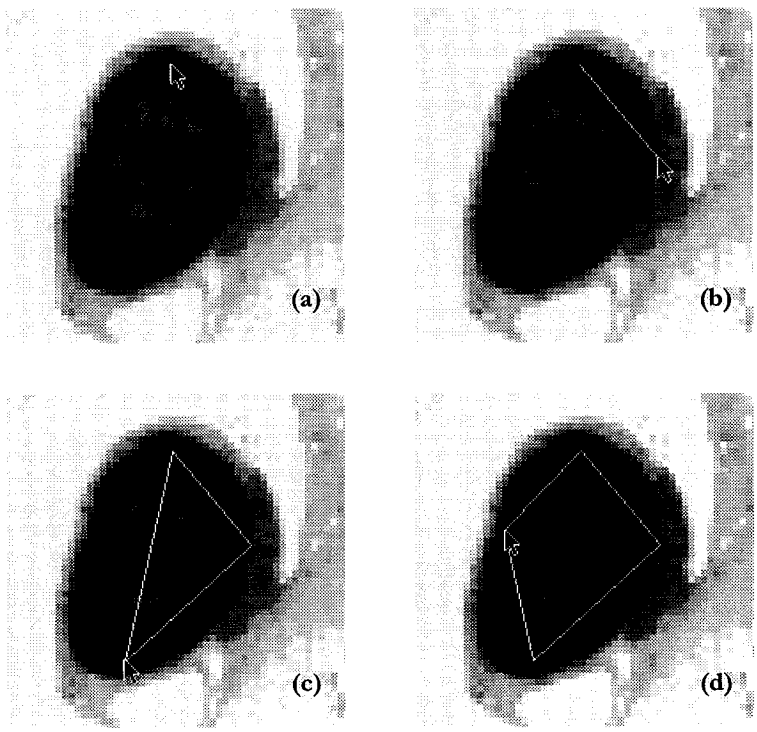


Figure 4.1: The sequence of drawing a closed contour using the mouse.

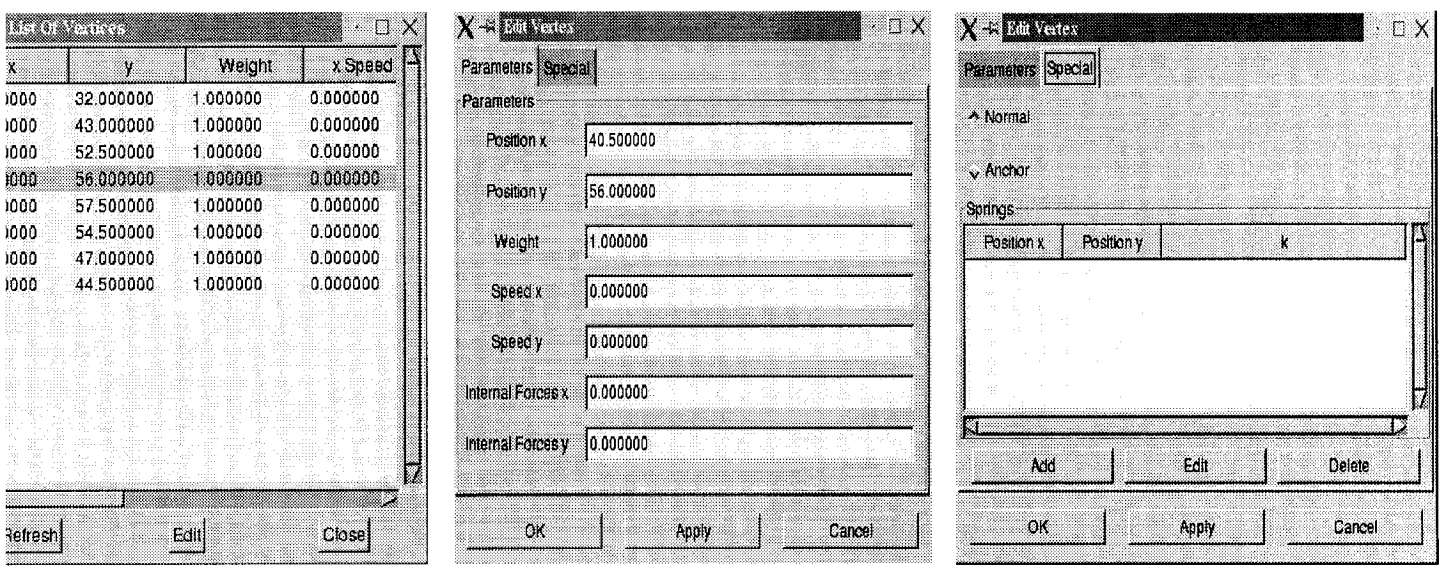


Figure 4.2: Vertex List, and edit menu for a selected vertex that can be manually modified by the user.

Oxiana includes both the gradient and GVF to calculate the external forces. The user can change the external force from gradient to GVF and vice versa, at any time. All the force weighting factors can be changed through the user interface by the user. The user can save the current active contour at any time and later the contour can be loaded for further applications.

Oxiana had to be debugged first and then to be modified in order to handle other features, which are described in this section.

4.5.1 Debugging

In the programme as implemented by Hamelin, Labonté & Pelletier (1999), there were two bugs that needed to be fixed beforehand. One was related to an "if condition" for finding the external force value of the corresponding vertices. The other one was related to the re-sampling process of merging close vertices, which was using a wrong pointer.

4.5.2 New features

Oxiana was improved to include the application of open contours and pressure forces, and some other features that are discussed in detail below.

4.5.2.1 Open contours

Oxiana was originally designed to automatically connect the first vertex to the last initialised one; therefore, it could only implement closed contours. We improved the programme in order to allow open contours. To do this, the code was modified in such a way that the first and the last vertices remain unconnected and are considered as anchors, if the contour is open.

Figure 4.3 shows an example of the sequence of drawing an open contour.

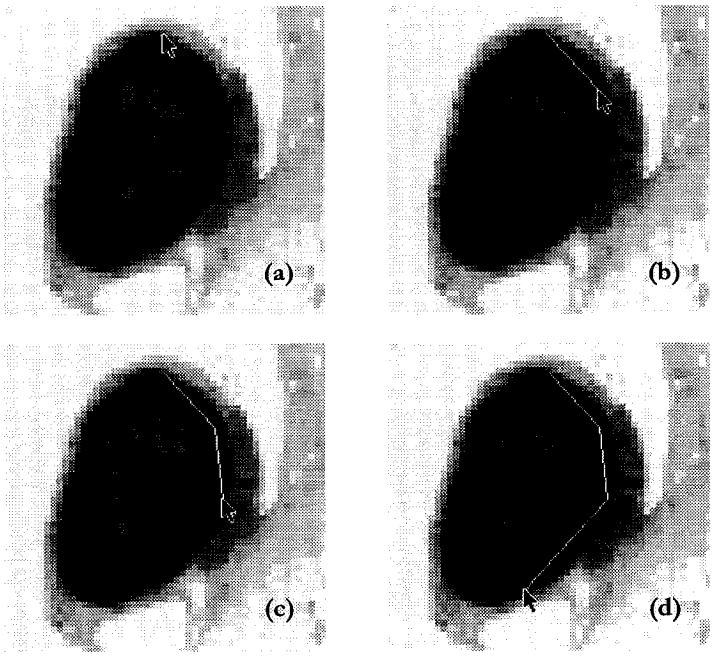


Figure 4.3: The sequence of drawing an open contour.

In a step-by-step process, the user can draw an active contour in several steps, iterating the active contour after each vertex is added. This provides connected open contours that can be considered as a simple contour in the end. This has the advantage of easily applying as many anchor points as desired, instead of implementing the entire active contour all at once and then anchoring the intermediate vertices using the edit vertex interface.

4.5.2.2 Pressure

Positive and negative pressure forces were the next feature added to Oxiana. As defined in section 3.2.3, the pressure forces are implemented as vectors perpendicular to the vertices. By combining Equations 3.9 & 3.17, we reach Equation 4.10, as follows:

$$\mathbf{F}_{total,i} = w_{ext}\,\mathbf{F}_{ext,i} + w_{int}\,\mathbf{F}_{int,i} + w_{damp}\,\mathbf{F}_{damp,i} + w_{pressure}\,\mathbf{F}_{pressure,i} \quad (4.8)$$

where $w_{pressure}$ and $\mathbf{F}_{pressure,i}$ are a pressure weighting factor and a unit force vector, respectively. $w_{pressure}$ is defined by the user, and can be a negative or positive value to deflate or inflate the active contour model, respectively. $\mathbf{F}_{pressure,i}$ is the radial unit vector to the vertex i.

4.5.2.3 Application to multiple slices

Oxiana has the ability to save the current active contour, and to then reload it for use on the same image or other images. Therefore, we can apply and iterate an active contour on a slice from a dataset and when the segmentation of a region of interest is completed, by opening the next slice and loading this active contour, the active contour will appear at the same coordinate positions as in the previous slice, and it can then be iterated on the new slice. Hence, no active contour initialisation is required for the second slice. This results in time saving and simplicity of operation. Figure 4.4 illustrates this feature of Oxiana on three successive slices of an MRM dataset. An initial active contour is implemented and iterated to segment a desired structure in a slice (*e.g.*, slice 109), and the converged active contour is loaded on the successive slice (*e.g.*, 110), for the same structure. Thereafter, by iterating the active contour on this slice, it fits itself to the boundary of the structure in this slice and segments the desired structure. In this example we continued this process on the next slice (*e.g.*, 111).

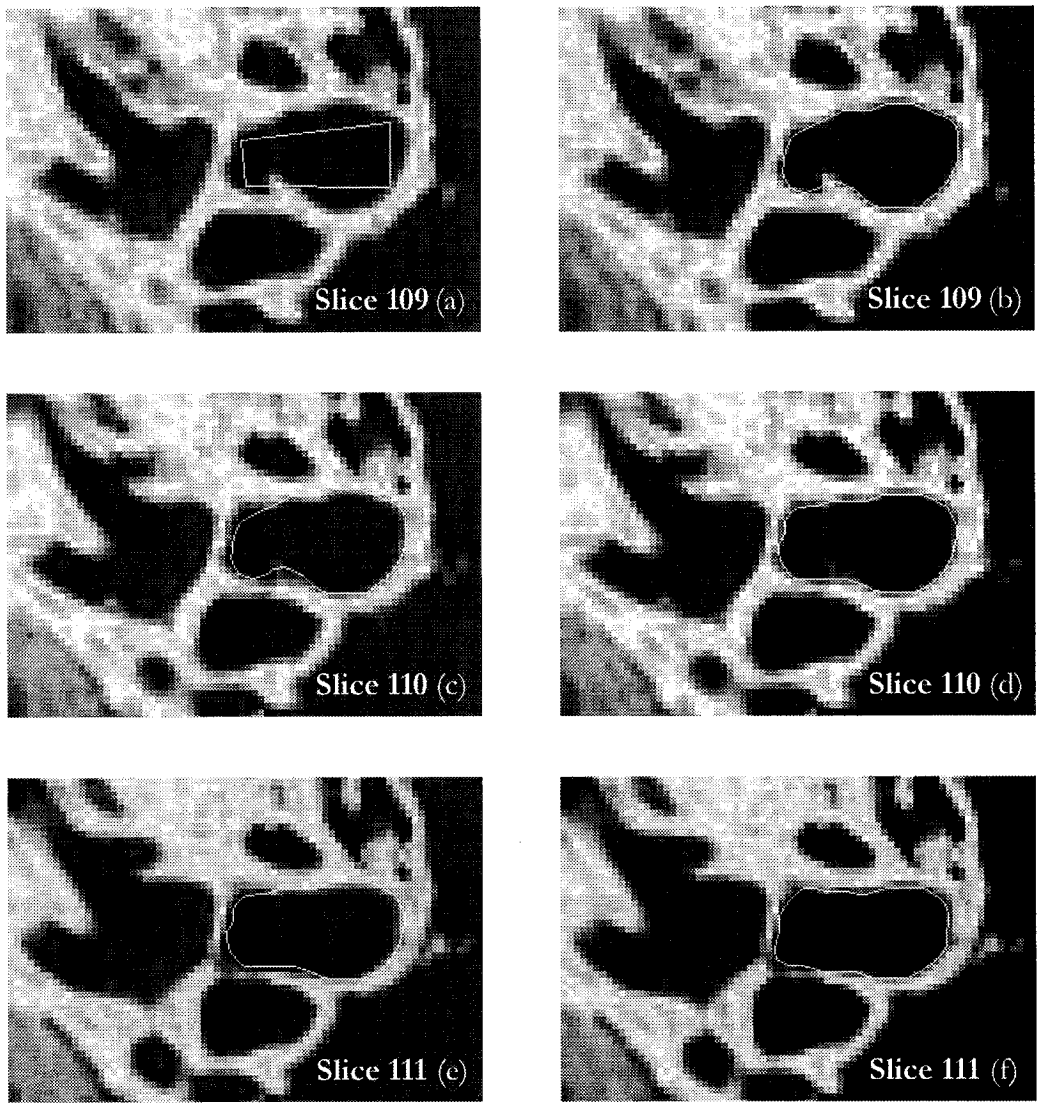


Figure 4.4: Example of the multi-loading property of active contours in Oxiana: (a) the initial active contour (white line) on slice 109 of MRM dataset, (b) the final contour after 150 iterations when the active contour converges to the boundary, (c) the active contour loaded on the same structure in slice 110, (d) the final contour on slice 110 after 30 iterations, (e) the active contour loaded on the same structure in slice 111, and (f) the final contour on slice 111 after 20 iterations.

4.5.2.4 Exporting

Oxiana can be a complementary tool for the computer programmes Fie (Fabrication d'imagerie extraordinaire) and Tr3 (for triangulating 3-D surfaces between serial-section contours), both written in our laboratory. Using Fie the user manually segments the structures of interest. Fie produces a text file containing locations of vertices, slice names, names of contours and various other information which is used by the Tr3 programme to create 3-D models for visualisation (using VRML) and finite-element simulation. Further information about Fie and Tr3 are available at http://audilab.bmed.mcgill.ca/~funnell/AudiLab/sw/.

Oxiana can output active-contour information that can be easily fed to Fie. The output of Oxiana is a text file including vertex coordinates, contour name, slice name, and z-coordinate. The last three items are defined by the user. Using any text editor, the user can copy and paste the information directly into the file used by Fie and Tr3.

4.5.2.5 Avoiding collapse

Discrete dynamic contours have a tendency to shrink due to the internal force characteristics, although it was claimed by Lobregt & Viergever (1995) that the shrinking problem is solved in discrete dynamic contours. In cases where the external forces cannot balance the force equation, either due to a high negative pressure force or because the initial contour was located far from the boundary, the active contour would keep shrinking and would eventually collapse. This results in losing the active contour information, since the programme re-draws the active contour after the user-defined number of iterations. A check was added to verify if the content of the active contour is NULL, *i.e.*, the contour has collapsed, after the user-defined number of iterations. In such a case the programme terminates the iteration process and avoids re-drawing the collapsed active contour. Moreover, a warning message will advise the user to save the current active contour before losing it.

4.5.3 Springs vs. anchors

Springs and anchors are constraints that are used to assist the convergence of active contours to desired boundary regions. A spring acts as a local constraint force that attracts a selected vertex to a desired location on the image. An anchor, by contrast, immobilises a vertex when applied.

To fix a vertex on a weak boundary, we can use either anchors or springs. Using Oxiana, a spring is easily applied to the desired vertex. For doing this, the desired vertex is selected through the pop-up vertex list and the user can change the x & y coordinates of the spring by typing them (*e.g.*, any actual boundary pixel) and the spring weighting factor. In the next iterations the spring affects the selected vertex and draws it to the spring point. To anchor a vertex, the desired vertex can be easily set to anchor mode through the vertex list, and also its location can be changed manually, if necessary, for instance to position it exactly on the boundary pixels. In such cases, the difference between a spring and an anchor is that the anchor must be located on the boundary and the user should recognise the boundary; while using a spring a user need only locate the spring point somewhere close to the boundary. A spring helps to move the vertex close enough to the boundary that the active contour can converge.

A spring can be set as positive or negative in order to either attract or repel the selected vertex. The difference between a spring and a pressure force is that the spring is applied locally, on only one vertex, while the pressure force is applied globally to the active contour and affects all vertices.

5. MATERIALS

This chapter describes different image datasets that we used for our active-contour experiments. The datasets include histology and Magnetic Resonance Microscopy (MRM) images. There are three kinds of image datasets available for the study of middle-ear images, including histology, MRM and x-ray Computer Tomography (CT). Since x-ray CT cannot show the soft tissues as well as MRM images, we chose not to use it. The two datasets used here are typical of the histology and MRM datasets available.

5.1 Histology

Histology is the study of tissue, which is an organized collection of cells and their supporting structures. In contrast with CT and MRM techniques, histology provides colour images with higher resolution. The preparation of histological images such as the middle ear includes several processes (Ham & Cormack, 1979).

First, the tissue is fixed with formaldehyde in order to prevent post-mortem decomposition, preserve structure and intensify subsequent staining. Next is decalcification of bones or calcified cartilages using decalcifying fluid. This process dissolves the inorganic salts that would cause difficulty in slicing the structures afterwards. The decalcified structure is embedded in, for example, paraffin or celloidin, to prevent distortion during slicing. Then, the structure is sliced and stained. Staining the slices is done by using hematoxylin and eosin or other stains. For instance, under the influence of hematoxylin, the tissue absorbing this stain will take on a blue to purple colour. For the tissues absorbing eosin, a pink to red colour is achieved, depending on the properties of the tissue. The last process is mounting the stained slices on a glass slide, for protection.

There are three issues related to the use of histological data for 3-D modelling purposes. Firstly, due to the thinness of sectioning, serial sectioning results in a large quantity of slices; since the staining and mounting process is time consuming, typically every fifth or tenth slice is mounted and the slices in between are discarded. Secondly, the mounting process results in misalignment between the slices; therefore, the histological data require an alignment

process before 3-D reconstruction. Thirdly, serious distortion may occur even after the embedding process.

The histology dataset that we used for our experiments was prepared by C. C. Northrop and S. R. Levine from The Temporal Bone Foundation, Inc., Boston. The data are from a human ear, sectioned at 20μm in the horizontal plane; every fifth section was mounted on microscope slides. The slides were scanned using a Polaroid slide scanner with a resolution of 2700×2700 pixel/inch. The size of each image is 1976×1684 pixels. Figure 5.1 shows an example of a middle-ear histological slice. Although our original histological dataset consists of colour images, in our experiments, which are discussed in the following chapter, we used the histological dataset after converting them to grey-scale images.

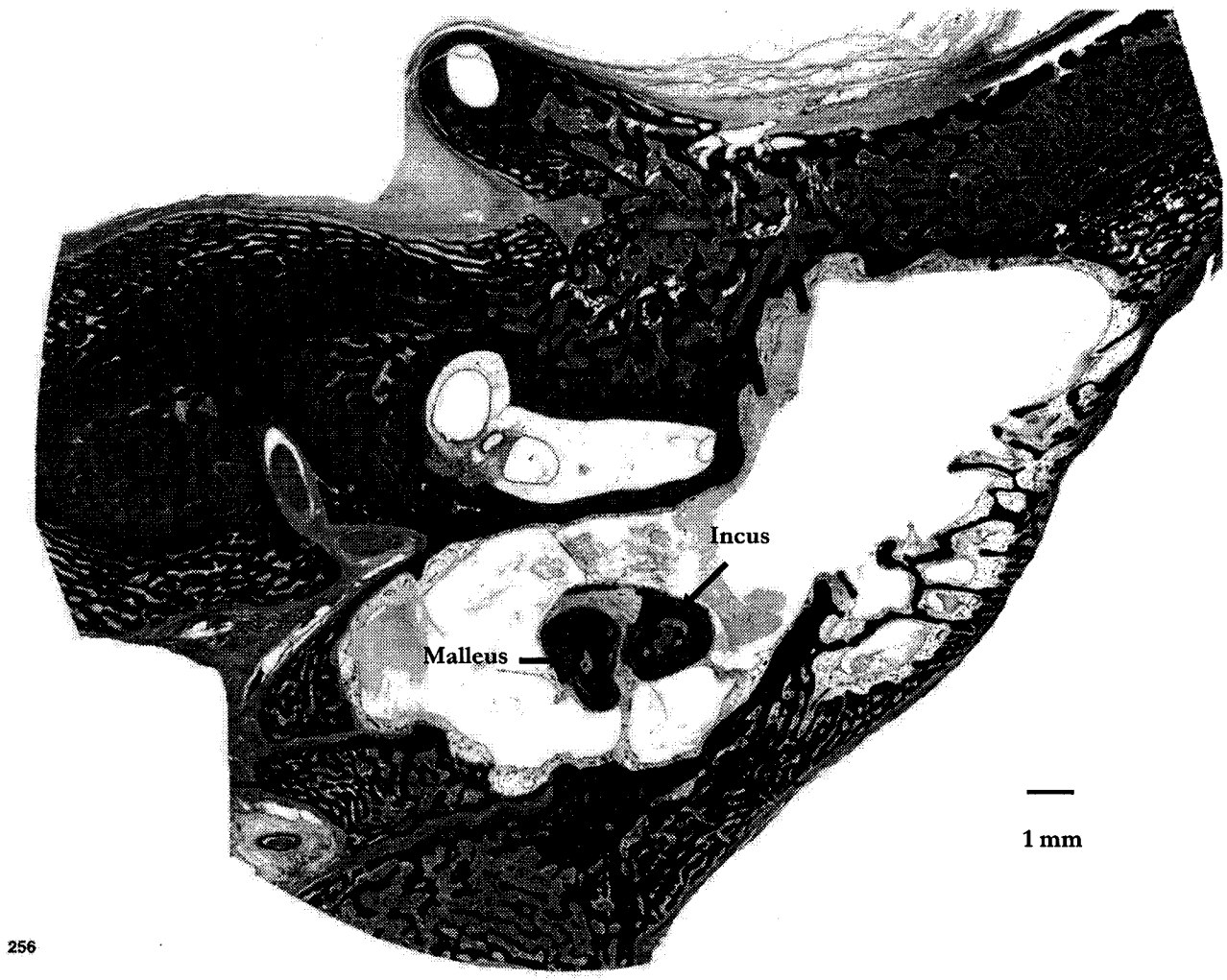


Figure 5.1: Slice number 256 from histological dataset of human middle ear.

5.2 MRM

Magnetic Resonance Microscopy (MRM) is an extension of Magnetic Resonance Imaging (MRI) to the microscopic domain, with higher spatial resolution than MRI. They are both based on Nuclear Magnetic Resonance (NMR). The principles of NMR, MRI, and MRM are provided in several publications (*e.g.*, Curry *et al.*, 1990; Bronzino, 1995; Brown *et al.*, 1999). MRM is a new form of microscopy that overcomes many limitations of conventional microscopy, such as the need for slicing and staining, or ionizing radiation, which are all invasive and destructive procedures. Both MRM and MRI are powerful, non-invasive and high-quality imaging modalities for three-dimensional cross-sectional scanning with the ability to demonstrate anatomical structures and pathological changes. Soft-tissue differentiation is generally better than with x-ray imaging.

The MRM dataset we used for our experiments was made available by M. Henson & O.W. Henson, Jr., from the University of North Carolina at Chapel Hill. Middle-ear specimens were scanned at Duke University in the Center for In Vivo Microscopy. The dataset number 13641 is from a human ear. The dataset consists of 180 transverse sections, each 187×256 pixels, and the voxel size is approximately 120μm. Figure 5.2 shows a sample slice.

