# Segmentation and 3D Visualization of IntraVascular Ultra-Sound Images

by

Soumya Shivakumar

(Under the direction of Suchendra M. Bhandarkar)

## Abstract

IntraVascular UltraSound (IVUS) image segmentation is still an open problem as no appropriate solution has been discovered yet. The segmentation of IVUS images involves extracting the vessel (outer) and lumen (inner) boundaries from the cross-sectional images. This project applies two widely used image processing techniques separately for the vessel and lumen boundary detection, gradient extraction and adaptive $k$-means clustering, respectively and then employs a well-known contour representation, parametric deformable model to extract both the contours. The contours thus extracted are reconstructed into a 3D data model to assist in visualizing the coronary arteries as a tube-like structure. This helps physicians to visualize and measure the deposition of plaque for appropriate calcium score diagnosis.

Index words: Snakes, Gradient Vector Field, Active Contours, 3D Reconstruction and Volumetric Visualization of IVUS Images, Parametric Deformable Models, $k$-means Clustering, Adaptive $k$-means Clustering

# Segmentation and 3D Visualization of IntraVascular Ultra-Sound Images

by

Soumya Shivakumar

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Master of Science

Athens, Georgia

2011

Segmentation and 3D Visualization
of IntraVascular Ultra-Sound Images

by

Soumya Shivakumar

Approved:

| | |
|---|---|
| Major Professor: | Suchendra M. Bhandarkar |
| Committee: | Tianming Liu<br>Walter D. Potter |

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2011

# Acknowledgements

All original IVUS videos and images are courtesy of Piedmont Heart Institute, Atlanta.

I gratefully acknowledge my advisor, Dr. Suchendra M. Bhandarkar, whose expertise in this field helped me improve my knowledge and writing abilities. This thesis would not have been possible without his inspiration. I extend a special thanks to my committee members, Dr. Tianming Liu and Dr. Walter D. Potter who helped me work through this thesis. I rediscovered my interest in medical imaging through Dr. Liu, whose work in brain imaging and bio-imaging continues to inspire me till date. I especially attribute my writing skills to the constant constructive criticism received from Dr. Potter. I am eternally grateful to him for the same.

I would also like to thank my husband, without whose love and support, I would not be here today. Finally I would like to thank my parents who have carved me into what I am and have always remained a source of love and inspiration for me.

# Contents

# List of Figures

# Chapter 1

# Introduction

Coronary Artery Disease (CAD) is currently the leading cause of death in the United States. Effective means of treating CAD are available, but the main problem lies with the detection and identification of the disease. One of the risk factors associated with the disease is the cholesterol or calcium deposition in the coronary arteries. These deposits, also known as plaque, block the flow of blood through the arteries, therefore increasing the risk of potential cardiac arrest or stroke. Hence early detection of arterial plaque and determination of the degree of plaque deposition is very useful to determine the appropriate treatment. This project aims to provide a visualization of the interior coronary arterial walls to enable a view of the inner lumen and obtain the volume of plaque deposition.

## 1.1 IVUS Imaging

IntraVascular UltraSound (IVUS) is a very recent *'in vivo'* medical imaging technology. 'In vivo' means that it is performed on a living organism. A special catheter with an ultrasound probe at one end is inserted into the blood vessels. The other end of the catheter is connected to an ultrasound detector. The catheter is then pulled out slowly and images of the inner

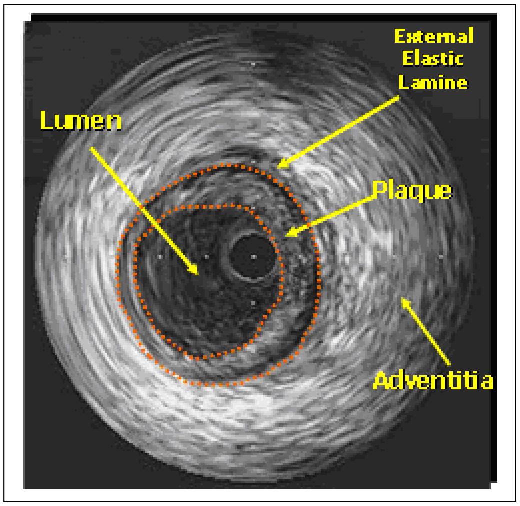

Figure 1.1: Regions in an IVUS image
Source: http://www.incor.usp.br/spdweb/projetos_eng/ivus.htm

wall of the artery are taken at regular intervals, generally at every diastole. IVUS imaging thus enables us to visualize the inner wall of the blood vessels.