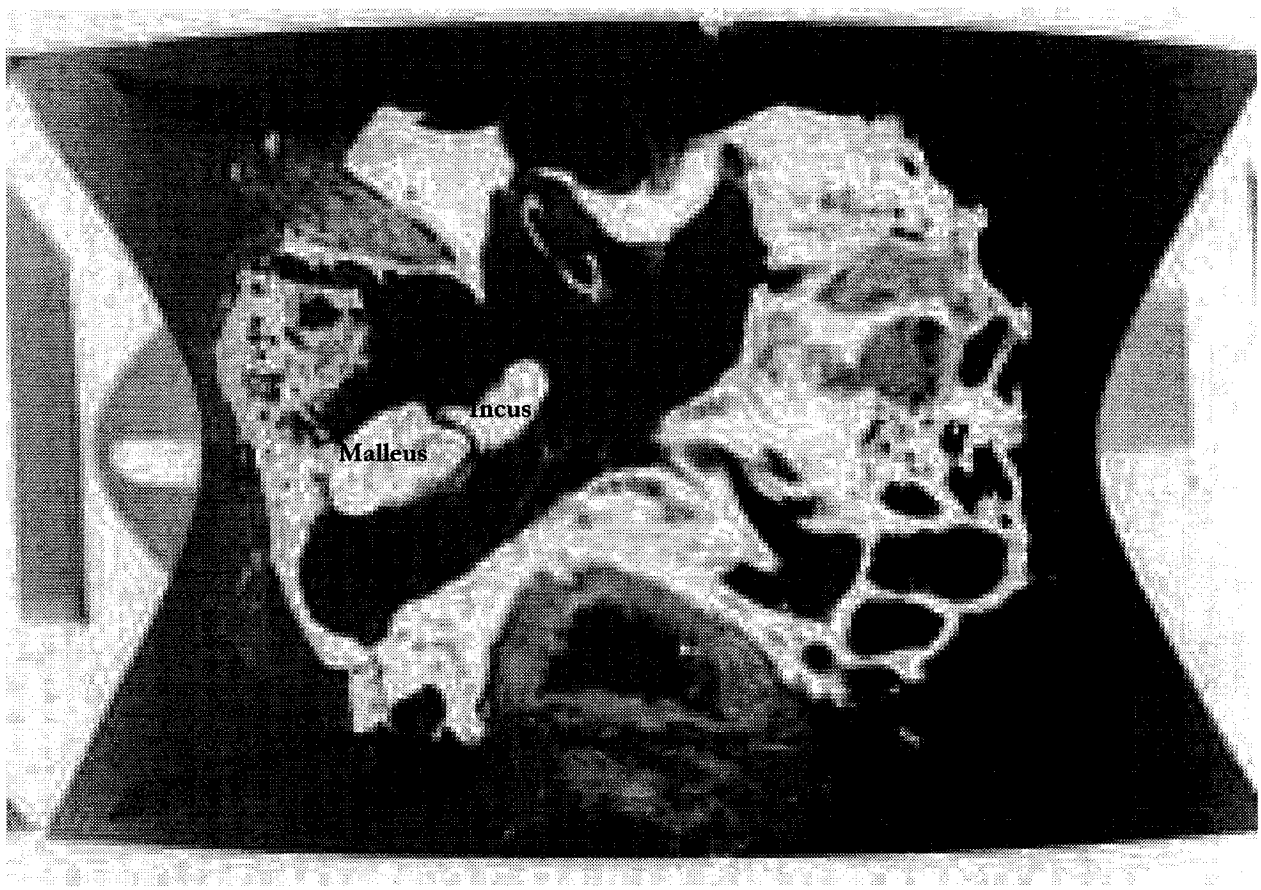


Figure 5.2: Slice number 110 from MRM dataset of human middle ear.

6. RESULTS

Our research on the segmentation of middle-ear images using active contours includes the following phases:

- Investigation of the calculation of GVF;
- Experiments with force weighting factors;
- Comparisons between parametric and discrete dynamic active contours for segmentation of histological structures;
- Comparisons between parametric and discrete dynamic active contours for segmentation of MRM structures;
- Experiments with discrete dynamic open contours for histological and MRM structures.

We discuss the phases in the following sections.

We used two different programmes for the two active-contour approaches. One programme, which implements parametric active contours, GVF and pressure, was written in Matlab (The MathWorks Inc.) by Xu & Prince (1997) from the Image Analysis and Communications Laboratory, Johns Hopkins University. Oxiana, developed in our lab, was used to implement the discrete dynamic contours.

6.1 Calculation of GVF

We did experiments with GVF in order to find the appropriate number of iterations for convergence. We applied different numbers of iterations to compute the GVF, and then we compared the external force maps for each number of iterations. Figures 6.3 and 6.4 show the external force maps for 5, 40 and 80 iterations for a histological image (Figures 6.1) and for an MRM image (Figure 6.2), respectively.

We found that 80 iterations is appropriate, although for many images 40 iterations seems good enough (as in Figure 6.4). It is important to note that this is not a quantitative analysis. The criterion for choosing 80 iterations is based on the observation that we actually see no

visual difference between 80 iterations and the next examined number of iterations above it, *i.e.*, 100 iterations. GVF computation is a time consuming process; therefore, by choosing 80 iterations, less time is required for GVF computation than with larger numbers of iterations.

Figure 6.1: Malleus bone in slice 196 of the histological dataset.

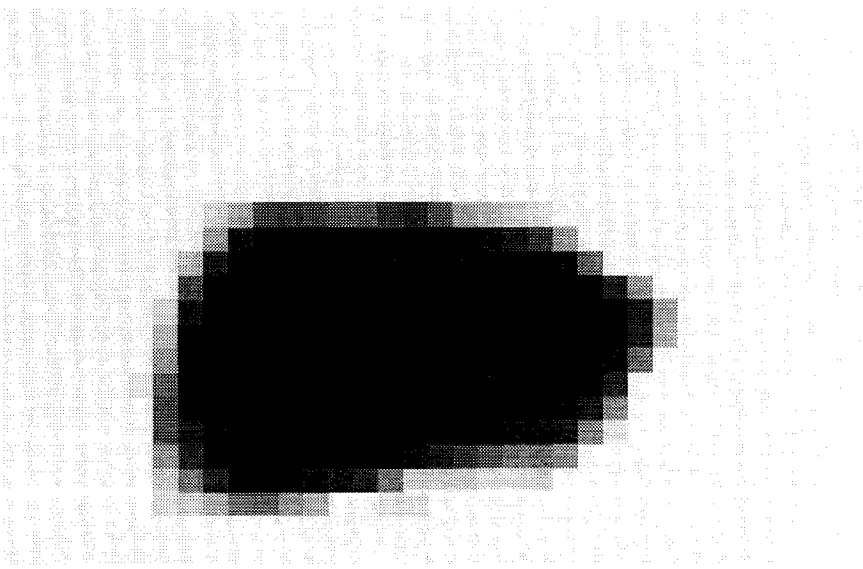
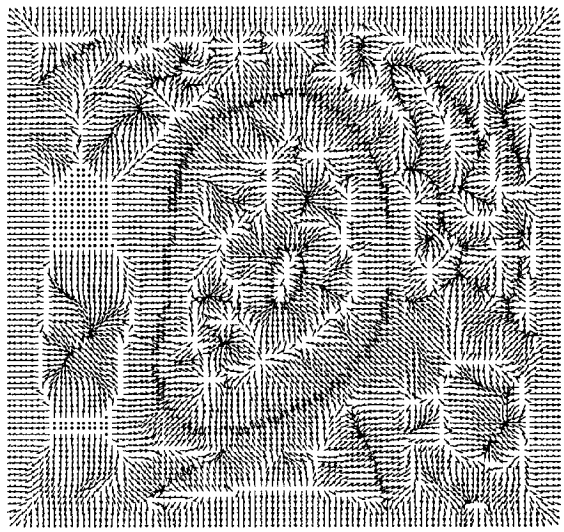
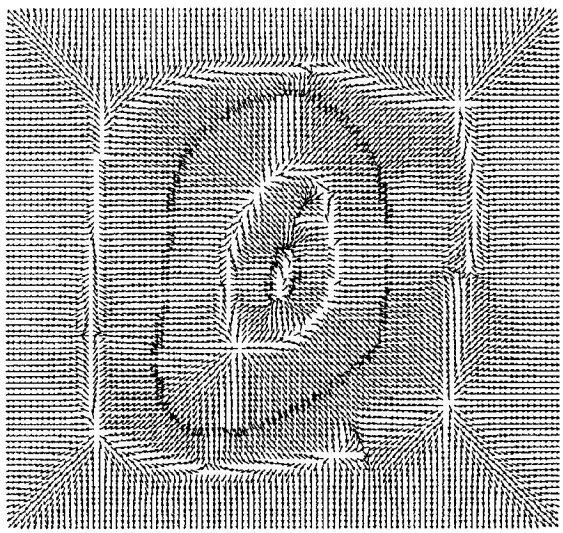
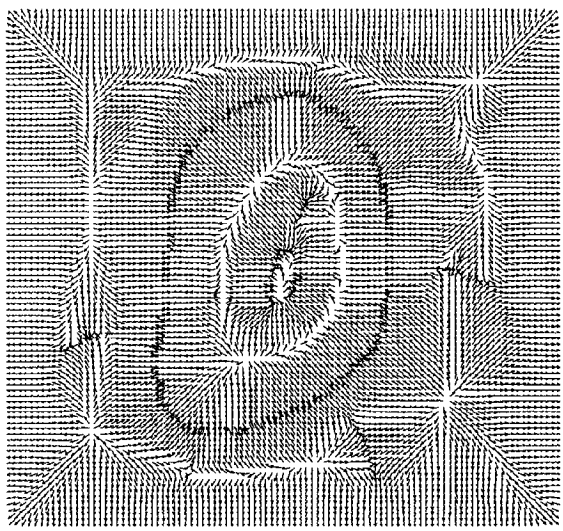


Figure 6.2: A cavity in slice 110 of the MRM dataset.







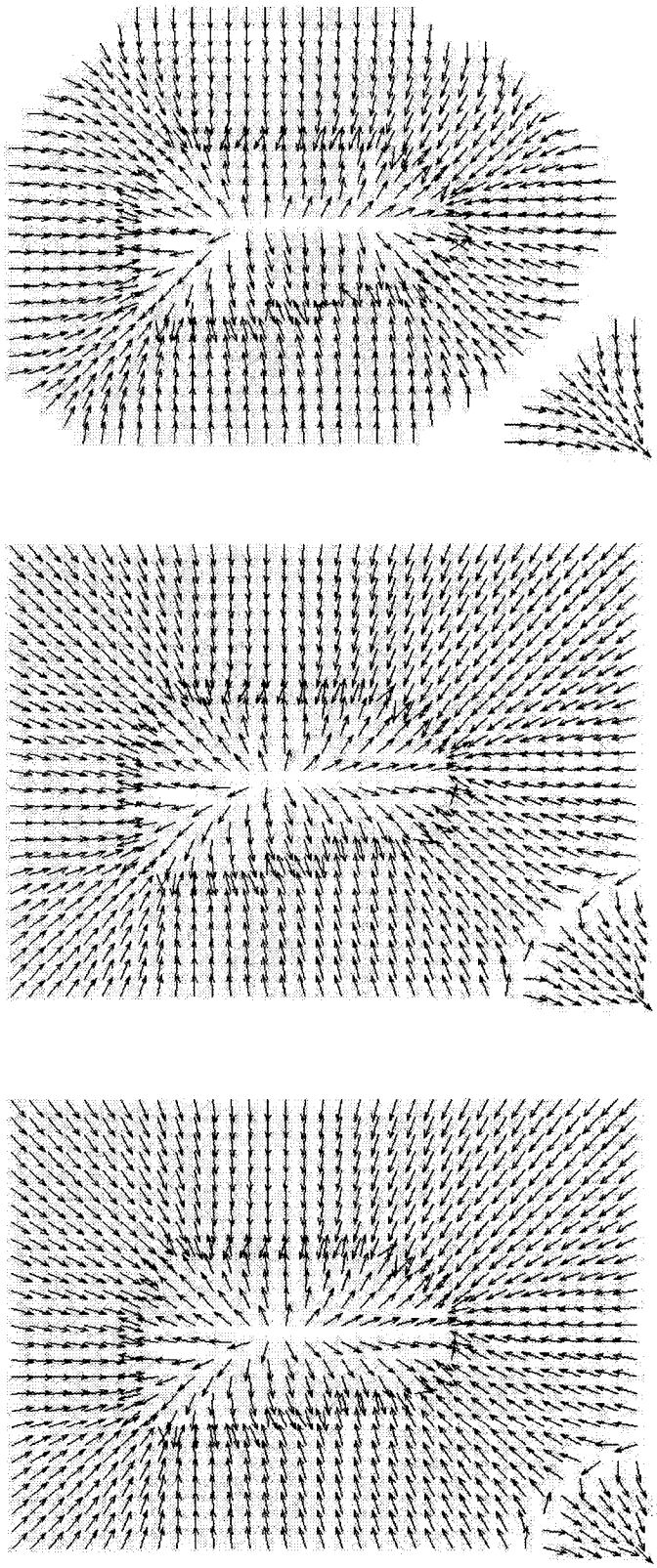


Figure 6.4: The external force maps produced by GVF of Figure 6.2: for (top) 5, (middle) 40 and (bottom) 80 iterations.

6.2 Force weightings for active contours

One of the most important issues for active-contour applications is choosing the appropriate weighting factors, *i.e.*, tension, rigidity and the external force weightings for the parametric approach, and the internal, external and damping weighting factors for the discrete dynamic contours. Experiments were performed to evaluate the performance of the two types of active contours with different combinations of force weighting factors, as discussed in the following sections.

According to the force-balance equations for parametric (Section 3.2) and discrete dynamic (Section 3.5) contours, the active contour evolves and then comes to rest when it has minimized its energy. The change of the location and number of the vertices reaches a very small number or zero. In consequence, the positions of the vertices remain practically unchanged. The combination of the internal and external force weightings has a crucial effect on the force-balance equation. Any inappropriate combination of the weighting factors affects the active-contour performance and produces undesired results.

6.2.1 Parametric contours

Figure 6.5 shows some examples of inappropriately chosen values for tension, rigidity and the external force weighting factors. In these examples, the values of the weightings are exaggerated in order to clearly show the effects. As discussed in Section 3.2.1, increasing the tension weight will cause the active contour to behave more like a rubber band trying to contract. Figure 6.5(a) shows this effect for $\alpha = 0.5$. Figure 6.5(b) shows the result of increasing rigidity to a high value (β=20), which causes the active contour to act as a stiff wire which cannot converge to the concavities. Increasing the external force too much causes the active contour to follow the external force influence and the internal force (tension and rigidity) can no longer preserve the contour's smoothness. Figure 6.5(c) shows this effect for κ=5.

It is important to note that the best combination of the weighting factors may vary depending on the characteristics of the region of interest (*e.g.*, the contrast and shape), the number of points of the initial contour, and the distance of the points from the boundary. We found that the combination suggested by Xu & Prince (1997), *i.e.*, κ=0.6, $\alpha = 0.05$ and $\beta = 0.01$, gave good results for most of the selected regions of interest of our datasets.

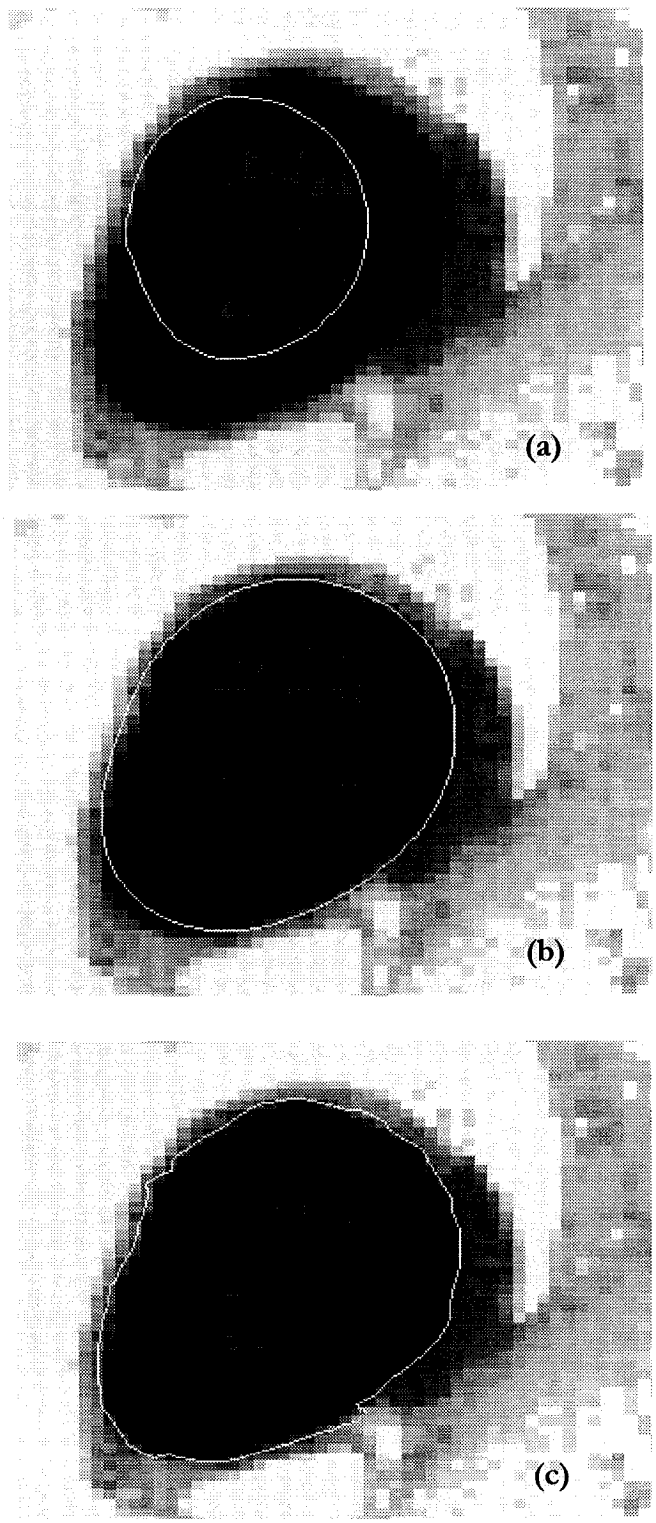


Figure 6.5: The effect of increasing internal force parameters: (a) tension weight (α =0.5), (b) rigidity weight (β =20) and (c) external-force weight (κ=5).

The acceptable range for each weight is as follows.

Increasing β will increase the rigidity of the model and would affect the shape even if close to start with. We found that the rigidity weighting factor can be increased from 0 to 0.03 with almost the same results. Decreasing the tension weight causes the active contour to follow the influence of the external force and lose its smoothness. The acceptable range that we found for tension was from 0.02 to 0.08. For values over $\alpha = 0.08$, the active contour must be initialised close to the boundary; otherwise, the tension force tries to contract the model and prevents the contour points from easily converging to the boundary. Figure 6.6 shows the example of an initial contour located close enough to the boundary (a) that converges for both $\alpha = 0.02$ (b) & $\alpha = 0.08$ (c), and an initial contour located farther from the boundary (d) than the one in (a), which converges for $\alpha = 0.02$ (e) but fails for $\alpha = 0.08$ (f) and continues shrinking due to the internal force characteristics of the parametric contours.

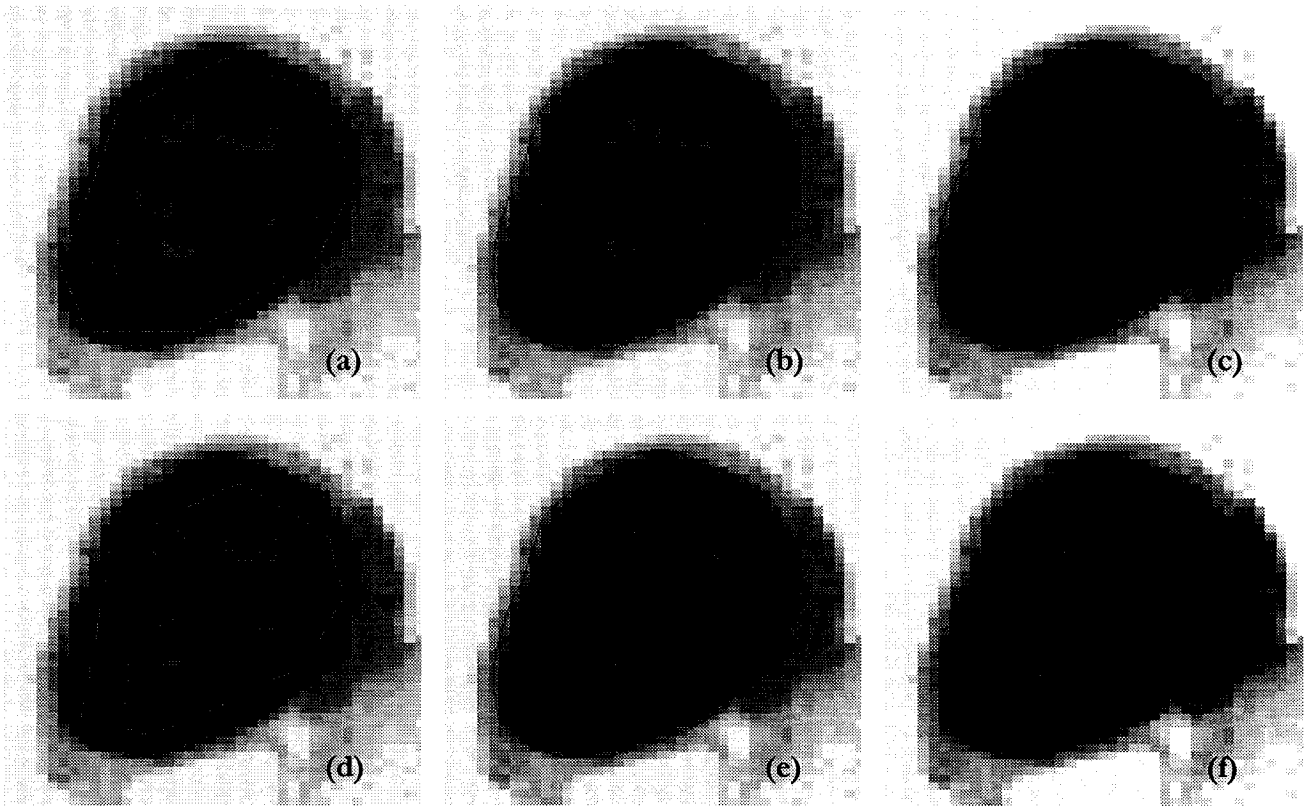


Figure 6.6: Examples of the relationship between the initial position of the active contour & tension: an initial contour located close enough to the boundary (a), this contour converges for α=0.02 (b), and for α=0.08 (c). Another initial contour located farther from the boundary (d) than the one in (a), and it converges for α=0.02 (e), but fails for α=0.08 (f).

We found that with parametric active contours the ratio of force weightings is more important than the values themselves. For instance, if the force weighting is increased four times, which indeed exceeds the previously recommended ranges for the force weightings (*i.e.*, κ =2.4, α = 0.2 & β=0.06), the active contour behaves as it does for κ=0.6, α = 0.05 & β=0.01, but it requires that the initial contour be closer to the boundary. Although keeping the rigidity weighting factor equal to zero when applying our suggested combination does not noticeably affect the active-contour behaviour, we found that when using the combinations that are several times bigger than our suggested combination, if the rigidity is zero the behaviour of the active contour changes: the effect of tension becomes more evident, and the initial contour must be located closer to the boundary. In fact tension and rigidity have modifying effects on each other and it becomes noticeable when the force weightings are proportionally increased. The other combinations that we used for our experiments on histology and MRM datasets are discussed in Sections 6.3 and 6.4.

6.2.2 Discrete dynamic contours

Figure 6.7 shows simple examples of inappropriate combinations of the weightings for a discrete dynamic contour. In any of these examples the forces cannot balance each other, the total force never reaches zero or a small number, and consequently the positions of the vertices continue to change. In this situation the active contour energy is never minimized and it keeps evolving. In such a case the active contour approaches the boundary but it is not smooth, as seen in Figure 6.7. The weighting factors are slightly exaggerated in order to clearly illustrate the details for the limited size of image. For instance, decreasing the damping and the internal force weightings to a smaller value than required, or increasing the external force more than required, *e.g.*, w_{ext}=6, will cause the active contour to follow the influence of the external force without having the proper effects of the internal and the damping forces to keep the model smooth. Figure 6.7(a) shows this example. In another case, if the internal force weighting is higher than both the damping and external forces weightings in such a way as to unbalance Equation 4.16, *e.g.*, w_{int}=2.5, the position of each vertex is dominated by the internal force of that vertex, which is either in the same direction as its radial unit vectors or in the opposite direction. This situation can be seen in Figure

6.7(b). Finally, in the case where the damping weighting factor exceeds its proper limit, *e.g.*, w_{damp}=3.5, it will result in a non-smooth contour model. This is the result of a situation where the total force from Equation 4.6 is never minimized for each vertex and consequently, too much velocity is fed back (in the form of positive feedback) to the force balance equation through the damping force. Figure 6.7(c) shows this example. Also, if any two out of the three weighting factors are kept in their acceptable ranges, and only the remaining one is set to zero, the absence of this force will affect the force balance equation and the behaviour of the active contour. For instance, if the external and damping force weightings are kept and the internal force weighting is set to zero, the active contour follows the external force influence and the contour is not smooth. Figure 6.8(a) shows this effect. In another case, if we set the damping force to zero, this force weighting combination does not result in a smooth active contour. This is because the damping force, which to some extent provides the smoothness of the active contour, is absent. However, in this case the active contour converges to the boundary. Figure 6.8(b) shows this effect. In the case where the external force weighting is zero, the effect of the internal force is dominant and leads the active contour to shrink, so the active contour cannot converge to the boundary. Figure 6.8(c) shows this effect.

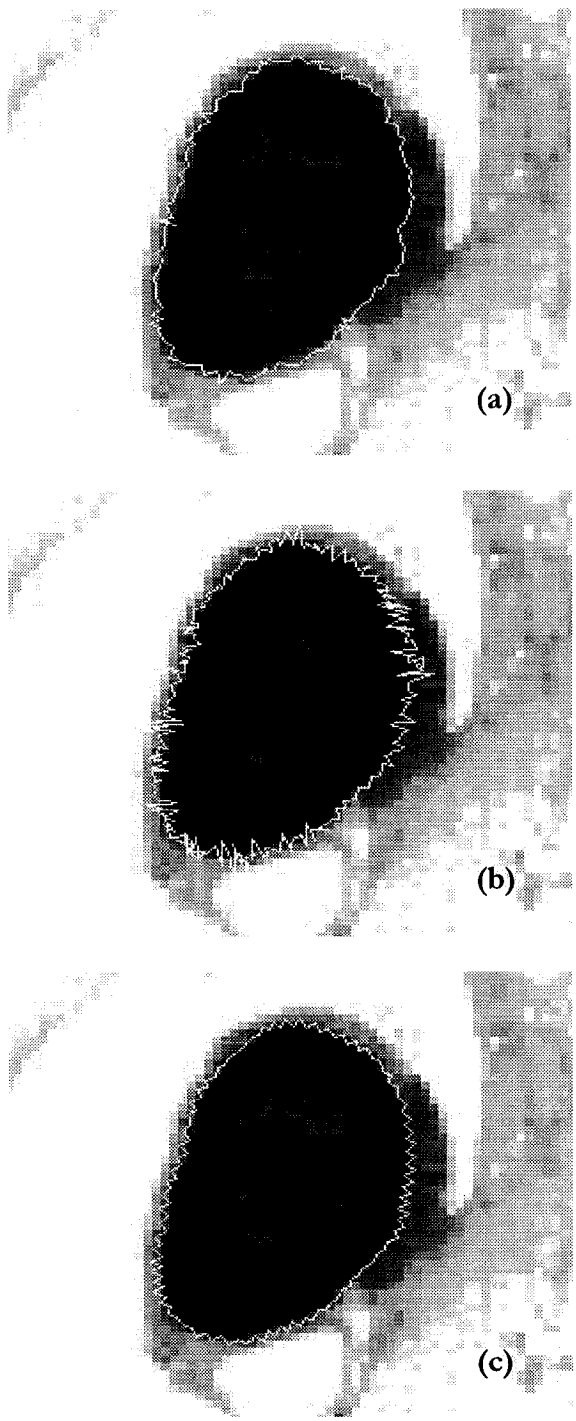


Figure 6.7: Effect of increasing force term values: (a) the external force weight, (b) the internal force weight and (c) the damping force.

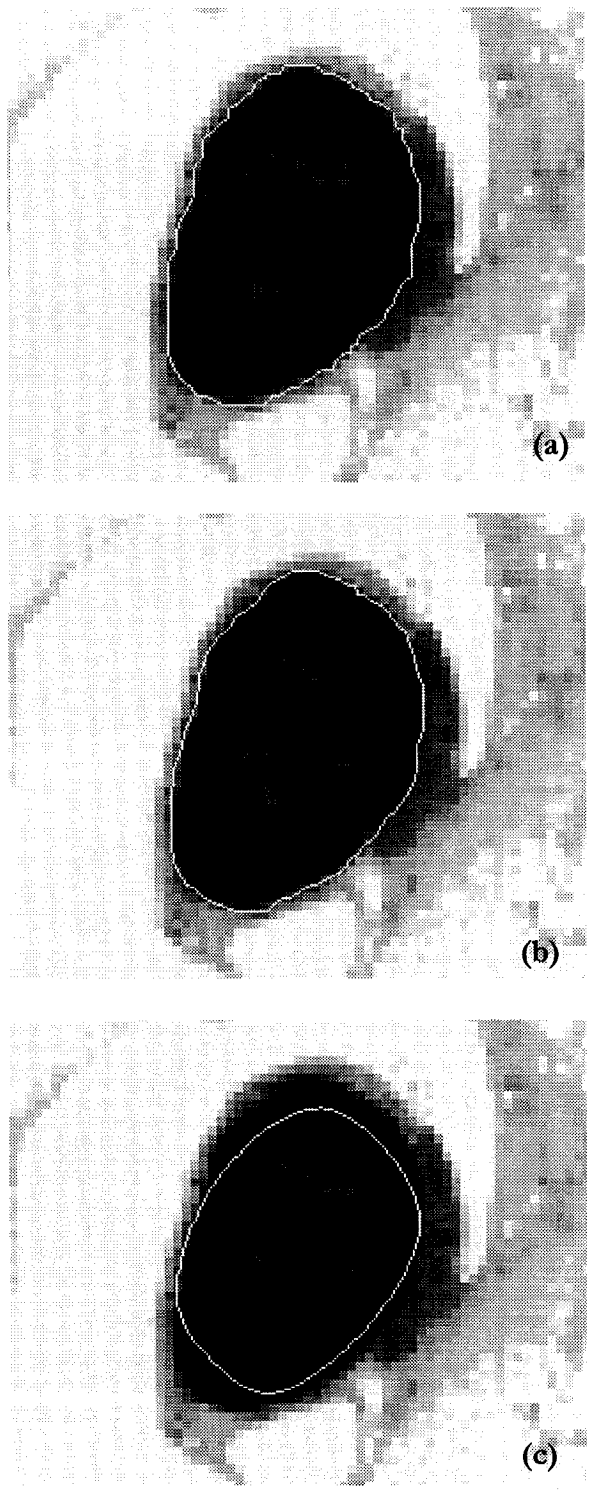


Figure 6.8: Examples of setting force weightings to zero. (a) the internal force weight is set to zero, (b) the damping force weight is set to zero, and (c) the external force weight is set to zero.

For the datasets that we used for our experiments, the proper weighting-factor ranges that we found were w_{int}=0.5±0.3, w_{ext}=0.6±0.3 & w_{damp}=1±0.5. The combination can be varied depending on the characteristics of the regions of interest (*e.g.*, contrast). This will be discussed later in Sections 6.3 and 6.4 on experiments with discrete dynamic contours.

We found that only with discrete dynamic contours, and not with parametric contours, the ranges of the force weightings are more important that the ratios. For instance, by increasing the force weightings four times, which indeed exceeds the ranges for the force weightings (*i.e.*, w_{int}=2, w_{ext}=2.4 & w_{damp}=4), the active contour does not behave as it does when w_{int}=0.5, w_{ext}=0.6 & w_{damp}=1, and it affects the smoothness of the active contour. This is due to the fact that the total force is proportional to the force weighting factors; therefore, by increasing all the weighting factors two times the total force will be doubled. Figure 6.9 shows the result of the best combination of force weightings(a) and then shows the effect of increasing the weighting factor twice (b) and four times (c). By increasing the weighting factors the smoothness of the contour is degraded.

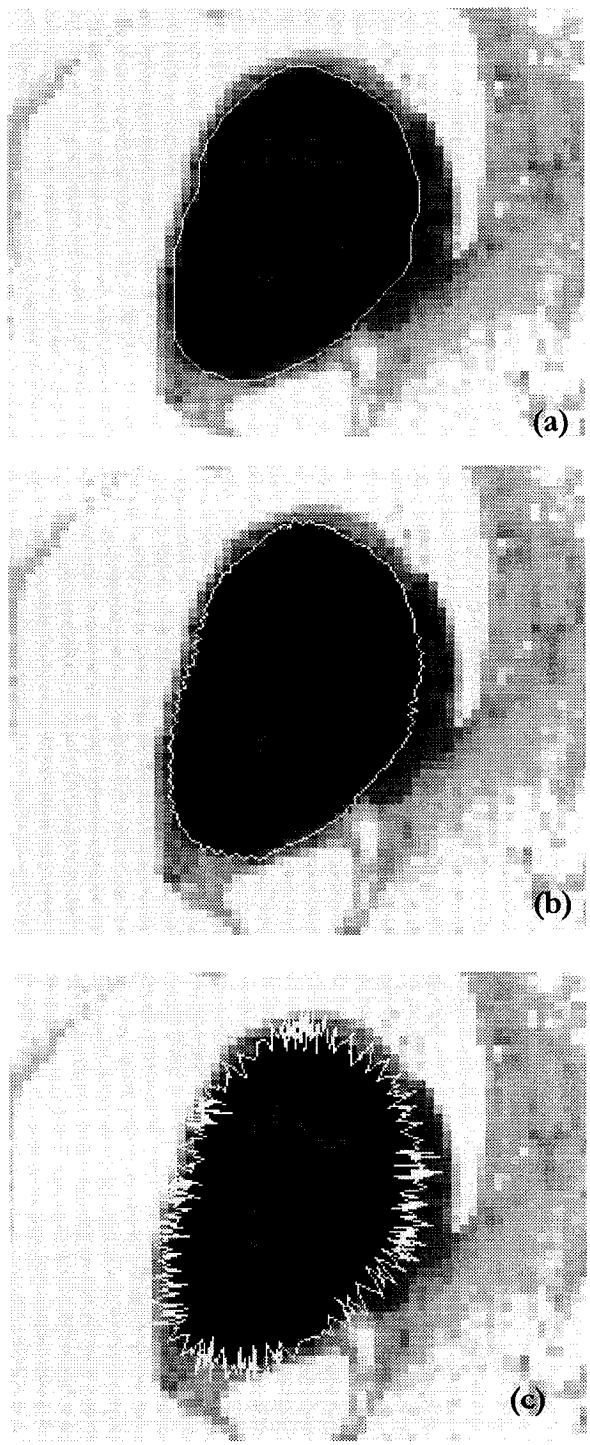


Figure 6.9: Examples of increasing the force weighting factors: (a) the best combination, (b) two times the best combination and (c) four times the best combination.

6.3 Parametric & discrete dynamic contours with the histological dataset

This section includes the experiments with closed contours on the histological dataset, comparing parametric and discrete-dynamic contours. The studies contain experiments with the gradient alone, with GVF, and with the gradient plus pressure force. To precisely compare the performance of the two active-contour approaches, identical initial contours were applied for both approaches.

The cross-sectional anatomical structures chosen for the experiments were mainly two tiny bones of the middle ear: malleus and incus. Figure 6.10 shows the bones as quasi-oval dark regions.

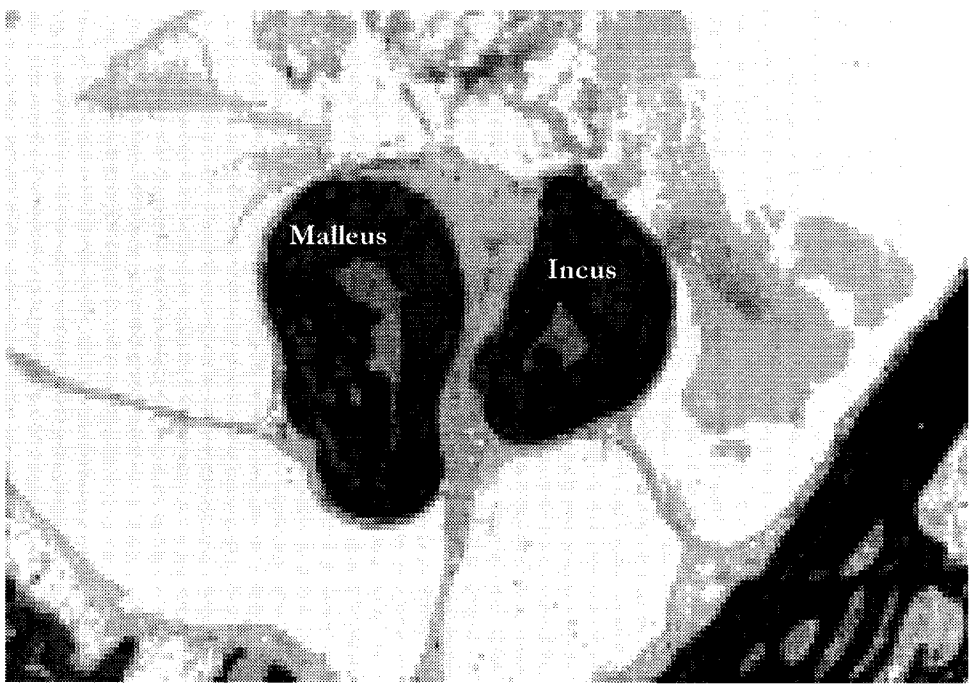


Figure 6.10: Histological slice 241 showing the malleus & incus bones as quasi-oval dark regions.

6.3.1 Timings

Both our programmes, Oxiana (discrete-dynamic approach) and the Matlab programme (parametric approach) are run under Linux on a system with a 1-GHz Intel processor and 1GB of RAM. Due to the faster performance of programmes written in C than of those implemented in Matlab, the speed of the computation is much higher with Oxiana. We compared the execution times for computing GVF and gradient, and for active-contour iterations, using the two programmes. The speed of GVF computation is different between Oxiana and the Matlab programme. For instance, for each histological slice (770 x 500 pixels), the computation of GVF takes approximately 20 seconds with Oxiana and approximately 3 minutes 30 seconds with the Matlab programme. Calculation of the gradient with Oxiana and the Matlab programme takes 2 and 4 seconds, respectively. It is important to note that Matlab has a built-in function for computing the gradient and therefore the speed difference between Oxiana and the Matlab programme is not as big as for the computation of GVF.

Again, the speed of iterations was different with Oxiana than with the Matlab programme, and it is much higher with Oxiana. The information about the time required for active contour iterations is provided in Table 6.1. This is a comparison between the programmes that implement the active-contour approaches and cannot be considered as a comparison between the approaches themselves.

We concluded that the computation for each iteration of an active contour with Oxiana takes approximately 200μsec, while with the Matlab programme each iteration takes approximately 1.5 sec. For this experiment, we used an initial contour composed of 11 vertices as shown in Figure 6.11, for both active contours. The time difference between the iterations in Oxiana and the Matlab programme is firstly due to the lower speed of computation of the programmes implemented in Matlab, and secondly is related to the algorithm implementation that computes the inverse of the n-by-n (n being the number of vertices)) pentadiagonal matrix (described in Section 4.1). Although the matlab INV function (for calculating $(A+\gamma I)^{-1}$ from Equation 4.6) is a built-in function, in general it is a time-consuming calculation that affects the speed of the Matlab programme, for each iteration.

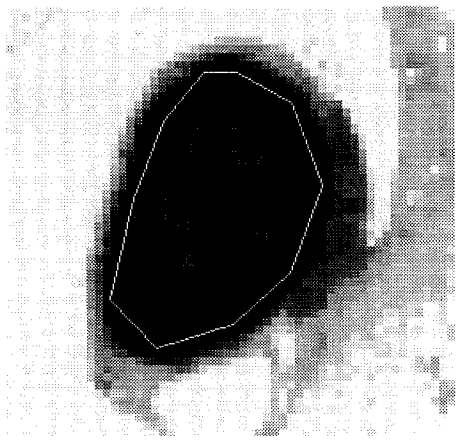


Figure 6.11: Example of an active contour. It was used for the calculation of the speed of Oxiana and the Matlab programme. The initial contour consists of 11 vertices on a selected structure. This contour was used for both active contours, *i.e.*, parametric and discrete dynamic contours, to compute their iteration times, as provided in Table 6.1.

Oxiana (iterations/sec)	Matlab programme (iterations/sec)
10,000/2	5/7
20,000/4	10/15
50,000/10	20/30
100,000/20	40/60

Table 6.1: The comparison between the number of iterations for the same initial contour in Oxiana and in the Matlab programme. We needed to apply very large numbers of iterations for Oxiana, to be able to calculate the interval between each iteration.

6.3.2 Gradient

In this experiment we applied the gradient as the external force for active contours. Table 6.2 provides the details of the experiments on some of the structures of interest.

Slice number	Structure	No. of points for the initial contour	Discrete approach	Parametric approach
			No. of iterations	No. of iterations
181*	Malleus	11	30	40
187	Malleus	10	35	50
196	Malleus	11	85	100
246*	Incus	11	50	65
266	Incus	12	60	80
306	Incus	13	75	90
306	Malleus	12	60	80

Table 6.2: Applying gradient alone for discrete dynamic and parametric active contours. The structures marked with asterisks are shown in Figure 6.12.

The most important thing for the initial contour was that they must be close enough to the boundary in order to converge. The number of points itself is not important, except in so far as they allow the creation of an initial contour close enough to the boundary. Figure 6.12 shows the results of boundary delineation using parametric and discrete dynamic contours for the selected structures marked with asterisks in the table. The two active-contour approaches had similar boundary-delineation results using the gradient technique. The gradient technique gave good results for all the structures of interest.

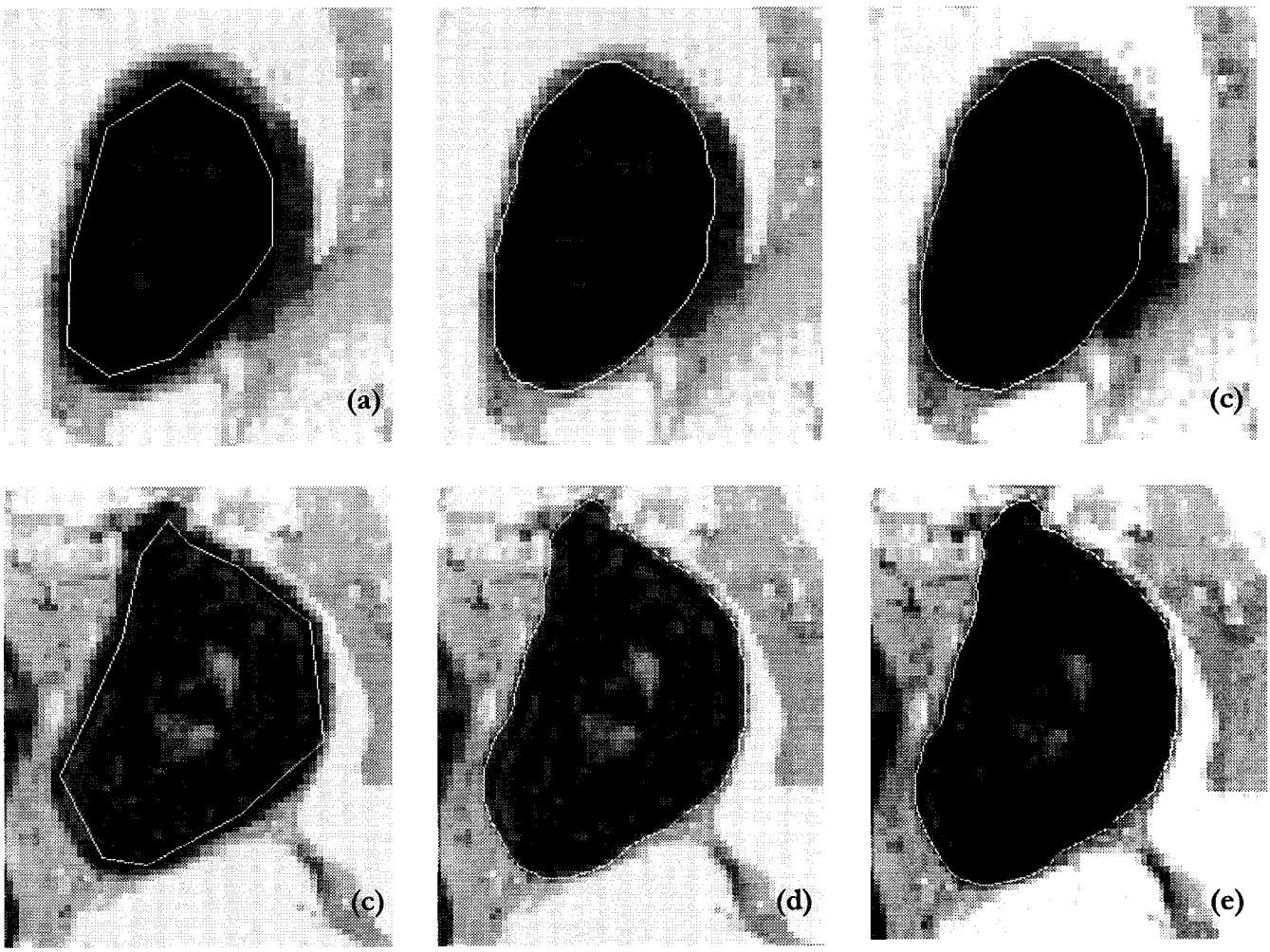


Figure 6.12: Segmentation using the two contour approaches. The active contours were tested with the malleus from slice 181: (a) the initial contour, (b) using discrete-dynamic approach and (c) using parametric approach, and for the incus from slice 246: (c) the initial contour, (d) using discrete-dynamic approach and (e) using parametric approach.

Since the capture range of the gradient is limited, however, the initial contour must be located close enough to the boundary; otherwise the active contours may not converge to the boundary. Figure 6.13 shows the results for an initial contour which was located close to the boundary, and for two which were located farther from the boundary and that failed to converge to the boundary.