IVUS imaging is carried out in the coronary vessels or arteries in order to be able to detect the build up of plaque, which can result in stenosis (narrowing) of the vessels. Narrowing of the arteries can in turn lead to a stroke or cardiac arrest, since blood cannot flow freely through these vessels. Since the progressive accumulation of plaque can be potentially lethal, it is advisable to detect the plaque deposits as early as possible. IVUS images can be studied to determine the extent or degree of stenosis in the arteries and appropriate measures can be taken to prevent further deposits or to reduce the current accumulation.

A sample IVUS image is shown in Figure 1. IVUS images are comprised of three main regions - lumen, plaque and adventitia. The lumen and vessel boundaries separate these three regions. The outer vessel boundary separates the adventitia and the plaque regions.

Plaque forms inside the vessel of the arteries and the inner contour of the plaque deposits defines the lumen boundary. The inner lumen boundary separates the plaque and the lumen regions. The lumen region is the area which is left free for the blood to flow through the coronary arteries.

## 1.2 Segmentation

Image segmentation is a process of clustering pixels into prominent image regions. In other words, image segmentation aims at finding individual objects in an image. This helps in distinguishing image objects from its background. Image segmentation could be used for a variety of applications including object recognition, occlusion boundary estimation, image compression, image editing, and image database look-up.

The basic principle of image segmentation lies in partitioning the image into clusters of ”similar” pixels. This can lead to creating a histogram of the pixel values, on which a certain threshold can be used to differentiate objects in the image from the background. Figure 1.2 gives an simple example of image segmentation on a gray-scale image. As seen in the figure, the objects are clearly distinguished from the background as well as from one another.

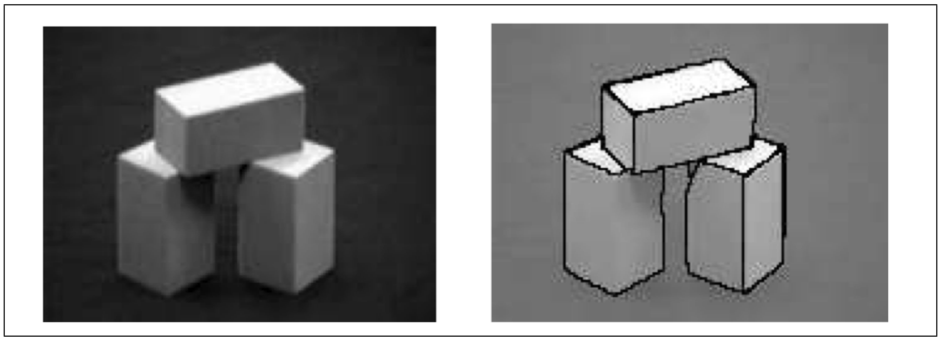

Figure 1.2: Example of Image Segmentation

Source: http://www.cs.toronto.edu/ jepson/csc2503/segmentation.pdf

## 1.3 Active Contours

Active contours, or snakes, are used to detect object boundaries in an image and to describe their shape. These active contours are curves that change shape based on the internal forces of the object and the external forces of the image. These forces are defined in such a way that the snake moves towards the object boundary due to the effect of these forces and eventually conforms to the boundary of that object. Snakes are widely used in many applications including edge detection, shape modeling, segmentation, and motion tracking. There are two types of active contours: parametric active contours and geometric active contours.

### 1.3.1 Parametric Active Contours

Parametric active contours [7], generate curves within an image that are represented using a parametric form. These curves move towards the edges in the image due to the effect of forces in the image. Image forces are of two types: internal forces and external forces. External forces are comprised of potential forces and pressure forces, which push the curve towards a feature in the image. Potential forces can be represented as the negative gradient of a potential function.