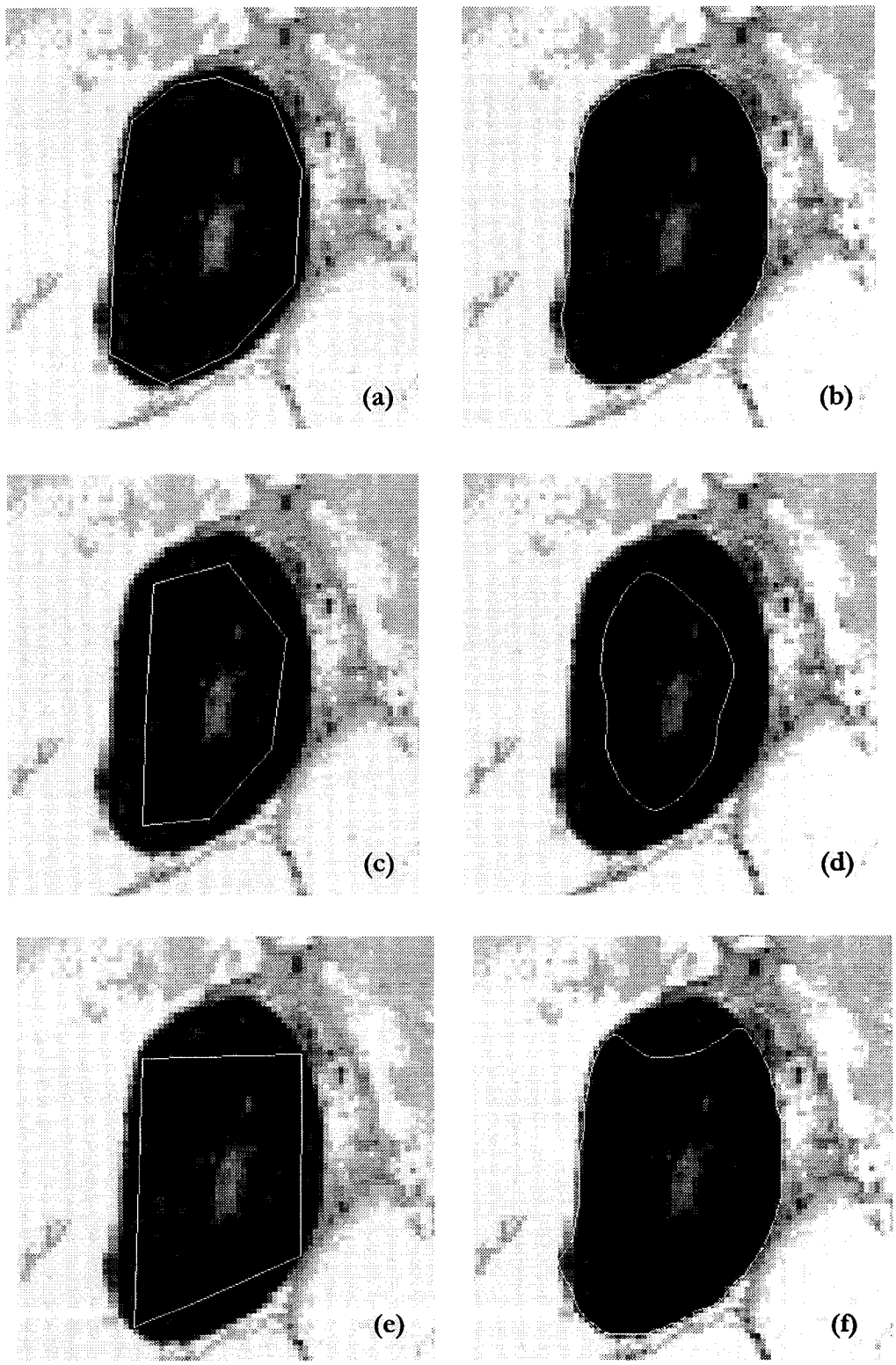


Figure 6.13: Segmentation with different initial contours. The active contours were applied to the malleus from slice 196: (a) the initial contour (white line) located close to the boundary, (b) the final contour for the initial contour shown in (a), (c) & (e) the initial contours located far from the boundary, (d) the contour from (c) after 600 iterations that is tending to collapse and fails to converge to the boundary and (f) the contour from (e) after 600 iterations that is not completely converged to the boundary and cannot converge any more.

Using the same initial contour, the parametric-active contour requires a larger number of iterations than does the discrete dynamic contour, as indicated in Table 6.2.

Using anchors or springs can overcome the limited capture range of the gradient for regions where the initial active-contour points are located far from the boundary. In the following sections we evaluate other suggestions, *i.e.*, GVF and pressure, to improve the capture range of the gradient.

6.3.3 GVF

In this section we evaluate GVF as the external force since the use of GVF attempts to overcome the capture-range limitation of the gradient method; so the initial contour can be located farther from the boundaries. Figure 6.14(a) shows an example of an initial contour located too far from the boundary, so that the active contour using the gradient fails to converge to the boundary (b), while GVF helps the active contour to converge to the boundary (c).

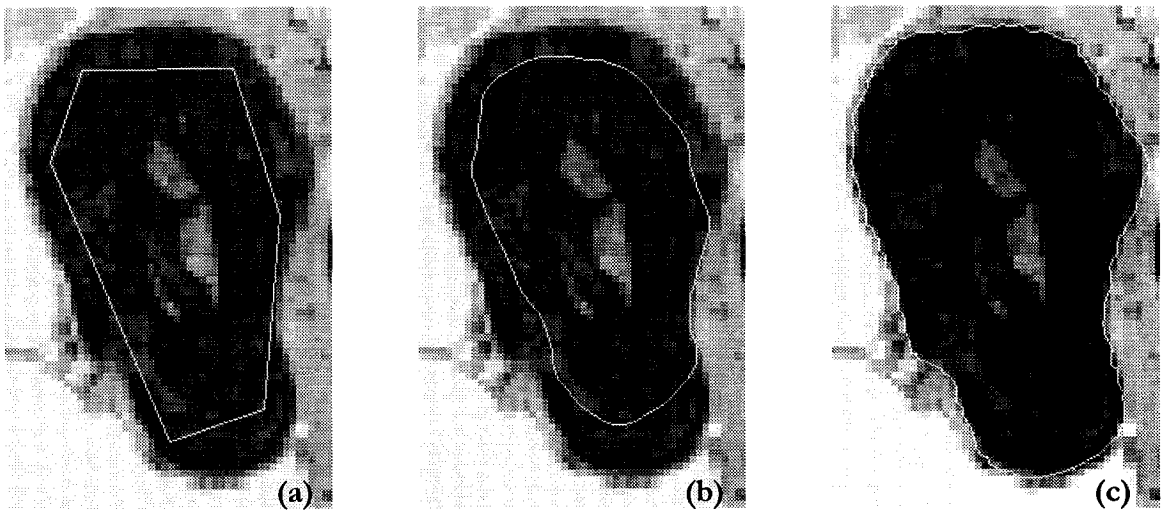


Figure 6.14: Comparison of active contour convergence for gradient and GVF. (a) An initial contour located too far from the boundary. (b) The result of using the gradient is that the active contour cannot converge to the boundary. (c) The result of using GVF that leads the active contour to converge to the boundary.

For some structures for which that the gradient gave good results, we used GVF to accelerate the convergence of the active contours to the boundary. For instance, we used the same initial contour comprised of four points for both parametric and discrete-dynamic

approaches. The structure used for this example was the malleus from slice 196, shown in Figure 6.13. The results, provided in Table 6.3, confirm that, in order to converge to the structure's boundary, active contours require smaller numbers of iterations when using GVF than when using the gradient method.

Approach type	No. of iterations for Gradient	No. of iterations for GVF ($\mu = 0.2$)
Parametric	3000	20
Discrete dynamic	2150	30

Table 6.3: A comparison between using the gradient alone and using GVF.

To further examine the capture range of GVF and the required number of iterations for parametric and discrete dynamic contours, simple triangular initial contours were used for all the selected structures. The triangle models were intentionally initialised far from the boundaries to examine the capture range of GVF, and inside the structures in order to protect the active contours from the effects of external forces related to the neighbouring regions. The same triangle was applied for both parametric and discrete-dynamic contour approaches. For most of the selected structures, however, these initial contours failed. This failure was related to the small high-contrast regions existing inside the structures of interest. Figure 6.15(a) shows examples of such small regions inside structures of interest, and the corresponding external forces are shown in Figure 6.15(b).

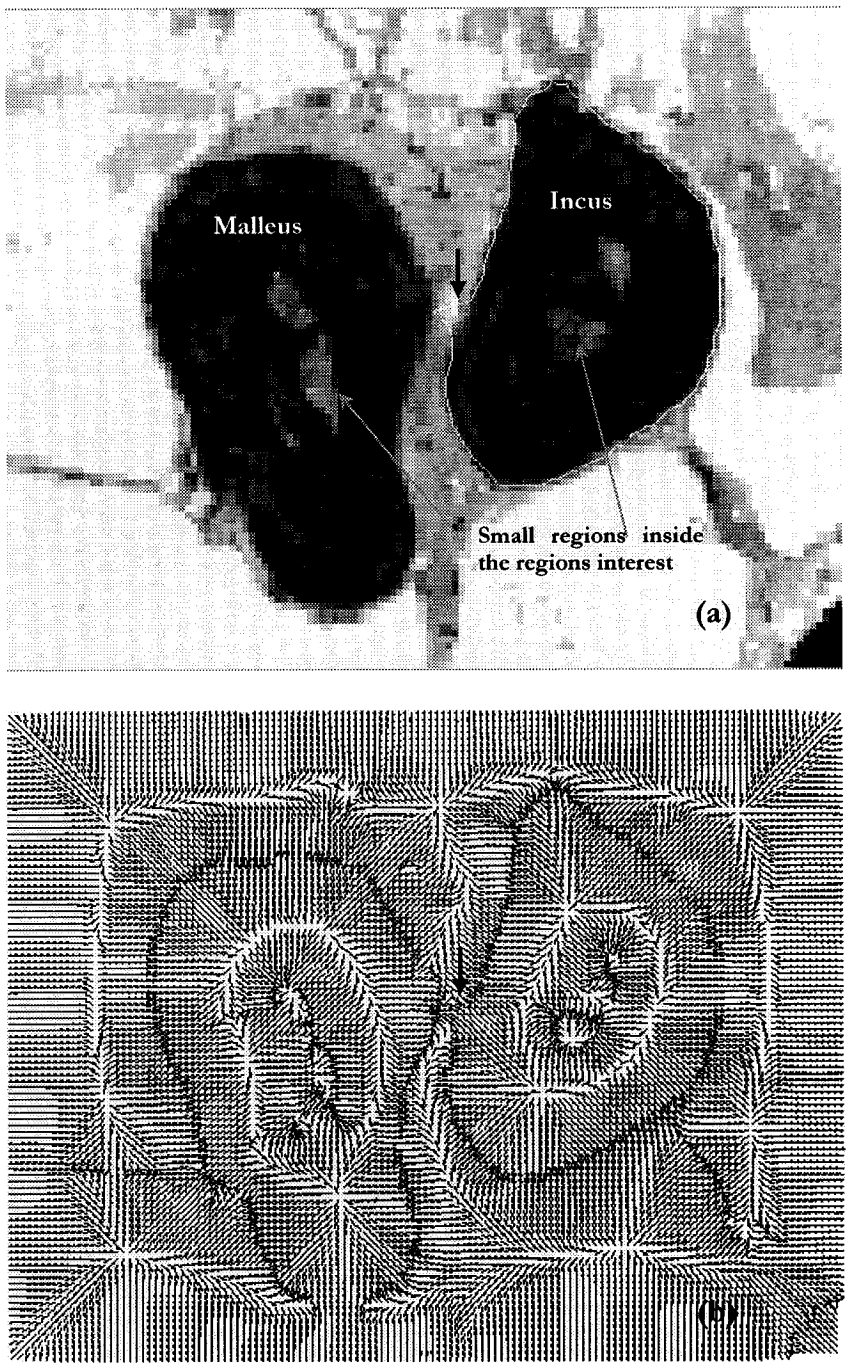


Figure 6.15: Example of a failure of convergence using GVF. Histological slice 246 shows the malleus and incus cross-sectional structures. Both structures contain small regions with high contrast (as marked) which produce false boundaries that attract the active contours. (a) An example of the active contour (white line) that overwhelms the boundary since the boundary of the incus is wrongly identified by GVF with μ=0.2 and (b) the external force map of (a) produced by GVF. The corresponding location of the wrongly identified boundary of the incus structure is shown by black arrows in Figures (a) & (b).

GVF produces external forces for these small regions. For example, if any vertex is located close to one of these small regions, the active contour converges to that region boundary and not to the desired boundary. Therefore, for structures containing small regions with high contrast, an appropriate initial model should contain as few points as possible, but it must be closer to the desired boundary than to the small-region boundaries. Figure 6.16 shows an example of an initial contour (white line) which succeeds for the malleus containing small regions.

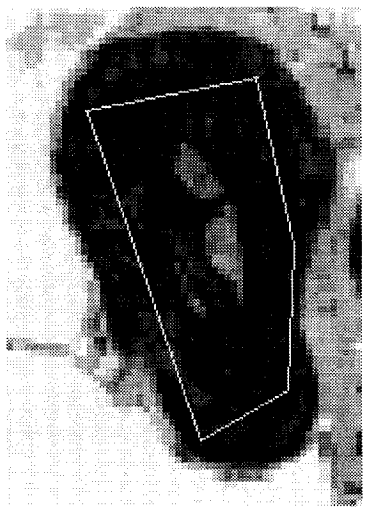


Figure 6.16: Example of initial contour containing high-contrast small regions. The active contour succeeded for the malleus in slice 246.

For both active-contour approaches, we first used μ=0.2, as suggested by Xu & Prince (1997). According to Equation 3.10, as μ is reduced, the effect of the Laplacian decreases and the gradient becomes dominant. We found that when μ is very small, *e.g.*, 0.02, the effect of GVF is not significantly different from that of the gradient alone. Therefore, we did not apply $\mu < 0.02$ since it will have the same limitations as the gradient method. For our experiments with GVF, μ was chosen to be as large as possible without causing wrong identification of the boundary.

Table 6.4 provides the information about the numbers of points of the initial models, the appropriate μ and the numbers of iterations required for convergence of parametric and discrete dynamic contours.

Slice number	Structure	No. of points for the initial contour	Discrete approach		Parametric approach	
			Appropriate μ weight	No. of iterations	Appropriate μ weight	No. of iterations
181	Malleus	**	failed		failed	
187	Malleus	**	failed		failed	
196*	Malleus	4	0.2	35	0.2	10
246*	Malleus	6	0.08	30	0.05	20
	Incus	5	0.08	15	0.05	20
266	Malleus	5	0.05	25	0.05	15
	Incus	7	0.15	75	0.08	80
306	Malleus	**	failed		failed	
	Incus	**	failed		failed	

Table 6.4: Applying GVF for discrete dynamic and parametric approaches. The structures marked with an asterisk are shown in Figures 6.17 & 6.18. ** represents all initial contours, even those initialised close enough to the boundary.

We found that GVF causes the boundaries of structures with low boundary contrast, *e.g.*, malleus in slice 181, to be wrongly identified and the final contour to rest in the wrong place. Figure 6.17(a) shows an example of a wrongly identified boundary, where the active contour has passed the actual boundary. Figure 6.17(b) shows the external force map produced by GVF. The corresponding location of the wrongly identified boundary is shown by black arrows in Figures 6.17(a) & (b).

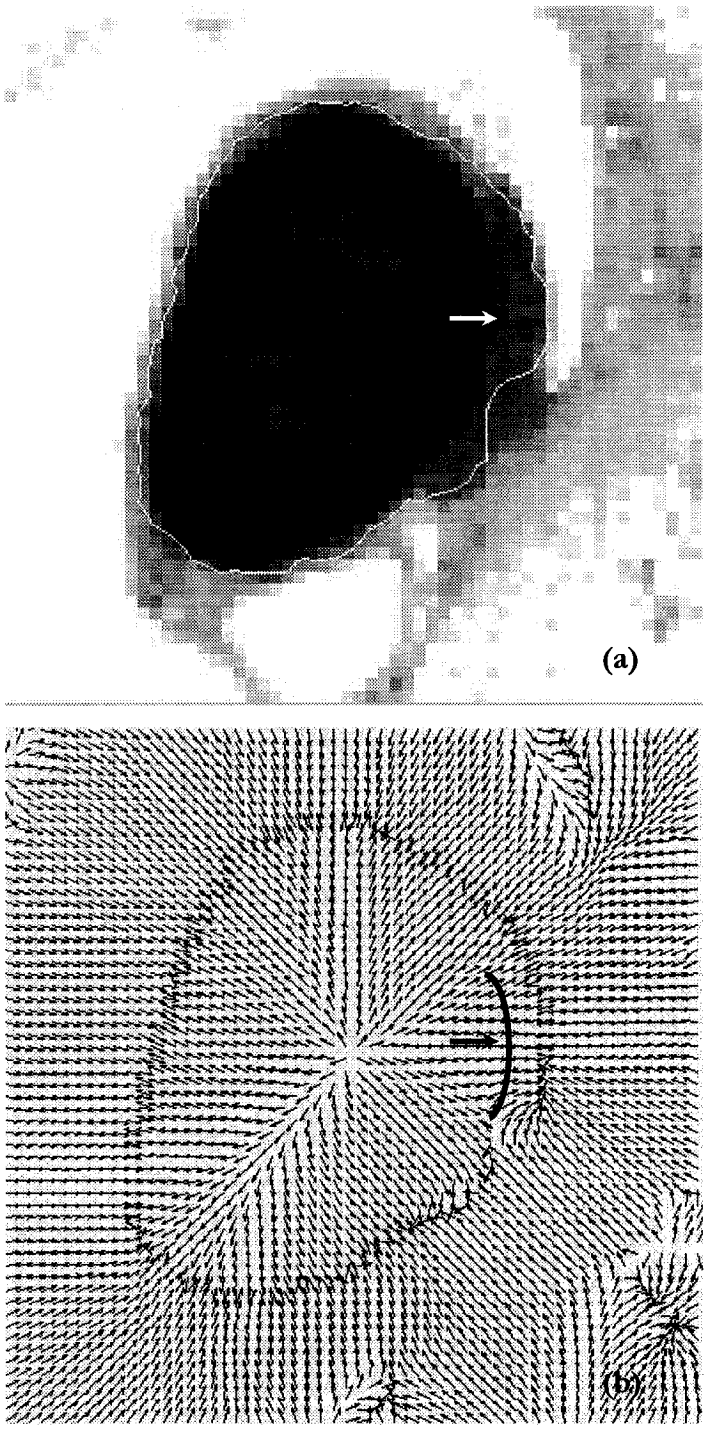


Figure 6.17: Example of the active contour passing the boundary when using GVF. Histological slice 181 shows the malleus cross-sectional structure. (a) The result of active contour (white line) that passes the right boundary due to the wrongly identified boundary by GVF with $\mu = 0.2$. (b) The external force map produced by GVF for $\mu = 0.2$; the actual location of the right boundary is shown by a black line. The arrows point to the location of the actual boundary.

We found that the segmentation results of both approaches were very similar. Figures 6.18 & 6.19 show examples for the initial contours and the final contours using discrete dynamic and parametric active contours. Compared with discrete dynamic contours, parametric active contours required mostly smaller numbers of iterations to converge to the boundaries of the regions of interest when using the same initial contour. We found that parametric and discrete-dynamic contours sometimes require different μ for satisfactory results.

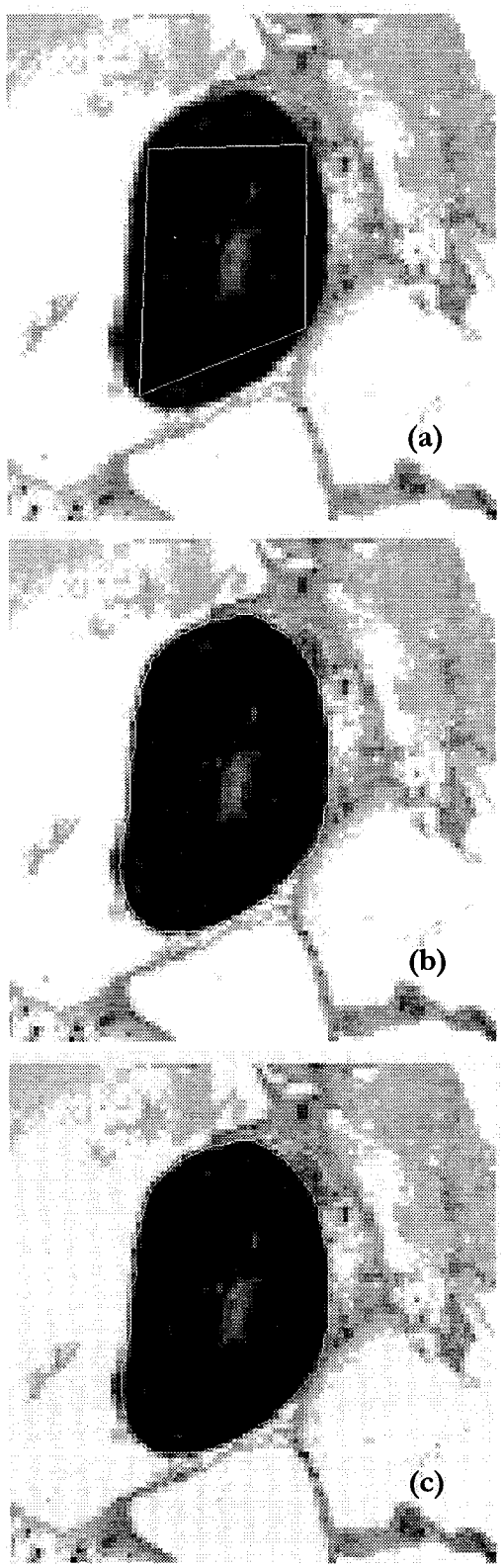


Figure 6.18: Example of parametric and discrete dynamic contours using GVF. The active contours were applied on the malleus in histological slice 196: (a) the initial contour (white line) composed of four points, (b) final contour using discrete dynamic contour and (c) final contour using parametric active contour.

Figure 6.19: Example of parametric and discrete dynamic contours using GVF. The active contours were applied on the incus from histological slice 246: (a) the initial contour (white line) composed of six points, (b) final contour using discrete dynamic contour and (c) final contour using parametric active contour.

For structures with very low boundary contrast, there were no appropriate μ, as marked 'failed' in Table 6.4. Even by applying μ=0.02 (the smallest applicable factor), the active contours of both approaches pass the boundaries. GVF was not a successful method for such structures. Figure 6.17(a) shows an example for a structure with low boundary contrast.

We also found that GVF may lead the active contours to wrongly identify the boundary regions between nearby structures even with high boundary contrast. GVF extends the influence of the stronger boundary over the weaker one. To prove that this issue was not related only to the low contrast of the weaker boundary, both structures were cut from the original image and the GVF with μ=0.2 was applied to them. The segmentation results were then successful for both separate structures. Figure 6.15 shows an example of such structures. The problem with wrongly identified boundaries is reduced when μ decreases. Figure 6.20 shows the external force map of Figure 6.15(a) produced by GVF with μ=0.2 and μ=0.05.

In an attempt to avoid the wrong boundary identification caused by GVF, we applied different combinations of the force weighting factors. We found that decreasing the external force weighting factor and increasing rigidity and tension weights for parametric contours, and internal and damping force weight for discrete dynamic contours, did not remove this issue. This is because the GVF leads the active contour to wrongly identify the boundary. Therefore, changing the external force weighting factor cannot correct the position of the boundary, and internal force has no control over it.

In conclusion, μ used for GVF computation depends on the boundary contrast of the region of interest. Since the contrast is not consistent within a slice, nor through slices, the appropriate μ must be chosen by trial and error. GVF fails for regions with low contrast. In addition, GVF may intensify the effects of the neighbouring regions within the vicinity of a few pixels; it is not a good method for the regions of interest in our histological dataset. GVF is good for simple images, but for complex images including regions with low contrast and/or weak edges, GVF may give undesired results.

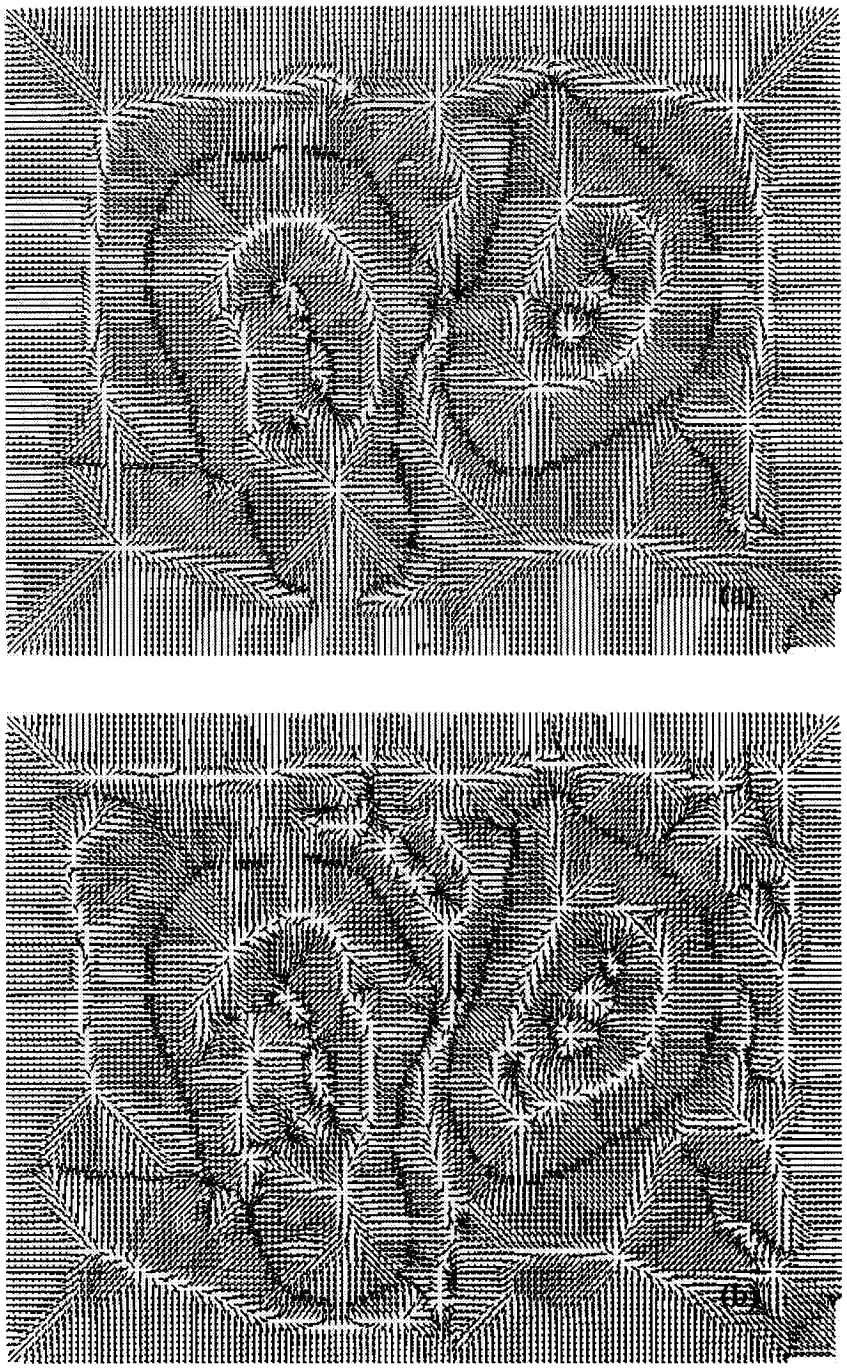


Figure 6.20: External force maps produced by GVF using different μ values (of the malleus from slice 196): (a) with μ = 0.2 , and (b) with μ = 0.05. The black arrows point the left boundary of the incus structure that is wrongly identified with μ = 0.2.

6.3.4 Gradient plus pressure

As an alternative approach to improving the capture range of the gradient method, we applied a pressure force, discussed in Section 3.2.3. The pressure force is independent of the image information, in contrast with the gradient and GVF that are computed based on the image information. In this section we evaluate the use of gradient plus pressure force as the external force for both parametric and discrete dynamic contours.

To compare the number of iterations required for active-contour convergence with and without pressure, an experiment similar to that of Table 6.3 (Section 6.3.3) was done for both active-contour approaches using the same initial contour. The structure used for this experiment was the malleus from slice 196. Table 6.5 shows the results of the experiment. The appropriate pressure weighting factor used for both approaches was 0.08. In this example, even the gradient alone is good for the segmentation but using the pressure makes the convergence faster.

Approach	No. of iterations for Gradient	No. of iterations for Gradient with pressure
Parametric	3000	130
Discrete dynamic	2150	195

Table 6.5: A comparison between using the gradient alone and the gradient plus pressure.

For our experiments with the gradient plus pressure, we used the same simple initial contours that were previously used for GVF experiments (Table 6.4) on the structures of interest. Table 6.6 shows the results for the appropriate pressure weighting factors and the numbers of iterations required for the convergence of the active contours. The criterion for choosing an appropriate pressure weighting factor was that it should be as large as possible without leading the active contour to overwhelm the boundary. As the initial contours are located inside the structures of interest, we applied a pressure force that inflates the active contours, *i.e.*, a positive pressure force.

Both parametric and discrete dynamic-contour approaches using gradient plus pressure give very similar boundary delineation of the structures of interest. Figure 6.21 shows the examples of the results of segmentation with discrete dynamic and parametric contours using gradient plus pressure.

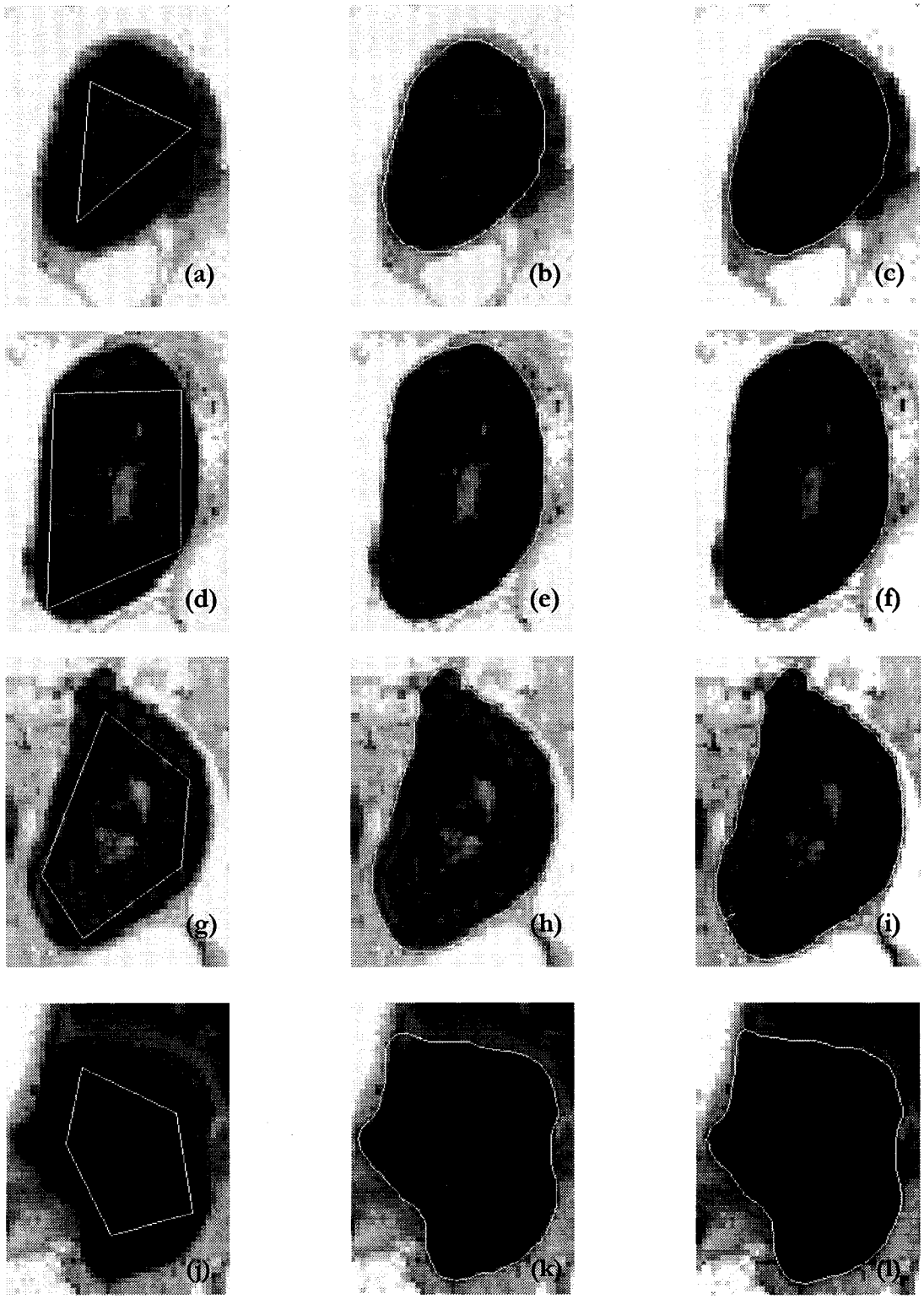


Figure 6.21: Examples of using gradient plus pressure. The middle column shows the discrete dynamic contours and the right column shows the parametric active contours: (a) the initial contour on the malleus from histological slice 181, (b) the final discrete dynamic contour and (c) the final discrete dynamic contour; (d) the initial contour on the malleus from histological slice 196, (e) the final discrete dynamic contour and (f) the final discrete dynamic contour; (g) the initial contour on the incus from histological slice 246, (h) the final discrete dynamic contour and (i) the final discrete dynamic; (j) the initial contour on the incus from histological slice 306, (k) the final discrete dynamic contour and (l) the final discrete dynamic contour.

Slice number	Structure	No. of points for the initial contour	Discrete dynamic approach		Parametric active approach	
			Optimal pressure weight	No. of iterations	Optimal pressure weight	No. of iterations
181*	Malleus	3	0.05	350	0.05	360
187	Malleus	3	0.05	650	0.01	2015
196*	Malleus	4	0.11	160	0.08	195
246*	Malleus	6	0.06	165	0.06	160
	Incus	5	0.06	325	0.03	360
266	Malleus	5	0.06	295	0.03	365
	Incus	7	0.04	410	0.03	390
306	Malleus	5	0.03	225	0.02	290
	Incus	8	0.02	975	0.01	1670

Table 6.6: Applying the gradient with pressure for discrete dynamic and parametric approaches. The structures marked with an asterisk are shown in Figure 6.21.

The comparison between Table 6.6 and Table 6.2 shows that for a particular pressure weighting factor, many more iterations were required when using the gradient with pressure than when using GVF. However, GVF failed for several structures while the gradient with pressure force was successful in all selected structures.

We found that the appropriate pressure weighting factor depends on the boundary contrast. This is similar to the situation with μ of the GVF, which also depends on the boundary contrast. For structures with low boundary contrast when using the gradient alone, by increasing the external force weighting factor we can intensify the boundary regions with low contrast. However, by increasing the external force weighting factor we increase the effect of false boundaries that are produced by the small high-contrast regions inside the selected structure. The same problem exists when using GVF. This is because the external force produced by the gradient or GVF depends on image information, so an increase of the external force weighting is applied to the whole image and can at the same time intensify the undesired edges that can attract the active contours. In contrast, since the pressure force is

independent of image information, by using pressure we may solve the problem that exists with false interior boundaries. This worked for all the cases we tried.

Oxiana permits the user to apply a negative or positive pressure, or to turn off the pressure, at any time. For example, if the user applies a positive pressure force and the active contour passes the boundary, the user can apply a negative pressure to drive the active contour back towards to the boundary, and then turn off the pressure and iterate the active contour again in order to converge to the boundary using gradient alone.

6.3.5 Summary

In the experiments on our histological dataset, we evaluated gradient, GVF and gradient-plus-pressure as the external force for the parametric and discrete dynamic approaches. The two active-contour approaches give similar boundary delineation of the structures of interest.

The gradient technique gives good results for all the selected structures, provided that the initial contour is located close to the boundary, since the capture range of the gradient is limited. Both GVF and the gradient plus pressure improve the limited capture range of the gradient. As a result a smaller number of iterations for active contour convergence is required when using GVF or the gradient plus pressure. When using the same initial contours, active contours require smaller numbers of iterations with GVF than with the gradient plus pressure.

For regions with low boundary contrast GVF may wrongly identify the boundary and lead the active contours to overwhelm the boundary. Also, GVF may lead the active contours to wrongly identify the boundary regions between nearby structures even with high boundary contrast. The gradient plus pressure succeeded for most regions which GVF technique failed for. Since the pressure force is independent of image information, by using pressure we may remove the issue that exists with false boundaries.

Both μ, used for GVF computation, and the pressure force weighting, used for the gradient plus pressure, depend on the boundary contrast of the region of interest, and since the boundary contrast is not consistent within a slice, nor through slices, the appropriate μ and pressure weighting factors must be chosen by trial and error.

6.4 Using active contours for segmentation on the MRM dataset

This section contains experiments with closed active contours on the MRM dataset. These experiments include comparative studies of parametric and discrete dynamic contours. The studies also include experiments with the gradient alone, with GVF and with the gradient plus pressure forces. To precisely compare the performance of the two active-contour approaches, the same initial contours were applied for both contour approaches.

The cross-sectional anatomical structures chosen for the experiments included bones (*e.g.*, malleus and incus), soft tissues (*e.g.*, tensor tympani muscle), and some cavities. It is important to note that MRM images have lower contrast and resolution than do histological images. We used structures with a variety of shapes and boundary contrasts since we were interested in evaluating the performance of the active contours on these structures. Figures 6.22 & 6.23 show the examples of the structures of interest.

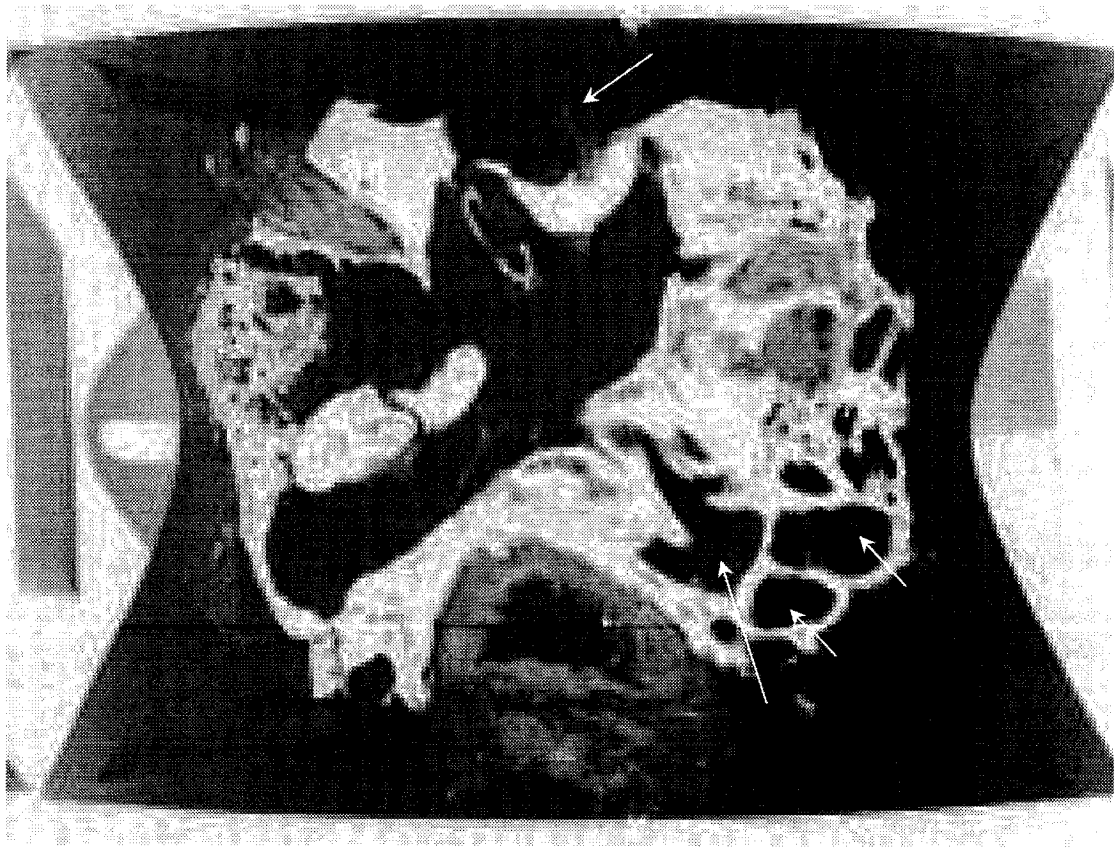


Figure 6.22: MRM slice 110 showing the malleus, incus and some cavities (as labelled).

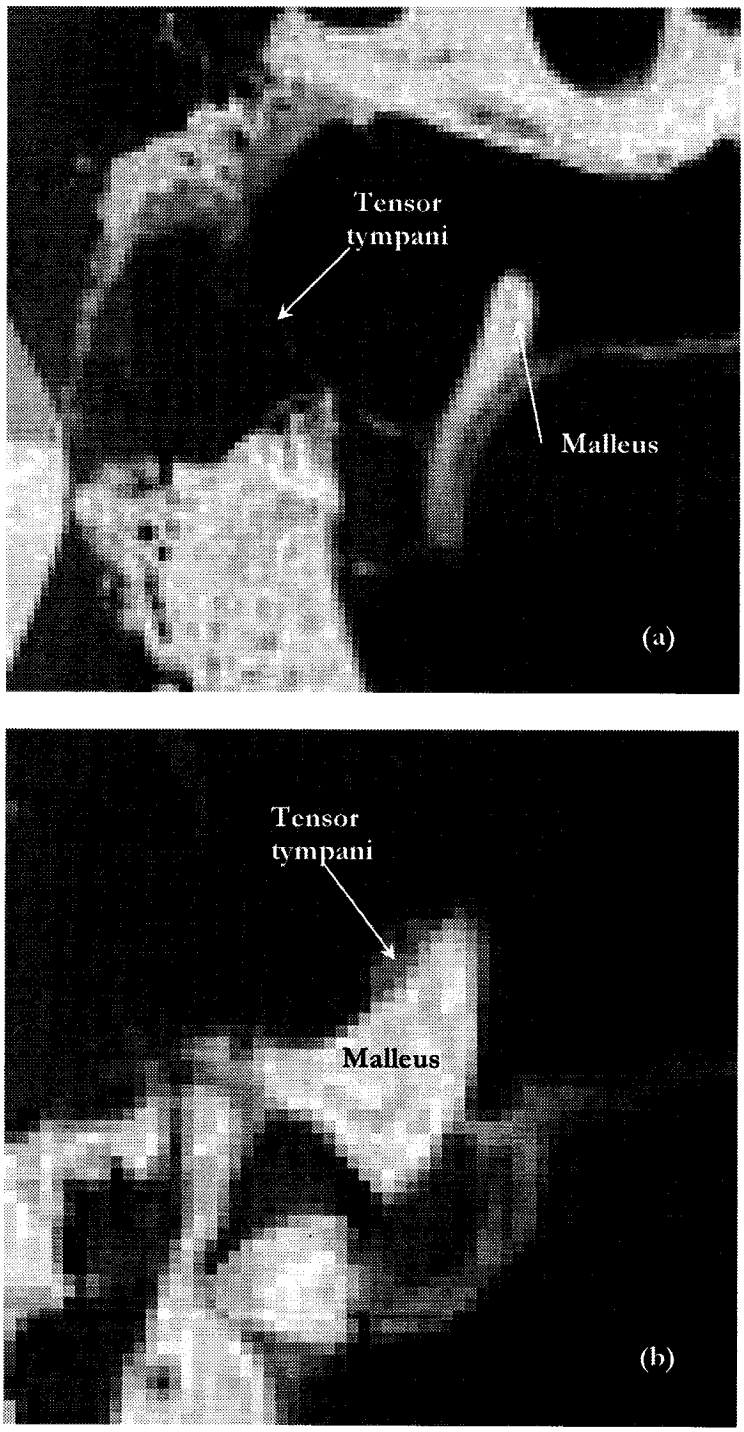


Figure 6.23: The malleus bone and tensor tympani muscle are illustrated in MRM slices: (a) 83 & (b) 90.

6.4.1 Gradient

In this section we use the gradient alone as the external force to evaluate the performance of parametric and discrete dynamic contours on our MRM dataset. Similar to our experiments on the histology dataset, the gradient alone gives good results provided that the initial contour is located close to the boundary.

For almost all the structures containing concavities (mostly narrow concavities), *e.g.*, cavity 3 in slice 110 (Figure 6.22), active contours could not converge to the concavities when using gradient alone. This is because the gradient produced force vectors that are normal to the edge surface and therefore for narrow concavities the external force cannot drive the active contour towards the concavity boundaries, as described in Section 3.3. This is shown in Figure 6.24.

Figure 6.24: Final contour using the gradient on Cavity 3 from slice 110. The active contour (white line) cannot converge to the narrow concavities using the gradient alone.

Table 6.7 shows the results of the experiments using the gradient for the two active-contour approaches on structures with low boundary contrast and/or containing concavities.

Slice number	Structure	No. of points for the initial contour	Discrete approach	Parametric approach
83	Malleus*	5	195	215
	Tensor tympani*	6	150	170
90	Malleus*	3	160	250
	Tensor tympani*	3	70	85
110	Incus*	4	80	110
	Malleus*	5	105	160
	Cavity 1	4	85	95
	Cavity 2	4	40	45
	Cavity 3*	13	35	40
	Cavity 4	Any initial contour even if located close to the boundary	Failed	Failed

Table 6.7: Applying gradient for discrete dynamic and parametric contours. For the structures marked with an asterisk, the external force weighting was increased to 4.0 to enhance the external force.

Parametric and discrete dynamic contours gave similar segmentation results. Figure 6.25 shows results of boundary delineation for some structures of interest using parametric and discrete dynamic contours.