Internal forces comprise of elasticity forces, which are designed to keep a curve from breaking under pressure and bending forces, which keep a curve rigid and prevent it from bending too much. Internal forces act within a curve itself and ensure to keep it smooth during deformation. External forces are obtained from the image data and are designed to move the curves towards the edges in the image. The work in this thesis is based on the implementation of the parametric active contour model. A standard parametric snake model is described in detail in Section 1.4.

Internal forces are responsible for imposing a smoothness constraint on the snake contour, whereas external forces are responsible for getting the snake contour close to the desired features of the image that the snake contour is supposed to conform to, typically edges. Internal forces make the snake contour to act like an elastic membrane which changes shape in a particular direction but remains smooth and does not break under the pressure of the pulling forces. The pulling forces are the external forces present in the images which act upon the snake contour and enable it to move in a particular direction, where the energy is minimized.

### 1.3.2 Geometric Active Contours

Geometric active contours [8], are based on the theory of curve evolution and geometric flow. Geometric active contours represent curves and surfaces implicitly as a level set of an evolving scalar function. Since curves evolve only through geometric measures, the evolution is not dependent on the parametrization of the curve and this in turn rids the dependence on changes in topology. Geometric active contours share certain similarities with a sub-family of parametric models [18], but these similarities have been established only for active contours derived from the energy minimization framework and to contours using elastic forces. Other classes of parametric contour models do not have equivalent geometric contour models.

### 1.3.3 Parametric Vs. Geometric Active Contours

Although parametric and geometric active contours have equivalences, they are quite different in many aspects. Some of the differences between parametric and geometric active contours are listed below.

- Parametric active contours have an implicit representation of the energy model, whereas geometric active contours have an explicit representation.

- Parametric active contours are iteratively evolved and are parameterized after every iteration. This facilitates a direct interaction between the user and the contour itself. However, since geometric active contours are parameterized only after the contour evolution, direct interaction is not possible.

- Since parametric active contours are dependent on both the internal and external forces in the energy framework, they are easily controllable in terms of rigidity and elasticity. Geometric active contours however, do not work well in the existence of weak image edges [10], because geometric active contour models require the edges in the image to be continuous. If the edges are weak or broken, geometric active contour models may result in contour leaking.

- Since geometric active contours can handle topological changes effectively, it is only intuitive that they are better suited in cases where topology plays an important role, such as cases where the contour evolution can exploit the object topology for better optimization.

- It is a well known fact that parametric active contours are dependent on the initialization of the snake contour. However, geometric active contours are initialization independent, because they are topologically independent. Hence, geometric active contours can be initialized even on the image border and that would not affect the final result. However, they are more sensitive to noisy artifacts in the images when compared to parametric contours.

- Parametric active contours are computationally less complex than geometric active contours. This is due to the fact that geometric active contours perform global deformation, which can be more time consuming and computationally more complex and hence, in turn, less efficient than their parametric counterparts.

On account of the reasons stated above which show that the parametric active contours are easily controllable, facilitate direct interaction with the user, are less sensitive to noise in the images, are more efficient in terms of time complexity and are computationally less complex than geometric active contours, the work in this thesis is biased towards using parametric active contours.

## 1.4 Standard Parametric Model

A traditional snake contour or active contour is defined by $\mathbf{x}(s) = [x(s), y(s)], s \in [0, 1]$, that moves through an image towards a feature where the energy functional described below is minimal.

$$E = \int_0^1 \tfrac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] + E_{ext}(\mathbf{x}(s))ds \tag{1}$$

where $\alpha$ and $\beta$ are weights related to the tension and rigidity of the snake contour respectively and $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ represent the first and second derivatives of $\mathbf{x}(s)$ with respect to $s$. $E_{ext}$ is the external energy force derived from the image, such that it attains a minimum at the image features, mainly edges. Possible external energy functions for a gray-scale image $I(x, y)$ are given by equations (2) or (3), whereas possible external energies for a binary image (black object on white background) are given by equations (4) or (5):

$$E_{ext}(x, y) = -| \bigtriangledown I(x, y)|^2 \tag{2}$$

$$E_{ext}(x, y) = -| \bigtriangledown [G_\sigma(x, y) * I(x, y)]|^2 \tag{3}$$

$$E_{ext}(x, y) = I(x, y) \tag{4}$$

$$E_{ext}(x, y) = G_\sigma(x, y) * I(x, y) \tag{5}$$

where $G_\sigma(x, y)$ is a two-dimensional Gaussian function with a standard deviation $\sigma$, and $\bigtriangledown$ is the gradient operator.

A snake contour that minimizes the energy functional should also satisfy the Euler equation.

$$\alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \bigtriangledown E_{ext} = 0 \tag{6}$$

This can be converted into a force balance equation as follows:

$$F_{int} + F_{ext} = 0 \tag{7}$$

where $F_{int} = \alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s)$ and $F_{ext} = -\bigtriangledown E_{ext}$. The internal force $F_{int}$ prevents the contour from bending while the external force $F_{ext}$ pulls or pushes the snake contour towards image features.

The variable $\mathbf{x}$ can be represented as a function of time, $t$ as well as $s$, $\mathbf{x}(s, t)$. This makes the snake contour dynamic and helps us to obtain a solution for equation (6). The partial derivative of $\mathbf{x}$ with respect to $t$ is set to the left hand side of equation (6) as shown in equation (8). Thus, when $\mathbf{x}(s, t)$ stabilizes, equation (6) is solved because $\mathbf{x}_t(s, t)$ no longer exists.

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \bigtriangledown E_{ext} \tag{8}$$

A numerical solution to equation (8) can be obtained by solving a discretized equation iteratively [7].

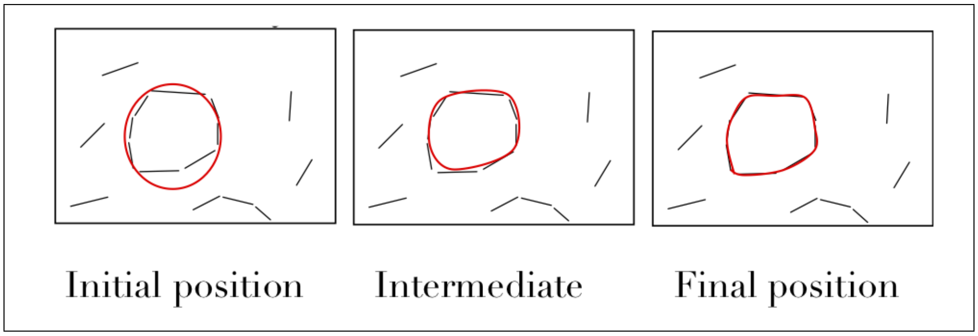

Figure 1.3: Movement of a snake towards the object boundary

## 1.5 Behavior of Traditional Snake Contours

Snakes basically try to match a specified deformable contour by solving an energy minimization equation. The forces defined in the energy minimization equation are computed from image features and properties and are such that they push or pull the snake towards features in the image. If a snake contour finds a low-energy feature in its neighborhood or vicinity, it moves towards that feature using lead gradient descent search. The snake contour therefore relies on appropriate initialization. Figure 1.3 shows the deformation of a snake contour towards the nearest boundary in the image.

The vicinity or neighborhood of the boundaries is defined as the capture range of the external boundary force. Beyond this neighborhood area, the image regions are homogenous, i.e., the gradient in the image is zero. Figure 1.4 shows the capture range generated by potential forces and the gradient vector field. The arrows seen in the figure represent the forces in the image. It can be seen in the images that the forces are directed towards the boundary of the object. The snake contour gets pulled in the direction of the forces towards the edge of interest.