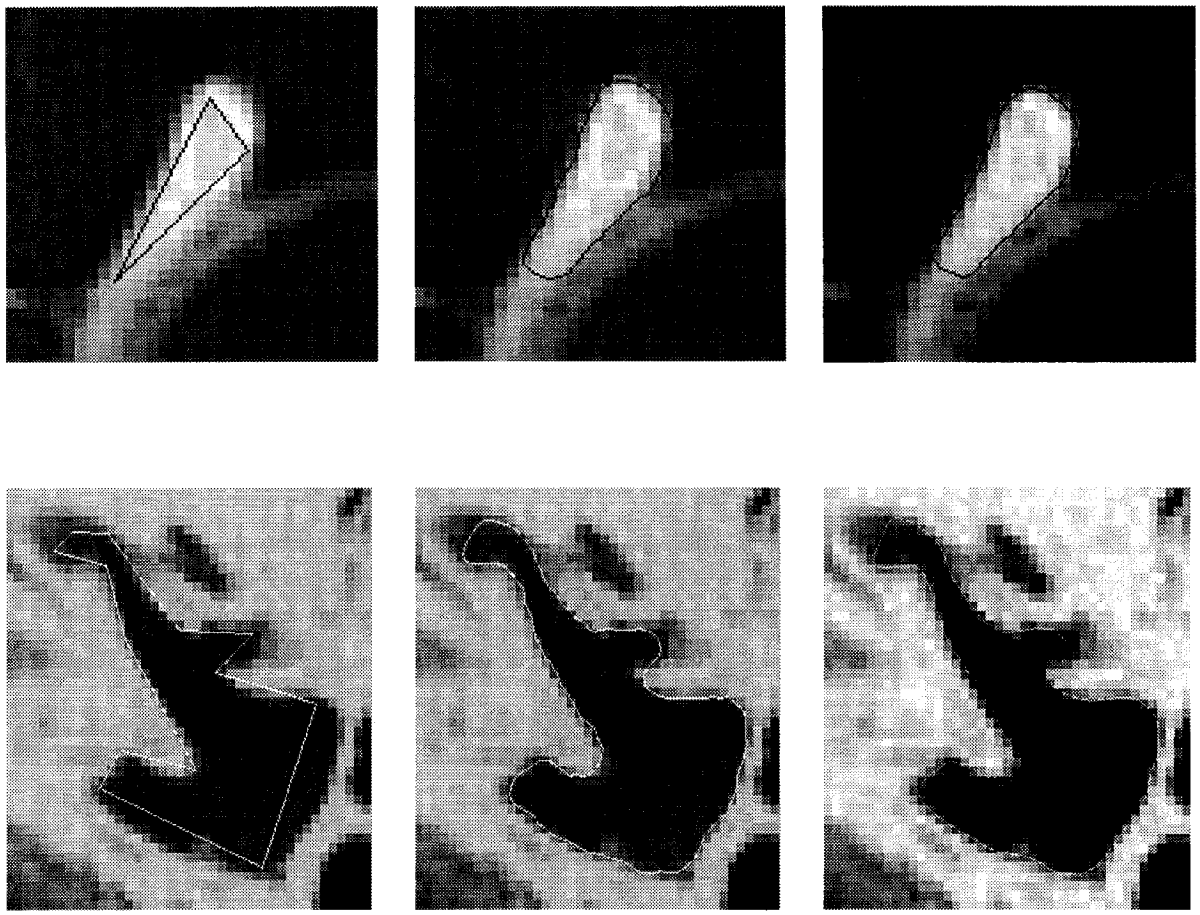


Figure 6.25: Boundary delineation with discrete dynamic and parametric active contours. The active contours were applied on the malleus from MRM slice 83: (a) initial contour (black line), (b) final contour with discrete dynamic contour and (c) final contour with parametric active contour. Cavity 3 from slice 110: (d) initial contour (white line), (e) final contour with discrete dynamic contour, and (f) final contour with parametric active contour.

For most of the selected structures, using the weighting combinations provided in Sections 6.1 & 6.2 resulted in active contours failing to converge to the parts of the boundary with low contrast and narrow concavities. By increasing the external force weighting factor to 4.0 we were able to the convergence failures The structures on which we used this combination are marked with an asterisk. In Table 6.7 cavities 1 and 2 did not require the external force weighting to be increased to 4.0, since their boundary contrast was high enough.

For the structure with the lowest boundary contrast, cavity 4, labelled as failed, increasing the external force weighting factor could not help the active contour converge to the boundaries.

Parametric and discrete dynamic contours had very similar delineation results for most structures. We found that parametric active contours required the starting contours to be located closer to the boundary than did discrete dynamic contours. Figure 6.26 shows an example of an initial contour that converges to the boundary using a discrete dynamic contour but fails when using a parametric active contour. In addition, parametric active contours required a larger number of iterations than discrete dynamic contours, for an identical initial contour model.

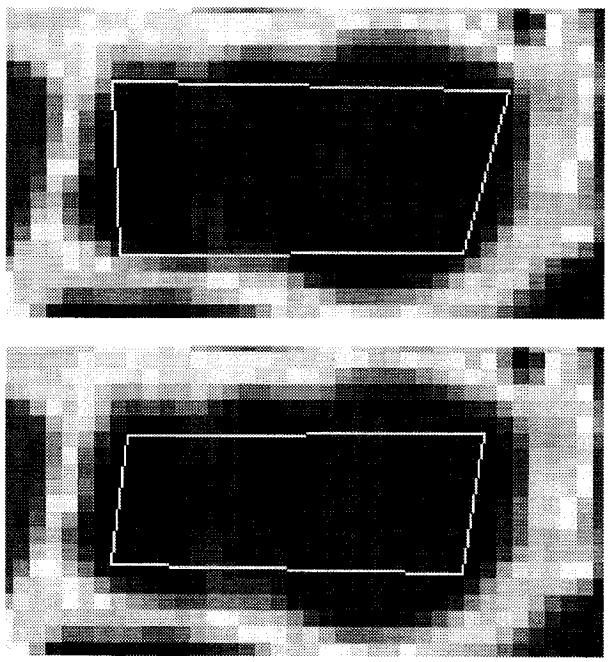


Figure 6.26: Examples of initial contours. The initial contour (white lines) is composed of four points on cavity 1 from slice 110, showing the sensitivity of parametric approach to the location of initial contours, compared with the discrete dynamic approach: (top) the initial converges to the boundary using both parametric and discrete dynamic approach, and (bottom) the initial contour is located farther from the boundary (compared with initial contour shown in (top)). This contour fails when using the parametric active contour and converges to the boundary when using the discrete dynamic approach.

6.4.2 GVF

In this section we evaluated GVF as the external force since the use of GVF attempts to overcome the capture-range and concavity limitations of the gradient method. We used the same initial contours in order to compare the two active contour approaches. The initial contours were simple triangles. These models were intentionally implemented far from the boundary to test the capture range of the GVF method. Figure 6.27 shows examples of triangle initial contours.

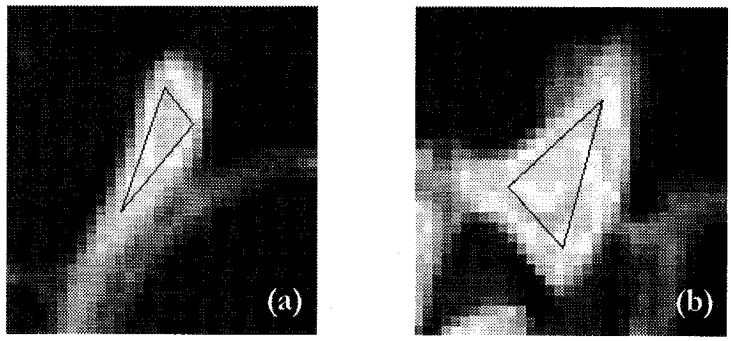


Figure 6.27: Examples of triangle initial contours. The initial contours (black triangles) used for parametric and discrete dynamic contours.

We started applying GVF with μ=0.2, as proposed by Xu & Prince. We found that this value leads the active contours to overwhelm the boundaries. Similar to our experiments with the histological dataset, using a μ less than 0.02 gives behaviour very similar to that of gradient alone, and therefore there is no advantage to using GVF with μ<0.02. For our experiments with GVF, μ was chosen to be as large as possible without its causing boundary misidentification. Table 6.8 shows the appropriate μ and number of iterations for some of the structures of interest with low boundary contrast and/or concavities.

Slice number	Structure	No. of points for the initial contour	Discrete approach		Parametric approach	
			Appropriate μ weight	No. of iterations	Appropriate μ weight	No. of iterations
83	Malleus	3	0.02	50	Failed	
	Tensor tympani	**	Failed		Failed	
90	Malleus	3	0.02	25	0.02	20
	Tensor tympani	**	Failed		Failed	
110	Incus*	**	Failed		Failed	
	Malleus*	**	Failed		Failed	
	Cavity 1	**	Failed		Failed	
	Cavity 2	**	Failed		Failed	
	Cavity 3	**	Failed		Failed	
	Cavity 4	**	Failed		Failed	

Table 6.8: Applying GVF for discrete dynamic and parametric contours. For the structures marked with an asterisk we increased the external force weighting to 4.0. ** represents for any initial contour even located close enough to the boundary.

No appropriate μ was found to give a good result for the structures labelled 'failed': all the values for μ led the active contours to overwhelm the boundaries. This was because GVF wrongly identified the boundaries and therefore active contours converged to the misidentified boundary. Figure 6.28 shows an example of a misidentified boundary produced by GVF and the results of the active contour.

In general, we found that only for a few selected structures, with high boundary contrast, could GVF give good results, and in these cases the appropriate values of μ are very small, *e.g.*, 0.02.

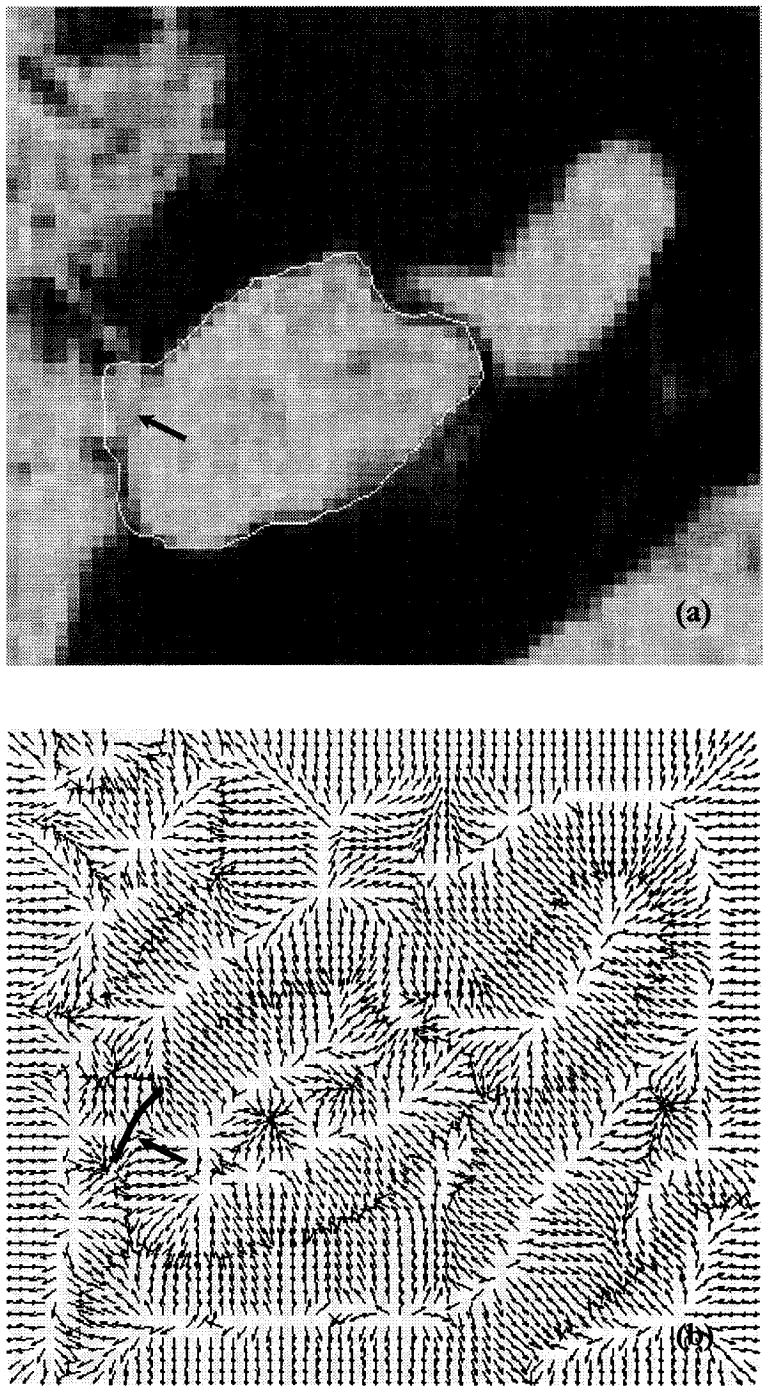


Figure 6.28: Segmentation of the malleus. The active contour, using GVF, was on slice 110 from the MRM dataset: (a) active contour overwhelms the left boundary and (b) the external force map of (a) produced by GVF (μ=0.02). The corresponding location of the misidentified boundary of the malleus is shown by black arrows in (a) & (b).

As with the histological dataset (Section 6.3.3), we found that GVF may lead the active contours to wrongly identify the boundary regions between nearby structures even with high boundary contrast, *e.g.*, cavities 2 & 3. This is an effect of a stronger boundary on the weaker one, because GVF extends the influence of the stronger boundary over the weaker one. To prove that this issue was not related to the contrast of the boundary, both structures were cut from the original image, then the GVF with μ=0.2 was applied to them and the segmentation results were successful for both separate structures. The wrongly identified boundary issue was generaly reduced when μ was decreased. Figure 6.29(a) shows an example of such structures, and the external force map of Figure 6.29(a) produced by GVF with μ=0.2 and μ=0.02 are shown in (b) & (c), respectively. However, in this example the active contour using μ=0.02 still fails to converge to the actual boundary.

In conclusion, μ used for GVF computation depends on the boundary contrast of the region of interest. The GVF technique failed for most MRM structures, since they generally have low boundary contrast. In addition, GVF may intensify the effects of the neighbouring regions within the vicinity of a few pixels. GVF is not a good method for the regions of interest in our MRM dataset.

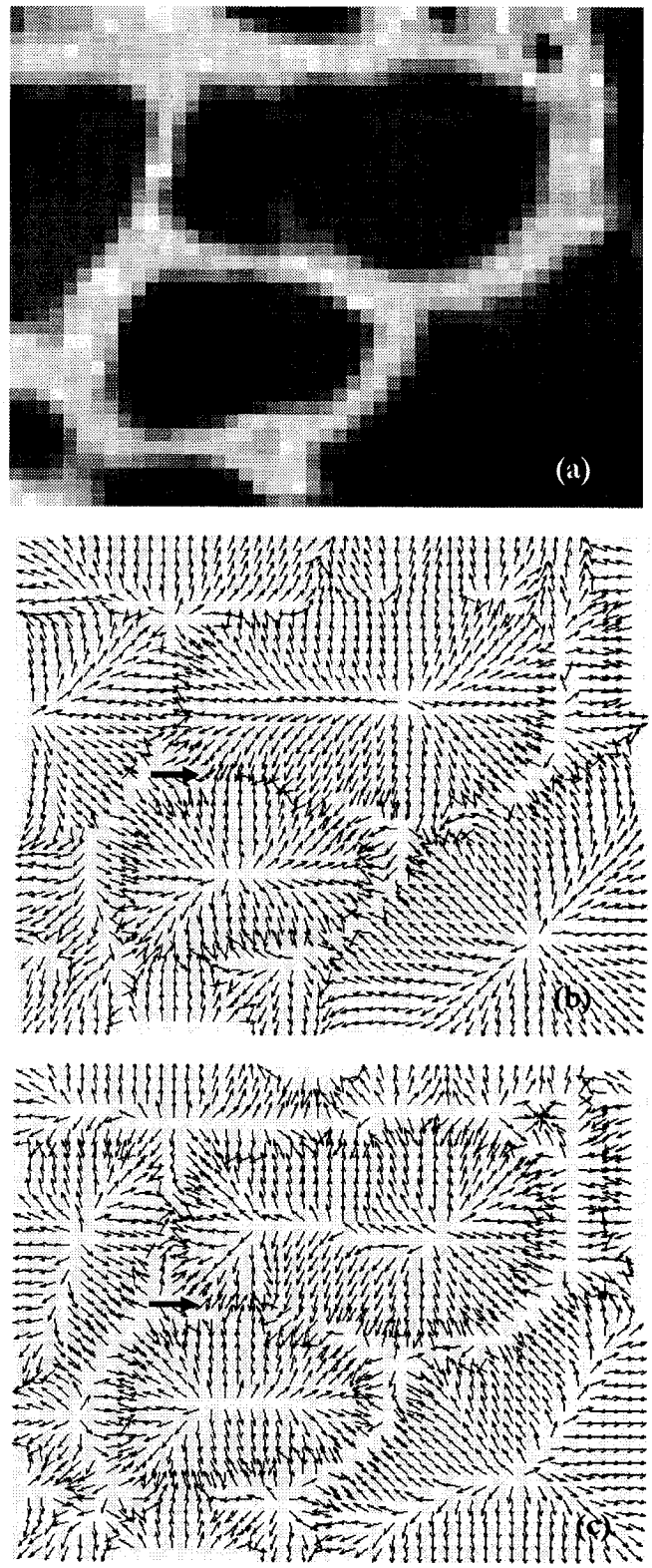


Figure 6.29: The external force maps produced by GVF using different μ. (a) MRM slice 110 shows cavities 2 & 3, (b) the external map produced by GVF with μ=0.2, (c) the external force map produced by GVF with μ=0.02. The neighbouring boundaries between the cavities are misidentified. This is because of the effect of a stronger boundary on the weaker one. The corresponding location of the misidentified boundary of the cavities is shown by black arrows on (a) & (b).

6.4.3 Gradient plus pressure

Similar to our experiments with histological images (Section 6.4.2), we applied a pressure force to improve the capture range of the gradient method and furthermore to compare it with GVF (Section 6.4.1) for the MRM dataset. To compare the number of iterations for active-contour convergence between parametric and discrete dynamic contours, the same initial contours used for GVF (Table 6.4) were applied with the pressure force. The initial contours were intentionally implemented far from the boundary to test the behaviour of active contours with an additive pressure force. Table 6.9 shows the results for the appropriate pressure weighting factors and the numbers of iterations required for the convergence of the active contours. The criterion for an appropriate pressure weighting factor was that the pressure weighting factor should be as large as possible but not lead the active contour to overwhelm the boundary. As the initial contours are located inside the structures of interest, we applied a pressure force that inflates the active contours, *i.e.*, a positive pressure force.

Slice number	Structure	No. of points for the initial contour	Discrete approach		Parametric approach	
			Optimal Pressure weight	No. of iterations	Optimal pressure weight	No. of iterations
83	Malleus	3	0.06	125	0.02	300
	Tensor tympani	**	Failed		Failed	
90	Malleus	3	0.04	160	0.03	100
	Tensor tympani	3	0.07	180	0.04	130
110	Incus*	3	0.06	285	0.05	205
	Malleus	3	0.03	270	0.03	235
	Cavity 1	3	0.05	230	0.04	150
	Cavity 2	3	0.08	125	0.07	80
	Cavity 3	**	Failed		Failed	
	Cavity 4	**	Failed		Failed	

Table 6.9: Applying the gradient with pressure for discrete dynamic and parametric contours. For the structure marked with an asterisk we increased the external force weighting to 4.0. ** indicates that no initial contour succeeded even when located close to the boundary.

Both parametric and discrete dynamic approaches using the gradient plus pressure give similar boundary delineations of the structures of interest. Figure 6.30 shows examples of boundary delineation using both parametric and discrete dynamic contours.

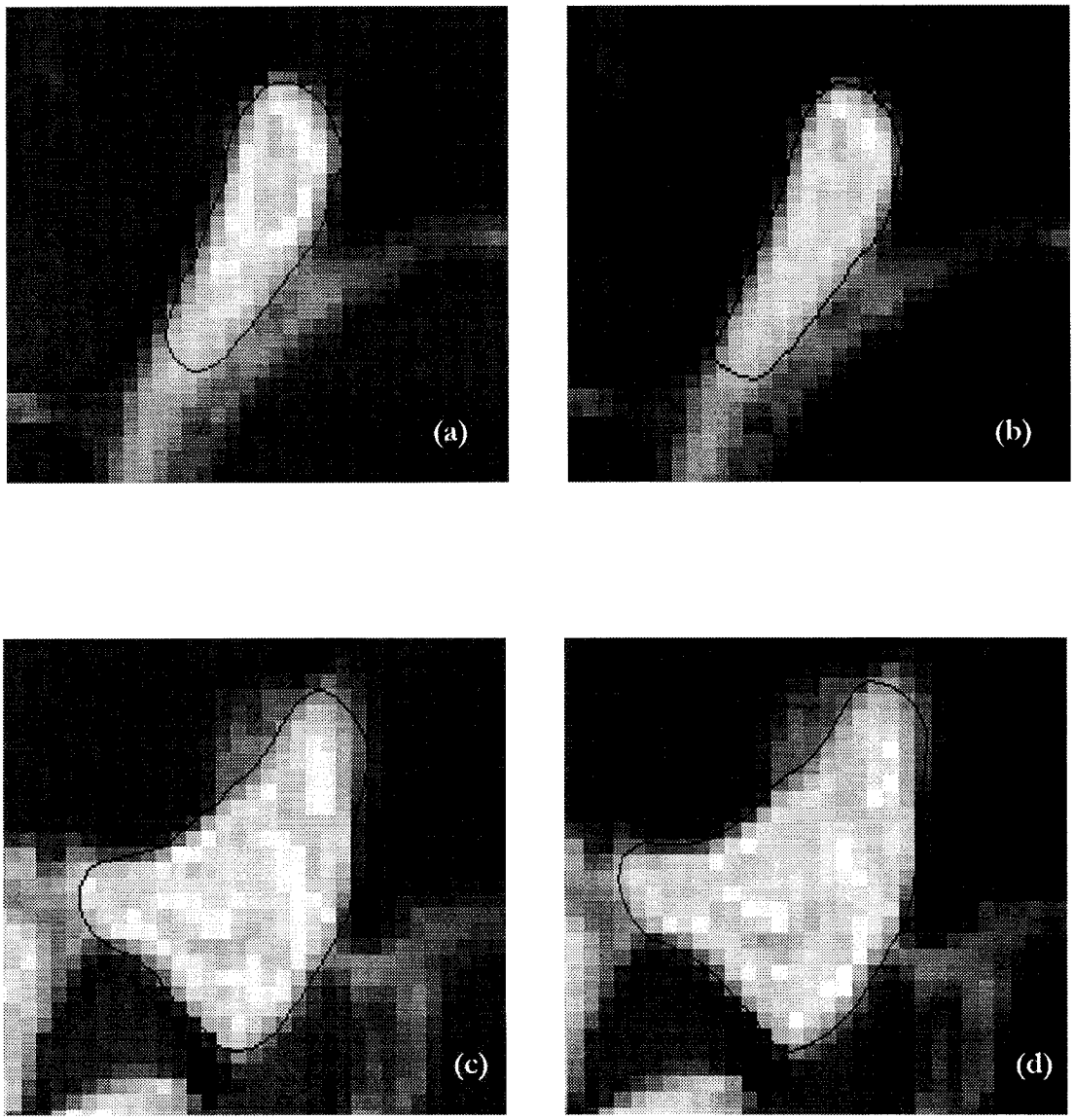


Figure 6.30: Examples of using gradient plus pressure on malleus. The malleus structures were from slices 83 (a & b) and 90 (c & d). The triangular initial contours are shown in Figure 6.21. (a) Result of using discrete dynamic contour, (b) result of using parametric active contour, (c) result of using discrete dynamic contour and (d) result of using parametric active contour.

The comparison between Tables 6.8 & 6.9, *i.e.*, results of segmentation using GVF and using gradient with pressure, shows that the pressure force worked successfully for most structures that GVF failed for. With the pressure force, the parametric active contours require a smaller pressure weighting factor and a smaller number of iterations than discrete dynamic contours do.