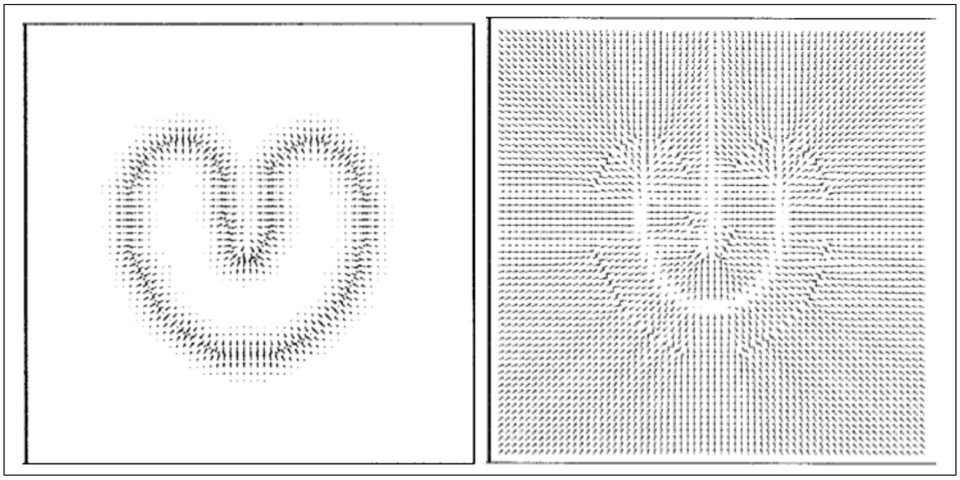

Figure 1.4: Capture range created by potential forces and the gradient vector field
Source: http://www.iacl.ece.jhu.edu/ chenyang/research/pubs/p084j.pdf

For standard parametric snake contours, the initial contour needs to be defined near the edge of interest if it has to deform appropriately or get pulled towards the appropriate boundary, because the capture range of the forces is very small. The initial contour needs to be initialized within the capture range of the forces, if the forces have to act on the contour. This limits the application or program to rely on user input to define initial snake contours. The user also needs to have prior knowledge of the image to be able to appropriately initialize the contour. Other limitations of the standard parametric model are listed in Section 3.1.

Figure 1.5 shows a snake contour being pulled away from the edge of the object. When the force pulling the contour is lifted i.e., the contour is released, the snake gradually moves back towards the edge of interest. The force governing this movement is the internal energy of the contour. The snake moves toward the minimized energy feature on the image to which it is already attached i.e., if a part of the snake is already attached to a desired image feature,