For only one structure with weak boundaries, labelled with an asterisk, *i.e.*, incus in slice 110, for which the pressure force leads the active contours to overwhelm the boundaries, increasing the external-force weighting intensifies the weak boundaries and prevents the active contours from overwhelming the boundaries. For this structure the appropriate external-force weighting was found to be 2.0.

Pressure force works successfully for the structures with concavities such as the malleus in slices 83 and 90. However, for almost all narrow concavities such as cavity 3, active contours of both approaches cannot converge to the concavity for an appropriate pressure weighting factor, and increasing the external force weighting factor is not effective in these cases.

Similar to our experiments with gradient plus pressure on the histological dataset (Section 6.3.4) we found that the appropriate pressure weighting factor depends on the boundary contrast. And similar to experiments with GVF (Section 6.4.3) the gradient plus pressure failed for the structures with low boundary contrast.

6.4.4 Summary

In the experiments on our MRM dataset, we evaluated the gradient, GVF and the gradient plus pressure as the external force for parametric and discrete dynamic approaches. The two active contour approaches give similar boundary delineation of the structures of interest. However, when using GVF and the gradient plus pressure, parametric active contours required smaller numbers of iterations for active contour convergence compared with the discrete dynamic approach.

The gradient technique gives good results for all the structures of interest, provided that the initial contour is located close to the boundary since the capture range of the gradient is limited. We found that parametric contours using the gradient require the initial contour to be closer to the boundary than do discrete dynamic contours, and for the same initial contour the parametric contours required a larger number of iterations for active contour convergence than do discrete dynamic contours.

GVF failed for most of the structures of interest, since MRM structures have low boundary contrast, and it was successful only for structures with high boundary contrast. However, GVF may lead the active contour to wrongly identify the boundary regions between nearby structures even with high boundary contrast. We found that for structures with low boundary contrast and/or containing concavities, the discrete dynamic approach is more successful than the parametric approach. However, for structures with high boundary contrast parametric active contours require smaller numbers of iterations than do discrete dynamic contours.

The gradient plus pressure worked successfully for most structures, especially those containing concavities, that GVF failed for. When using the gradient plus pressure, the parametric active contours require a smaller number of iterations than discrete dynamic contours do.

GVF and the gradient plus pressure both accelerate the convergence of the active contours to the boundary, but in equal conditions (a structure with the same initial contour), active contours using GVF converge more quickly to the boundary than do those using the gradient plus pressure. It is important to note that the gradient, GVF and the gradient plus pressure all failed for some structures with low boundary contrast and/or narrow concavities.

Similar to our experiments with histological dataset (Sections 6.3.2 & 6.3.3), the factor μ

(used for GVF computation) and the pressure force weighting (used for the gradient plus pressure) depend on the boundary contrast of the structure. The appropriate μ and pressure weighting factors must be chosen by trial and error because the boundary contrast is not consistent within a slice, nor through the slices.

6.5 Using open contours for segmentation

A closed contour completely encloses the region which corresponds to a structure. The use of open contours (in addition to the usual closed contours) facilitates the controlled handling of complex structures with shared surfaces or of thin structures.

The structures we used for segmentation with open contours were the shared surfaces and thin structures. The contrast of these regions is neither consistent within a slice, nor through the slices. This problem is especially present in the MRM dataset where the resolution and contrast of the images is not high enough to have a clear discrete view of the different structures.

In the experiments with closed contours we found that the gradient alone and the gradient with pressure force give better segmentation results than GVF. Therefore for open contours, we applied only the gradient alone and the gradient with pressure.

6.5.1 Histological dataset

Open contours were used to segment the shared boundaries between bones (*e.g.*, incus and malleus) and the soft tissues (*e.g.*, ligaments). Figures 6.31 and 6.32 show examples of such structures.

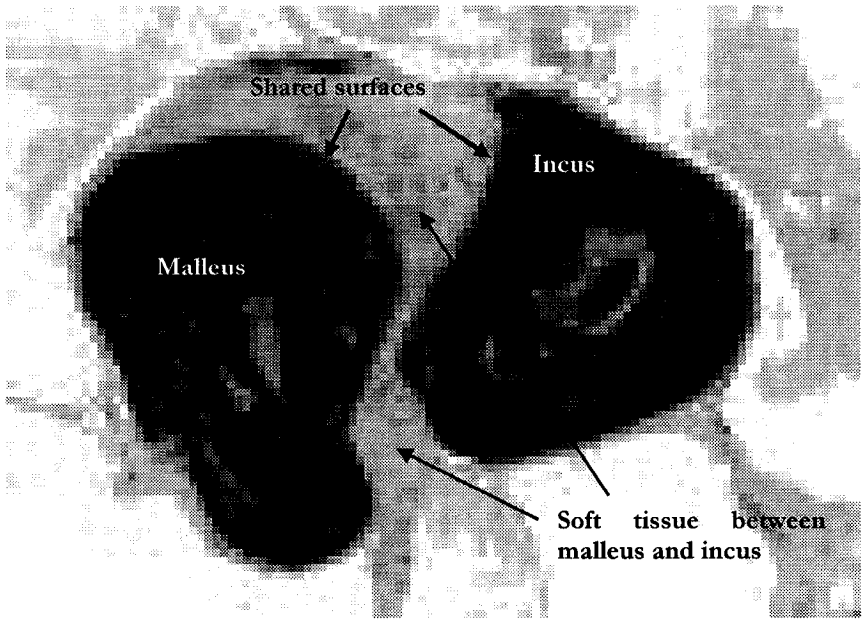


Figure 6.31: Histological image data showing the malleus and incus. The structures are in Slice 251. The soft tissue between the two bones is shown.

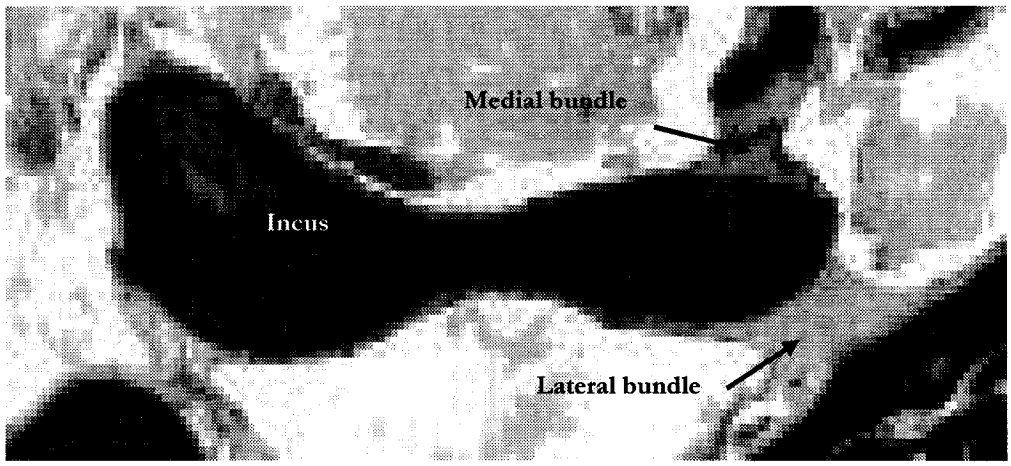


Figure 6.32: Histological image data showing the incus. The structure is in Slice 346. The lateral and medial bundles of the posterior incudal ligament are shown.

We first applied the gradient alone as the external force for the open contours. A pressure was then used to improve the capture-range of the gradient technique for the cases where the initial contour was not located close enough to the boundary. For other initial contours that were located close enough to the boundary, pressure force was used to accelerate the convergence of the active contour. Figure 6.33(a) shows an example of an initial contour located not close enough to the boundary and the results when using the gradient alone (b) and when using the gradient plus pressure (c).

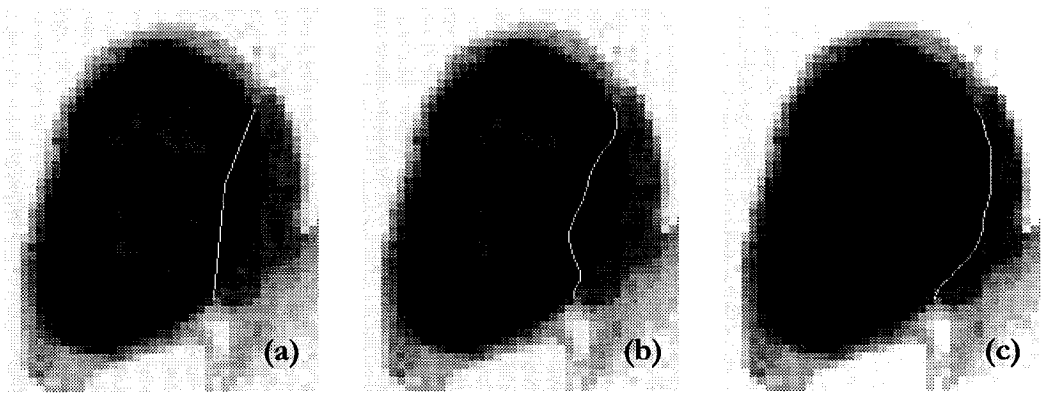


Figure 6.33: Open contours using gradient and gradient plus pressure. (a) Initial contour located far from the boundary, (b) the final contour using the gradient which does not converge to the boundary and (c) the final contour using the gradient plus pressure that converges to the boundary.

The criterion for choosing the appropriate pressure weighting factor is as described in Section 6.3.4. Table 6.10 provides the details of using the gradient alone and the gradient plus pressure for the segmentation of the shared surfaces of interest in several slices.

Slice number	Structure	No. of points for the initial contour	Gradient alone	Gradient plus pressure	
			No. of iterations	appropriate pressure weighting	No. of iterations
251*	The surface between malleus and the soft tissue between malleus & incus	3	595	0.1	80
	The surface between incus and the soft tissue between malleus & incus	4	755	0.08	75
286*	The surface between malleus and the soft tissue between malleus & incus	3	1945	0.03	270
	The surface between incus and the soft tissue between malleus & incus	3	545	0.02	235
341	Lateral bundle of the posterior incudal ligament	3	1055	0.03	170
346*	Lateral bundle of the posterior incudal ligament	3	395	0.02	270
	Medial bundle of the posterior incudal ligament	3	105	0.03	90
351	Lateral bundle of the posterior incudal ligament	3	220	0.03	80
	Medial bundle of the posterior incudal ligament	3	90	0.03	30

Table 6.10: Applying gradient alone and gradient plus pressure for discrete dynamic open contours on the histological dataset. The contour initialisation and final discrete dynamic and parametric contours are shown in Figure 6.27 for structures marked with an asterisk.

Simple initial contours including three vertices were initialised for the segmentation of the shared surfaces of interest. These initial contours were not successful for all the surfaces. This was because some false edges (produced by the small regions existing in the structures of interest as described in Section 6.3.3) were located in the vicinity of the initial contours and would prevent the active contour from converging to the desired boundary. In one case we had to use an initial contour with one extra point to locate the open contour closer to the desired boundary and farther from the false edge.

Figure 6.34 shows examples of the initial contours and the results of segmentation with open contours using the gradient alone and the gradient plus pressure. Open contours using the gradient alone and the gradient plus pressure give almost the same results. The gradient alone required more numbers of iterations for the open contour convergence to the boundary. Using the gradient plus pressure removes the capture-range limitation that exists with the gradient alone; and therefore, the initial contour can be located farther from the boundary than the initial contour used for the gradient alone.

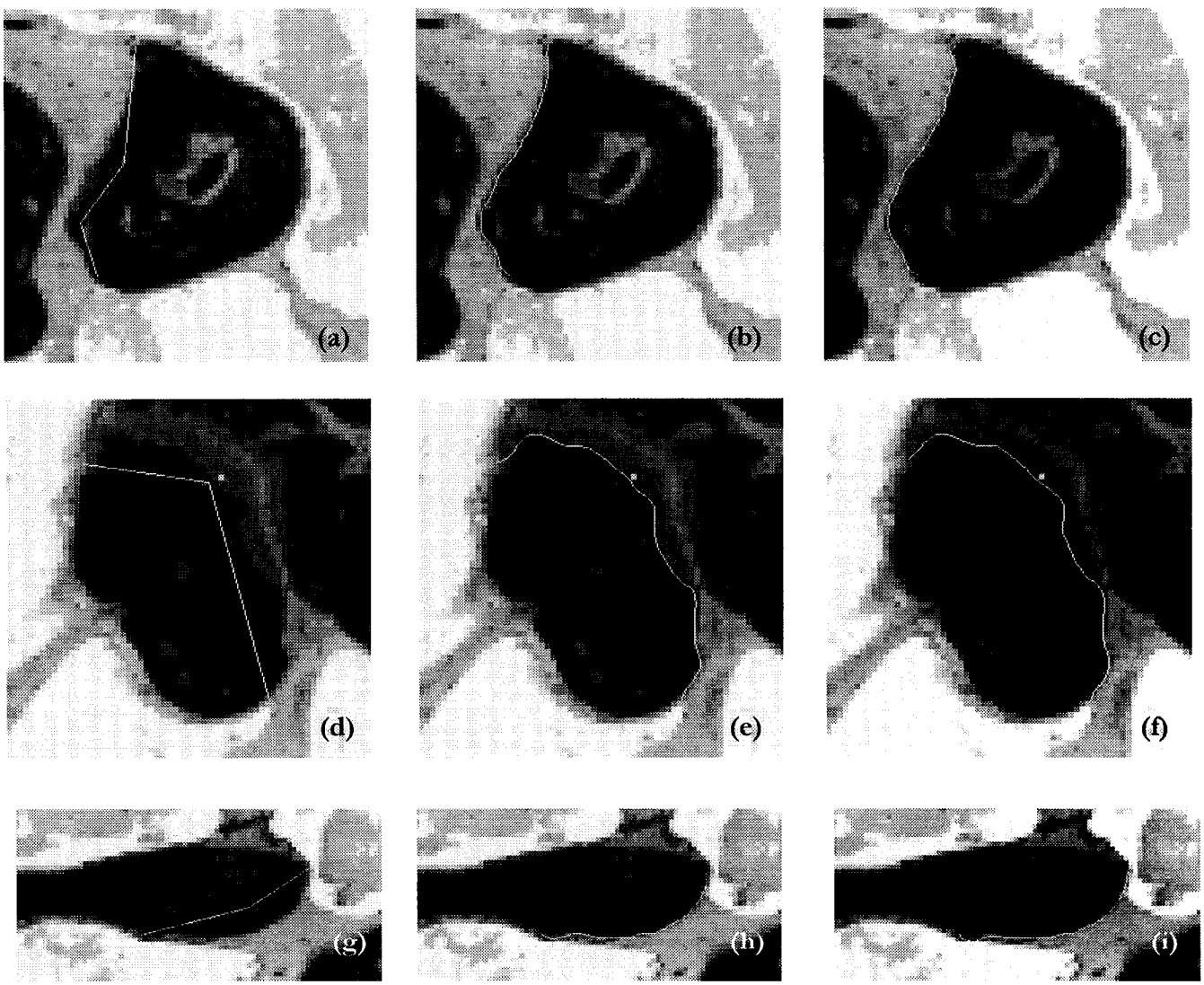


Figure 6.34: Examples of open contours using gradient alone and gradient plus pressure: (a) the initial contour on the slice 251 (the incus shared surface), (b) the final contour using the gradient, and (c) the final contour using the gradient plus pressure; (d) the initial contour on the slice 286 (the malleus shared surface), (e) the final contour using the gradient and (f) the final contour using the gradient plus pressure; (g) the initial contour on the slice 346 (the incus shared surface with later posterior ligament), (h) the final contour using the gradient, and (i) the final contour using the gradient plus pressure.

We were also interested in using open contours for segmentation of thin structures such as the eardrum. Figure 6.35 shows examples of the eardrum structure.

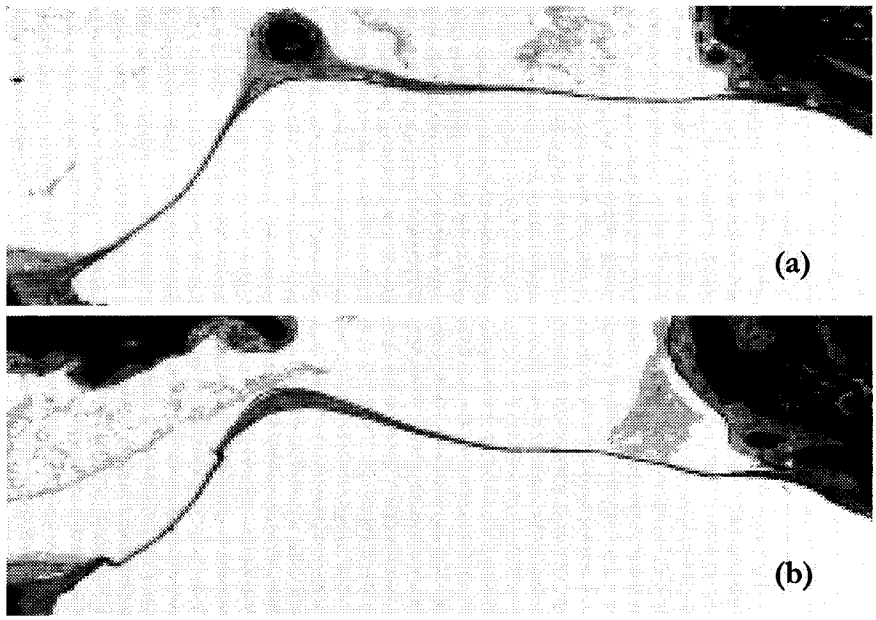


Figure 6.35: Examples of the eardrum. The structures were selected from (a) histological slice 506, and (b) histological slice 551.

Segmentation of very thin structures such as the eardrum requires that the open contour detect the middle pixel of the structure rather than the exterior pixels. The gradient alone was not successful to detect the central pixel along the length of the eardrum structure. As a result the open contour using the gradient often jumps to the other side of the edge. Figure 6.36(a) & (c) shows examples of this problem. The magnified versions of the regions of interest, showing that the active contour leaped to the other side of the structure, are shown in (b) and (d) for (a) and (c), respectively. To solve this issue an algorithm that detects the centre of the structure should be used to compute the external force for open contours.

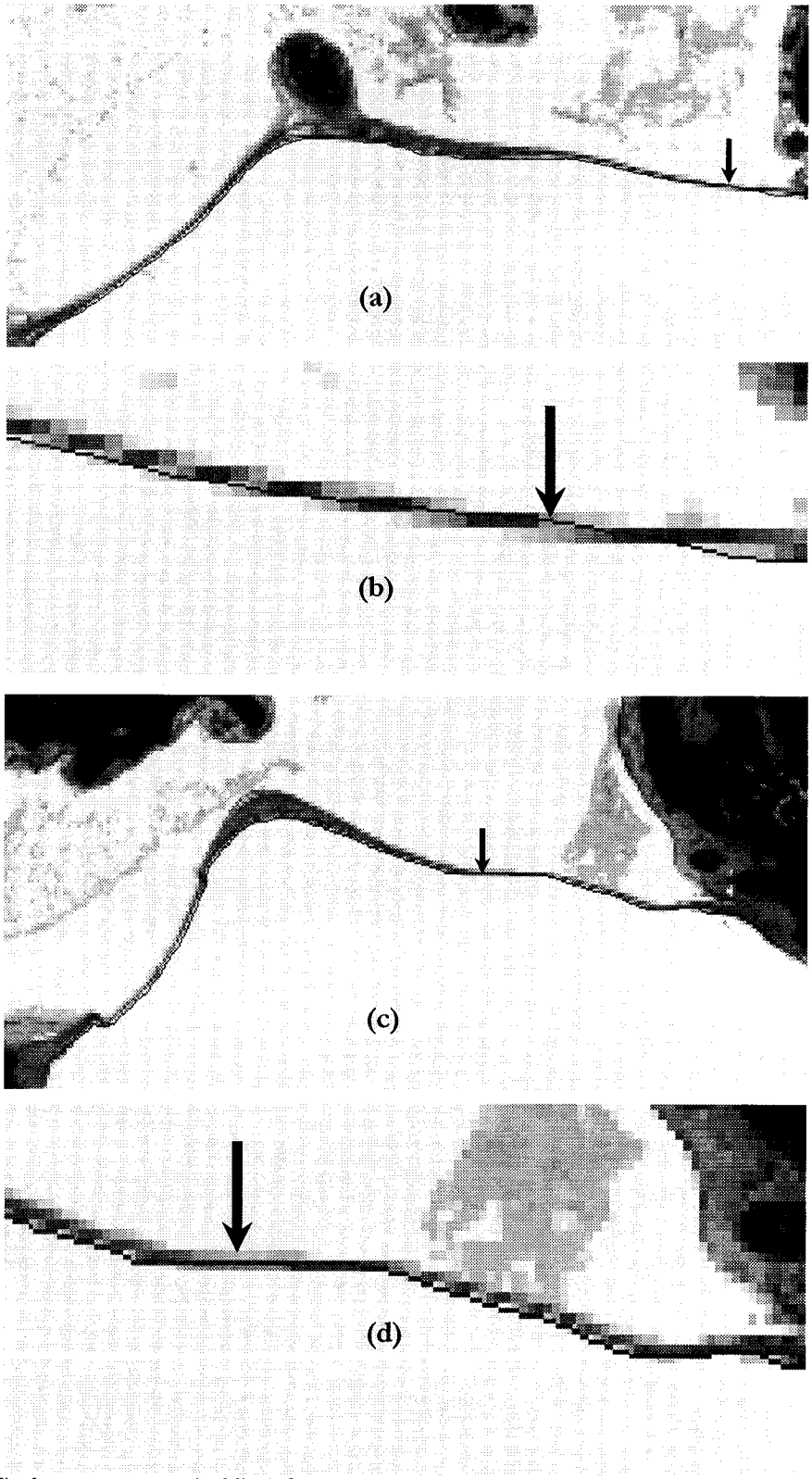


Figure 6.36: Final open contours (red line) for eardrum structure segmentation: on histological slice 496 (a) and slice 551 (c). (b) & (d) are the magnified versions of the regions of interest from (a) and (c), respectively. The black arrows point to part of the structures where the open contour jumps from one side of the edge to the other side because of the edge detector.

In our experiments with open contours on the histological dataset, we found that both the gradient alone and the gradient plus pressure give good results for segmentation of the shared surfaces between structures. However, since pressure force removes the capture-range limitation which exists with the gradient method, we found that when using the gradient plus pressure the initial contour can be located farther from the boundary than when using the gradient alone. Also, a smaller number of iterations is required for open contours to converge to the boundary when using the gradient plus pressure. For segmentation of thin structures such as the eardrum, we found that the gradient is not an appropriate technique since it only detects the exterior pixels along the structure's length while the central pixels are required for this purpose. In some cases, since the contrast of the eardrum varies along its length, the gradient cannot even properly detect the exterior pixels and it may cause the open contour to jump from one side to the other side of the structure.

6.5.2 MRM dataset

Open contours were used to segment the shared boundaries between bones and soft tissues. The shared surfaces we used for our experiments were the bony tympanic ring, shown in Figure 6.37(a), the shared surface of the incus and malleus with the soft tissue located between them, shown in Figure 6.37(b), and the shared surface between incus and the posterior incudal ligament, shown in Figure 6.37(c). MRM slices have generally lower contrast than the histological dataset. Therefore, the shared surfaces of interest are better seen in the histological data than in the MRM dataset.

Simple initial contours including three vertices were initialised for our experiments. The simple initial contours were successful for all our desired shared surfaces.

We first applied the gradient alone as the external force for the open contours. A pressure was later used to accelerate the active contour convergence to the boundary. The criterion for choosing the appropriate pressure weighting factor is described in Section 6.43.

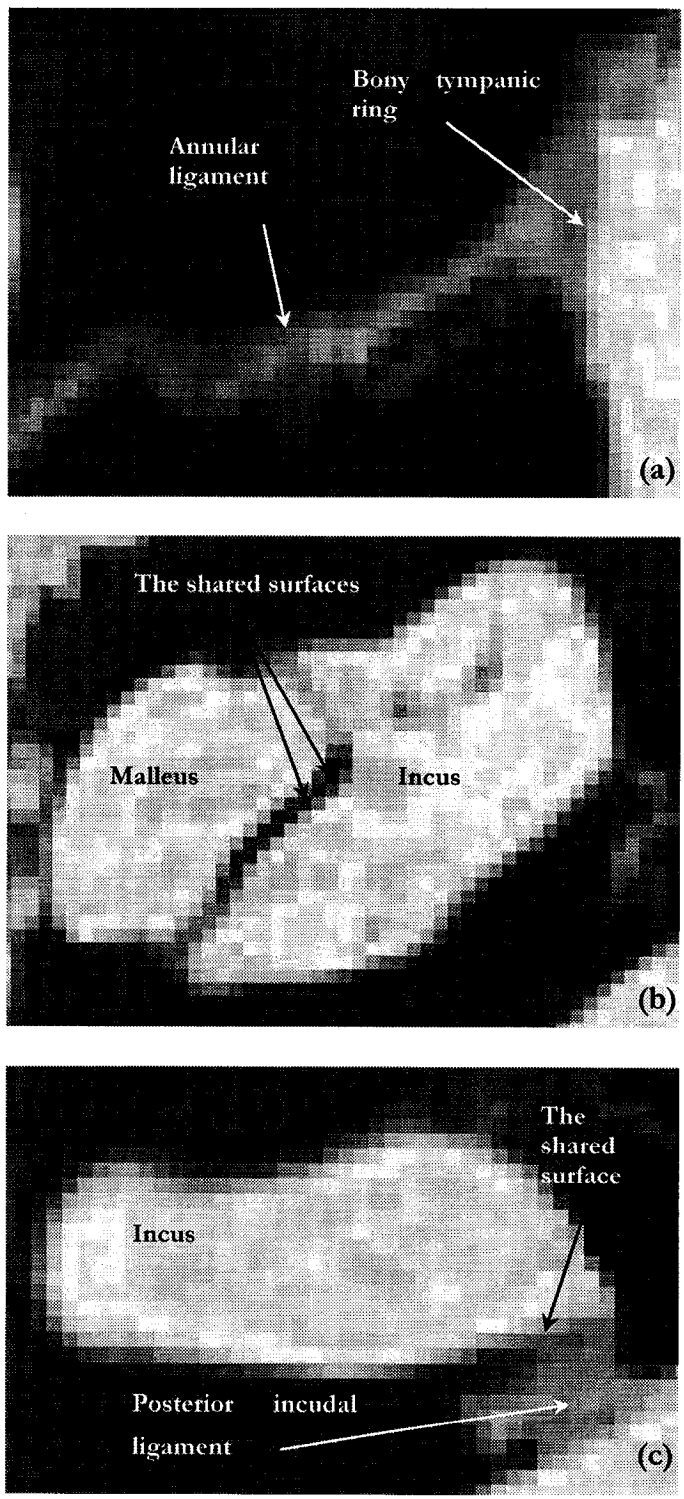


Figure 6.37: Examples of shared surfaces: (a) bony tympanic ring from slice 85, (b) between bones (incus & malleus) and the soft tissue located between them from slice 115, and (c) between the incus and the posterior incudal ligament from slice 129.

Table 6.11 provides the details of applying the gradient alone and the gradient plus pressure for the segmentation of the shared surfaces of interest in several slices.

Slice number	Structure	No. of points for the initial contour	Gradient alone	Gradient plus pressure	
			No. of iterations	appropriate pressure weighting	No. of iterations
85*	The bony tympanic ring	3	100	0.02	55
88	The bony tympanic ring	3	85	0.02	45
115*	The surface between incus & the soft tissue between malleus & incus	3	55	0.03	25
	The surface between malleus & the soft tissue between malleus & incus	3	90	0.03	45
118*	The surface between incus & the soft tissue between malleus & incus	3	100	0.03	45
	The surface between malleus & the soft tissue between malleus & incus	3	110	0.03	50
128	Shared surface between incus & the posterior incudal ligament	3	15	0.02	8
129*	Shared surface between incus & the posterior incudal ligament	3	15	0.02	8

Table 6.11: Applying the gradient alone and the gradient with pressure for discrete dynamic open contours in the MRM dataset. The contour initialisation and final discrete dynamic and parametric contours are shown in Figure 6.27 for structures marked with an asterisk.

In our experiments, open contours using the gradient alone and the gradient plus pressure give almost the same results. Similar to our experiment with open contours on the histological dataset in Section 6.5.1, the gradient alone required a larger number of iterations for convergence to the boundary. Using the gradient plus pressure removes the capture-range limitation that exists with the gradient alone; therefore, the initial contour can be located farther from the boundary than the initial contour used for the gradient alone. Figure 6.38 shows examples of the initial contours and the results of segmentation with open contours using the gradient alone and the gradient plus pressure.

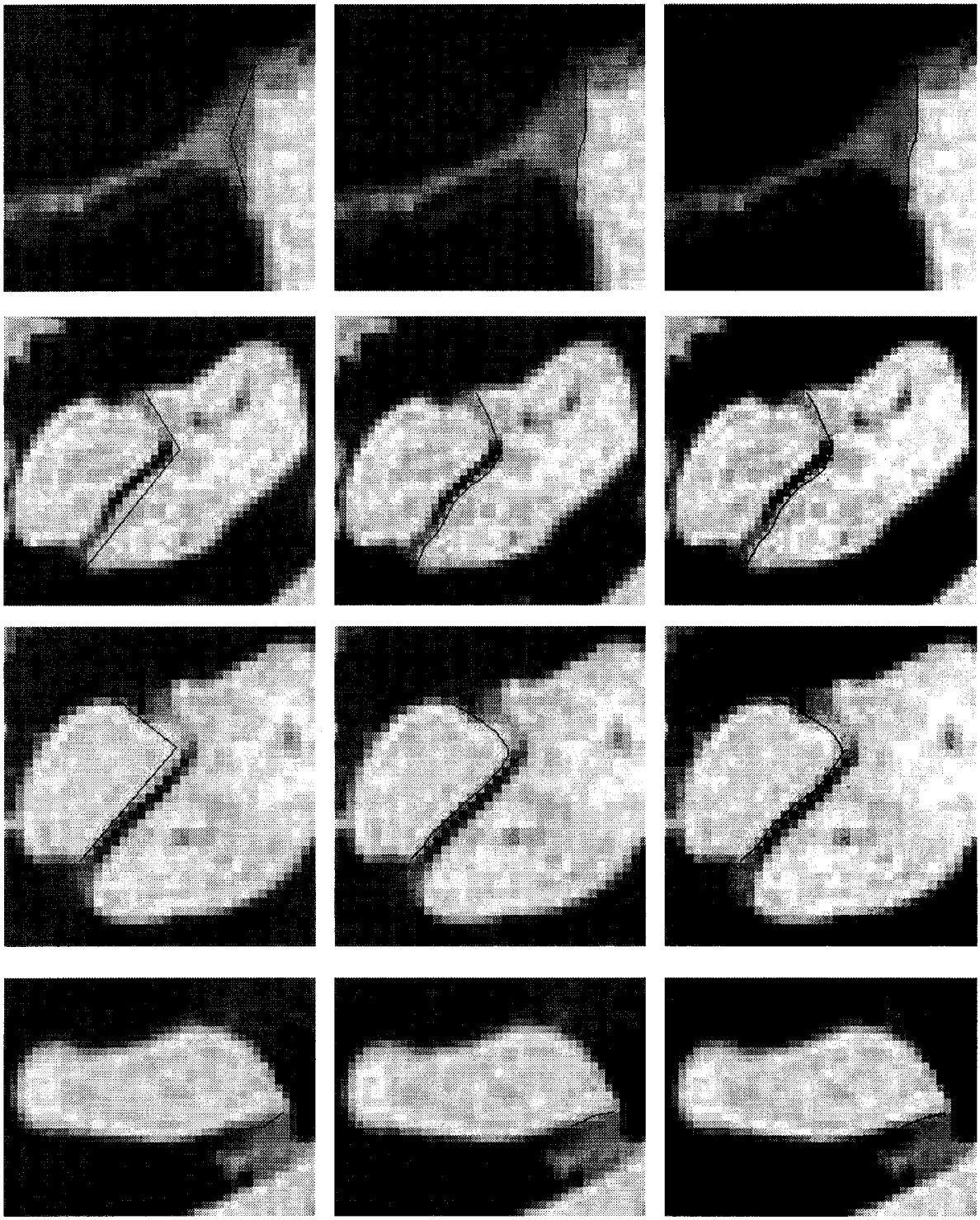


Figure 6.38: Examples of open contours using gradient and gradient plus pressure: (a) the initial contour on the slice 85 (bony tympanic ring), (b) the final contour using the gradient, and (c) the final contour using the gradient plus pressure; (d) the initial contour on the slice 115 (the incus shared surface with the soft tissue located between incus and malleus), (e) the final contour using the gradient and (f) the final contour using the gradient plus pressure; (g) the initial contour on the slice 118 (the malleus shared surface with the soft tissue located between incus and malleus), (h) the final contour using the gradient, and (i) the final contour using the gradient plus pressure; (j) the initial contour on the slice 129 (the incus shared surface with the posterior incudal ligament), (k) the final contour using the gradient, and (l) the final contour using the gradient plus pressure.

6.5.3 Summary

In our experiments with open contours on the histological and MRM datasets, we found that both the gradient alone and the gradient plus pressure give good results for segmentation of the shared surfaces between structures. However, since pressure force improves the capture-range limitation which exists with the gradient method, we found that when using the gradient plus pressure the initial contour can be located farther from the boundary than when using the gradient alone. Also, a smaller number of iterations is required for open contours to converge to the boundary when using the gradient plus pressure.

For segmentation of the thin structures in the histological dataset such as the eardrum, we found that the gradient is not an appropriate technique since it only detects the exterior pixels along the structure's length while the central pixels are required for this purpose. In some cases, since the contrast of the eardrum varies along its length, the gradient cannot even properly lead the active contour to converge to the exterior of the structure and it may cause the open contour to jump from one side to the other side of the structure.

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

Image processing and computer vision have been well-established academic fields since the 1970s. Although many academics have published applications in this field, few have been highly reliable and successful in practice. Academic emphasis tends to be directed more toward pure innovation and mathematical elegance – demonstrated with a few carefully chosen test results – than toward reliability (Guéziec, 2002).

Modern imaging devices provide exceptional views of internal anatomy, but the use of computers to quantify and analyse (*e.g.*, segment) the embedded structures with accuracy and efficiency is limited (McInerney *et al.*, 1996. Segmentation or boundary identification of structures from medical images is difficult due to the complexity and variability of anatomic shapes of interest. Furthermore, the shortcomings typical of sampled data, such as sampling artefacts, spatial aliasing and noise, may cause the boundaries of structures to be indistinct and disconnected (McInerney *et al.*, 1996). Traditional low-level image-processing techniques (*e.g.*, gradient operators), which consider only local information, can make incorrect assumptions and generate undesired object boundaries (*e.g.*, a contour with gaps).

As an alternative, instead of exploiting only image information as low-level techniques do, active contours also use the information about the boundaries as part of an optimisation procedure. Variant approaches and techniques have been proposed to improve the original form of active contours, *i.e.*, traditional parametric contours as originally proposed by Kass *et al.*, (1986). For instance, techniques such as pressure force (Cohen, 1991) and GVF (Xu & Prince, 1997) were proposed to improve the capture range of the external forces, which are generally based on the gradient of the image. Different active-contour formulations, such as discrete dynamic contours (Lobregt & Viergever, 1995), were also introduced to simplify the complexity of computation and improve the performance of the parametric contours.

We used histological and MRM datasets of the middle-ear for our experiments. In our experiments with active contours on middle-ear images, we used and compared traditional parametric and discrete dynamic contours with gradient, with gradient vector flow and with the gradient plus pressure force as the external force. For implementation of traditional active contours, we used a Matlab programme written by Xu & Prince at Image Analysis and

Communications Lab, Johns Hopkins University. We used Oxiana, a semi-automatic computer programme, developed in our lab, that implements the discrete dynamic contours. Closed contours were used for the experiments with both active contour approaches. We also demonstrated open contours for segmentation of shared boundaries between structures.

When using gradient as the external force, for the same initial contours, a larger number of iterations is required for the parametric contours to converge to the boundary than with the discrete dynamic contours. However, when using GVF and the gradient plus pressure, parametric active contours require a smaller number of iterations for active contour convergence, compared with the discrete dynamic approach, at least for the weighting factors used here.

The gradient alone gives good results for all the selected structures, but the initial contour must be located relatively close to the boundary in order to converge. This affirms the capture-range limitation of the gradient technique. The GVF and the gradient plus pressure increase the capture range of the gradient and also accelerate the convergence of the contours to the boundaries. As a result, a smaller number of iterations and less time are required for active contour convergence when using GVF or gradient plus pressure, compared with gradient alone.

When using the same initial contours, active contours require smaller numbers of iterations with GVF than with gradient plus pressure. Although GVF is good for simple images, we found that in real images (*e.g.*, our datasets) it is not a reliable technique. One reason is that GVF may wrongly identify the boundary and lead the active contours to pass the boundary, when applied to regions with low boundary contrast. The other reason is that GVF may wrongly identify the boundary regions between nearby structures even with high boundary contrast.

The MRM datasets have a lower resolution and contrast than the histological datasets. With the histological dataset, the gradient plus pressure succeeds for most regions where the GVF technique fails, especially for structures containing concavities. With the MRM dataset, however, for some structures with low boundary contrast and/or narrow concavities, active contours fail even when using the gradient plus pressure.

Both μ (used for GVF computation) and the pressure weighting (used for the gradient plus pressure) depend on the boundary contrast of the region of interest. Since the boundary contrast is consistent neither within a slice nor across slices, the appropriate μ and pressure

weighting factors must be chosen by trial and error.

In our experiments with open contours in the histological and the MRM datasets, we found that gradient and gradient plus pressure give good results for segmentation of the shared surfaces between structures. However, since pressure force improves the capture range of the gradient, we found that when using the gradient plus pressure the initial contour can be located farther from the boundary than when using the gradient alone. When using the gradient plus pressure, a smaller number of iterations is required for open contours to converge to the boundary.

For segmentation of thin structures such as the eardrum in our histological dataset, the results indicated that the gradient is not a reliable technique, since in some of our selected slices the contrast of the eardrum varies along its length, so the active contour may jump from one side to the other side of the structure.

7.2 Future work

- Digital colour images are composed of different (usually three) layers, each related to specific colour information (*e.g.*, red, blue and green). In grey-scale images that are derived from colour images, the image layers are combined (*e.g.*, averaged) and normalised. The colour histological images that we used for our experiments were converted to grey-scale images. We suggest that it may be profitable to exploit the colour information of the histological images, since the colour information may give extra information for the external (image) force and improve the performance of the active contours.

- As discussed in Section 3.4, the internal force should not attempt to minimise the curvature for areas with constant curvature. In order to satisfy the condition, a filter was suggested (Lobregt & Viergever, 1995) to convolve with the internal force of each vertex. We used the same simple filter as they have suggested in order to get a more robust result. As Lobregt and Viergever (1995) mentioned, the filter can be designed as an adaptive filter. We also propose to find a sophisticated design for the filter in order to get a more robust result.

- We are interested in testing the performance of active contours with other modalities, such as x-ray CT and PET, and especially ultrasound images that contain a larger amount of noise than other types of images.

- In our Oxiana programme for discrete dynamic contours, all user manipulations of the vertices must be performed using the Vertex List menu in the interface. It is suggested to improve the interface to make the use of the programme easier for the user, such as moving a vertex by explicitly using the mouse instead of using the Vertex List. Also, left clicking the mouse on a vertex could open a menu of all the manipulations possible for that vertex.

- As we found in our experiments with open contours for segmentation of thin structures (*e.g.*, eardrum), gradient cannot lead the active contours to rest on the centre of the structure and the contour may leap from one side to the other side of the structure. We propose to find an algorithm to detect the centre pixel, and it can be used to generate the external force for the open contours.

REFERENCES

Amini AA, Weymouth TE & Ramesh CJ (1990): "Using dynamic programming for solving variational problems in vision," IEEE Trans. on PAMI, 12 (9), 855-867.

Atkins MS & Mackiewich BT (1998): "Fully automatic segmentation of the brain in MRI," IEEE Trans. on Medical Imaging, 17 (1), 98-107.

Ayache N (1995): "Medical computer vision, virtual reality and robotics," Image and Vision Computing, 13 (4), 295-313.

Bankman IN (2000): "Handbook of medical imaging: processing and analysis," Academic Press.

Bizais Y, Barillot C & Paola RD (1995): "Information processing in medical imaging," Proceedings of 14th Int. Conf., Kluwer Academic.

Bronzino JD (1995): "The biomedical engineering handbook," CRC Press Inc & IEEE Press.

Brown BH, Smallwood RH, Barber DC, Lawford PV, Hose DR (1999): "Medical physics and biomedical engineering," Institute of Physics publishing, Bristol and Philadelphia.

Canny J (1986): "A computational approach to edge detection," IEEE Trans. on PAMI, 8 (6), 679-698.

Caselles V (1995): "Geometric models for active contours," IEEE proceedings of Int. Conf. on Image Processing, (3), 9-12.

Caselles V, Catte F, Coll T & Dibos F (1993): "A geometric model for active contours," Numerische Mathematic, (66), 1-31.

Caselles V, Kimmel R & Sapiro G (1995): "Geodesic active contours," IEEE Proceedings of the Fifth Int. Conf. on Computer Vision, 694-699.

Cohen LD (1991): "On active contour models and balloons," Computer Vision, Graphics, and Image Processing: Image Understanding, 53 (2), 211-218.

Cohen LD & Cohen I (1993): "Finite-element methods for active contour models and balloons for 2-D and 3-D images," IEEE Trans. on PAMI, 15 (11), 1131-1147.

Curry III TS, Dowdey JE, Murry Jr. RC (1990): "Chirstensen's physics of diagnostic radiology," Lea & Febiger, 432-505.

Davatzikos CA & Prince JL (1995): "An active contour model for mapping the cortex," IEEE Trans. on Medical Imaging, 14 (1), 65-80.

Dettman JW (1962): "Mathematical methods in physics and engineering," McGraw-Hill, NY.

Feng Y & Gelenbe E (1998): "Adaptive object tracking and video compression with dynamic contours," IEEE Proceedings of the Fifteenth National Radio Science Conf., NRSC, 1, INV3/1 -INV326.

Gonzalez RC & Woods RE (1992): "Digital image processing," Addison-Wesley, 2nd edition.

Guéziec A (2002): "Tracking pitches for broadcast television," Computer, 35 (3), 38-43.

Ham WA & Cormack DH (1979): "Histology," J.B. Lippincott Company, 8th edition.

Hamelin LP, Labonté F & Pelletier B (1999): "An active contour Snake algorithm," Internal report, McGill University, Montréal.

Health MD, Sarkar S, Sanocki T & Bowyer KW (1997): "A robust visual method for assessing the relative performance of edge-detection algorithm," IEEE Trans. on PAMI, 19 (12), 1338-1359.

Jain AK (1989): "Fundamentals of digital image processing," Prentice Hall.

Kass M, Witkin A & Terzopoulos D (1986): "Snakes: Active Contour Models," International Journal of Computer Vision, 3, 321-331.

Kreyszig E (1993): "Advanced engineering mathematics," John Wiley, 7th edition.

Leymaire F (1990): "Tracking and describing deformable objects using active contour models," Master's thesis, McGill University, Montréal.

Lindeberg T (1996): "Edge detection and ridge detection with automatic scale selection," Technical report ISRM KTH/NA/P-96/06-SE.

Lobregt S & Viergever MA (1995): "A discrete dynamic contour model," IEEE Trans. on Medical Imaging, 14 (1), 12-24.

Malladi R, Sethian JA & Vemuri BC (1995): "Shape modelling with front propagation: A level set approach," IEEE Trans. on PAMI, (17), 158-175.

Malladi R & Sethian JA (1996): "Level set and fast marching methods in image processing and computer vision," IEEE Int. Conf. on Image Processing, 1, 489-492.

Marr D & Hildreth E (1980): "Theory of edge detection," Proc. R. Soc. London, (207), 187-217.

McInerney T & Terzopoulos D (1996): "Deformable models in medical image analysis," IEEE Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis, 171-180.

Menet S, Saint-Marc P & Medioni G (1990): "Active contour models: overview, implementation and applications," IEEE Int. Conf. proceedings on Systems, Nan & Cybernetics, 194-199.

Menet S, Saint-Marc P & Medioni G (1990): "B-snakes: Implementation and application to stereo," Proceedings of the DARPA Image Understanding Workshop, 720-726.

Miller JV (1990): "On GMD's: "Geometrically deformable models for the extraction of closed shapes from volume data," Master's thesis, Rensselaer Polytechnic Institute, New York.

Miller JV, Breen DE, Wozny MJ (1990): "Extracting geometric models through constraint minimization," IEEE Proceedings of the First Conference on Visualization, 74-82, 464-465.

Osher SJ & Sethian JA (1988): "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jaciob formulation," Journal of Computational Physics. (79), 12-49.

Ranganath S (1995): "Contour extraction from cardiac MRI studies using snake," IEEE Trans. on Medical Imaging, 14, 328-338.

Schalkoff RJ (1989): "Digital image processing and computer vision," John Wiley.

Sethian A (1999): "Level Set Methods and Fast Marching Methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science," Cambridge University Press.

Staib LH, Duncan JS (1992): "Boundary finding with parametrically deformable models," IEEE Trans. on PAMI, 11 (14), 1061-1075.

Stytz M, Frieder G & Frieder O (1991): "Three-dimensional medical imaging: algorithms and computer systems," ACM Computing Surveys, 23 (4), 421-499.

Valdés R, Yáñez-Suárez O & Medina V (2000): "Trachea segmentation in CT images using active contours," The 22nd IEEE Annual EMBS Int. Conf., 3184-3187.

Van Wijhe RG (2000): "A finite-element model of the middle ear of the moustached bat," Master's thesis, McGill University, Montréal.

Wang H & Ghosh BK (1998): "Boundary finding with new balloon models," Proc. 20th IEEE Annual Int. Conf. Engineering in Medicine and Biology Society, 20(2), 686-689.

Webster JG (1992): "Medical instrumentation: application and design," Houghton Mifflin Company.

Williams DJ & Shah M (1990): "A fast algorithm for active contours," The 3rd IEEE Int. Conf. on Computer Vision, 592-595.

Xu C & Prince JL (1997): "Gradient vector flow: a new external force for snakes," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 66-71.

Xu C & Prince JL (1998): "Snakes, Shapes, and gradient vector flow," IEEE Trans. on Image Processing, 359-369.

Xu C, Yezzi Jr. A & Prince JL (2000): "On the relationship between parametric and geometric active contours," IEEE Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, 1, 483-489.

Yezzi Jr. A, Kichenassamy S, Kumar A, Olive P & Tannenbaum A (1997): "A geometric

snake model for segmentation of medical imagery," IEEE Trans. on Medical Imaging, 16 (2), 199-209.

Zhang J, Gao J & Liu W (2001): "Image sequence segmentation using 3-D structure tensor and curve evolution," IEEE Trans. on Circuits and Systems for Video Technology, 11 (5), 629-641.