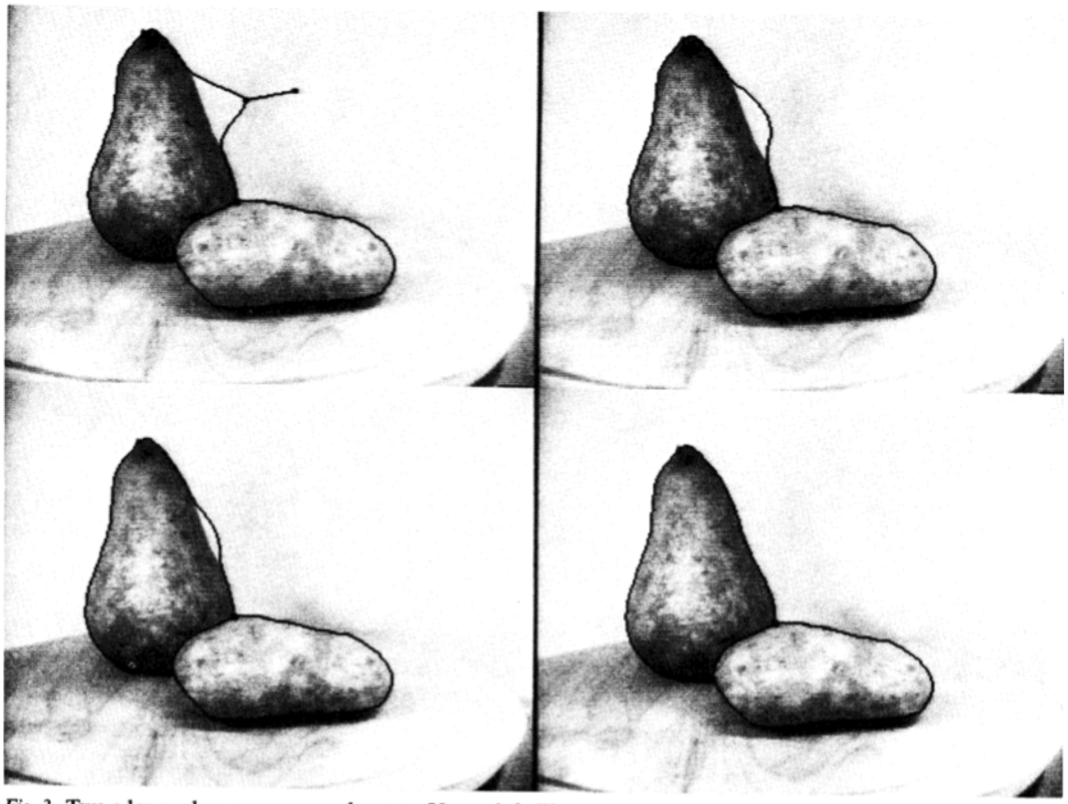

*Fig. 3.* Two edge snakes on a pear and potato. Upper-left: The user has pulled one of the snakes away from the edge of the pear. Others: After the user lets go, the snake snaps back to the edge of the pear.

Figure 1.5: Example of a snake contour being pulled towards the boundary of an object
Source: http://www.cs.ucla.edu/ dt/papers/ijcv88/ijcv88.pdf

the rest of the snake also gets conformed to the same feature. So, it is actually enough even if a part of the snake is inside the capture range of the external force governing the movement of the snake.

## 1.6 The $k$-means Clustering Algorithm

The $k$-means Clustering algorithm is a least-square partitioning method which partitions an image into $k$ clusters such that the pixels in the image belong to the cluster with the nearest mean. The basic concept of the clustering technique is to identify $k$ initial pixels in the image, which represent the initial cluster means. The remaining pixels in the image are then assigned to these means to create $k$ initial clusters, such that each pixel is assigned to the mean which is closest to it. The means are recalculated and the image pixels are reclustered with respect to the new means. This process is continued until the means do not change.

The step-by-step algorithm for $k$-means clustering is as below:

1. Place $k$ points at random positions in the 2D space represented by the image whose pixels are being clustered. These points represent the initial cluster means.

2. Assign each pixel in the image to the cluster with the closest cluster mean.

3. When all pixels have been assigned, recalculate the positions of the $k$ mean points.

4. Repeat steps 2 and 3 until the mean points no longer change. This produces a separation of the image into $k$ objects.

## 1.7 Adaptive $k$-means Clustering Algorithm

The $k$-means clustering algorithm requires a user to specify the number of clusters into which the image pixels can be clustered. This leads to the fact that the resultant clusters are dependent on the initialization. An adaptive $k$-means clustering technique eliminates this requirement. The simplest form of adaptive $k$-means algorithm adapts to the image and a different number of clusters are obtained for different images based on the image properties. The adaptive $k$-means algorithm is as follows:

1. Consider $k = 1$.
2. Perform basic $k$-means clustering, as described in Section 1.6.
3. Check if resultant clustering is satisfactory based on a certain preset condition.
4. If the result is satisfactory, go to Step 6.
5. If the result is not satisfactory, increment the value of $k$ by 1 and go to Step 2.
6. Save the segmented image. Stop.

The above algorithm generates a segmented image with $k$ clusters, the value of $k$ being different for every image. The preset condition for Step 3 in the case of adaptive $k$-means clustering technique is generally user-defined. For this project, the satisfaction criteria is that the correlation coefficient between the original image and the segmented image has to be greater than 0.99. The correlation coefficient $r$, used for this project is as given below:

$$r = \frac{\sum_m \sum_n (OI_{mn} - \overline{OI})(SI_{mn} - \overline{SI})}{\sqrt{\left(\sum_m \sum_n (OI_{mn} - \overline{OI})^2\right)\left(\sum_m \sum_n (SI_{mn} - \overline{SI})^2\right)}}$$

where $OI$ is the original image matrix, $SI$ is the segmented image matrix, $\overline{OI}$ is the mean of the matrix $OI$ and $\overline{SI}$ is the mean of the matrix $SI$

## 1.8 Gradient Vector Flow field

Gradient Vector Flow (GVF) field is an external potential force field obtained from the gradient of an image. This is a dense vector fields obtained by the diffusion of the gradient vectors of an edge map of the image. GVF Snakes are the active contours which use GVF as the external force in the force balance equation as follows:

$$\mathbf{x}_t(s,t) = \alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) + \mathbf{v} \tag{9}$$

where $- \bigtriangledown E_{ext}$ in (8) is replaced by $\mathbf{v}$, which is the GVF field. The snake contour that satisfies and solves this equation is called a GVF snake.

Initially, an edge map $f(x,y)$ is created from the image $I(x,y)$. An edge map is characterized by a large magnitude near the edges. Any of the external energy forces defined in equations (2), (3), (4) or (5) can be used to generate an edge map, depending on the type of image that is being worked on. The GVF field is then defined as $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$ and the energy framework is defined as follows:

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + | \bigtriangledown f|^2 |\mathbf{v} - \bigtriangledown f|^2 \, dx \, dy \tag{10}$$

From the above equation, we can see that if $|\bigtriangledown f|$ is small, then the sums of the squares of the partial derivatives of the GVF field dominate the second term in the energy framework, thus creating a slowly changing gradient field. On the other hand when $| \bigtriangledown f|$ is large, the second term is dominant in the energy framework. This can be minimized by setting $\mathbf{v} = \bigtriangledown f$. This ensures that the gradient is equal to the edge map when it has a large magnitude, whereas it leads to a gradually diffusing force field in homogenous regions.

Here, $\mu$ is a user-defined parameter used to regulate the tradeoff between the two terms in the energy functional. It is initialized to a value directly proportional to the amount of noise in the image, i.e., it is initialized to a high value if the amount of noise in the image is high and it is initialized to a low value if the amount of noise in the image is less. The *calculus of variations* [19], shows that the GVF field can be obtained by solving the following Euler equations (11) and (12).

$$\mu \bigtriangledown^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \tag{11}$$

$$\mu \bigtriangledown^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \tag{12}$$

where $\bigtriangledown^2$ is the Laplacian operator, $f_x$ and $f_y$ are the gradients in the $x$ and $y$ directions respectively. Within the homogenous regions, $u$ and $v$ are determined only by Laplace's equation since the gradient within these regions is zero. The GVF field in such regions is diffused from the edge gradient, with larger magnitude vectors along the edge and the value of the GVF field gradually reducing as it moves further away from the edges. This is the reason why the GVF field has vectors pointing into boundary concavities. This feature of the GVF field is very important in relation to this project because it facilitates the active contour to conform with the plaque or vessel boundary with high accuracy.

As seen earlier in the standard parametric model, $u$ and $v$ in the equations (11) and (12) can be solved by representing them as functions of time $t$ as shown in equations (13) and (14).

$$\begin{aligned} u_t(x,y,t) &= \mu \bigtriangledown^2 u(x,y,t) - [u(x,y,t) - f_x(x,y)] \\ &\quad .[f_x(x,y)^2 + f_y(x,y)^2] \end{aligned} \tag{13}$$

$$\begin{aligned} v_t(x,y,t) &= \mu \bigtriangledown^2 v(x,y,t) - [v(x,y,t) - f_y(x,y)] \\ &\quad .[f_x(x,y)^2 + f_y(x,y)^2] \end{aligned} \tag{14}$$

When equations (13) and (14) stabilize, a solution to the equations (11) and (12) is reached.

## 1.9 Project Overview

The work in this thesis focuses on the implementation of extensively used image processing techniques such as gradient extraction and adaptive $k$-means clustering for the segmentation of IVUS images to detect the inner lumen and outer vessel boundaries. The IVUS process generates a video of the inner wall of the arteries of the patient. This video is processed to obtain image frames. Each of the images is subjected to gradient computation and adaptive $k$-means clustering.

The gradient computation generates an edge map of the image, which helps in the detection of the outer vessel boundary. Adaptive $k$-means clustering separates the inner lumen from the other regions in the image. The clustering results in several boundary contours - separating the image regions - the innermost of which is typically the inner lumen boundary.

An active snake contour is initialized in each of the resulting images from edge map computation and adaptive $k$-means clustering. The contour is then deformed using the GVF snake model equations, which results in the generation or extraction of both the inner lumen and the outer vessel boundaries. These boundary contours are then stacked at regular intervals and interpolating contours are generated using a linear interpolation technique between the contours. An isosurface of the resulting 3D structure is extracted for volumetric visualization of the coronary arteries.

# Chapter 2

# Literature Review

IVUS image segmentation is a crucial step in the diagnosis of coronary diseases through IVUS images. Typically, manual segmentation is applied to trace the vessel and lumen boundaries in the IVUS images. Manual segmentation however is very error prone and is also known to be very time-consuming. Hence, automated image processing based approaches have been employed over the years to make this process easier, faster and also more accurate.

Image processing approaches over the years have tried to accelerate the process of IVUS image segmentation by exploiting features and properties from the images to overcome the problems posed by noisy artifacts, shadows caused by plaque deposits and dark patches caused by the existence of branches in the arteries. Most of the previous works in IVUS image segmentation are based on exploiting pixel information. These include, but are not limited to multi-agent image segmentation [25], region growing techniques [26], Sobel-like edge detection [27], and graph searching algorithms [28].

Multi-agent image segmentation employs a modular approach by generating a separate agent for each image object and trying to classify the objects from the image of interest into different agent categories based on a pre-defined set of rules. However, like any typical classification system, this type of segmentation is subject to conflicts in classification and may require conflict-resolution methods, which can add to the computational complexity of the rule-based system, which in itself is a very complex system.

Edge detection algorithms are known to fail in cases where the boundary has very low contrast as compared to the background, especially for the luminal boundary in the IVUS images. Graph-searching algorithms have the traditional problem of getting stuck in local minima and not reaching the optimal solution without the required expert guidance or human intervention. Hence, the idea of snakes or active contours was developed to automate the process and to overcome the above mentioned problems.

Snakes, or active contours, were first defined by Kass, Witkin and Terzopoulos [7] as splines or curves which are affected by external forces within an image domain and can move under the influence of these forces towards salient features in the images such as edges. They developed an energy minimization framework to represent the movement of these contours. Snakes are widely used in many applications, including edge detection, shape modeling, segmentation, and motion tracking.

Snake models are broadly classified into two types: parametric active contours [1, 2, 3, 5, 7] and geometric active contours [8, 9, 17, 18]. Different force fields used in the energy minimization equation lead to the formulation of various parametric snake models. However, the snakes defined with conventional force fields have a problem that the initialization of the contour requires human intervention. Since the force fields have a limited capture range, the active contour gets pulled towards undesired features in case of improper initialization.

Pressure or balloon forces [21] were defined to address the problem of vicinity or capture range, but these forces worked only in one direction. They could either inflate or deflate a contour towards the edges, i.e., the contour could not deform in all directions. The pressure or balloon forces need to be initialized either inside or outside the boundary, which again requires special human intervention in order to properly initialize the contour to enable it to travel in the desired direction.

Xu and Prince [1, 2] discovered a new force field, which they called the Gradient Vector Flow (GVF) field. Even though the gradient of an image is strong only near the edges and nearly zero in homogenous regions, they discovered a way to diffuse this gradient and extend the force field into the homogenous regions in the image in order to increase its capture range. GVF snakes are one of the most widely used deformable models in image segmentation currently. This model also solved most of the problems existing with previous snake models without giving rise to new problems as described in the following section.

# Chapter 3

# Motivations for Current Work

## 3.1 Limitations of Active Contours

Traditional Parametric Active Contours have several limitations as described below.

1. Initialization of the snake contour should be close to the true object boundary, otherwise, it may converge to an incorrect result, because the force fields have a very small capture range as seen in Figure 3.1 (b). Pressure forces and distance potentials have been proposed to address this problem.

2. Active contours have difficulties progressing into object concavities [2] as seen in Figure 3.1 (a). This is due to the fact that the force vectors in the image do not point into the concavity, but only point horizontally in opposite directions as seen in Figure 3.1 (c). Hence, the contour gets pulled in opposite directions near the mouth of the concavity, but does not move into it.

Related work in this area has solved the problem, but at the cost of creating new problems. Pressure forces or balloon forces [21], for example can deform an active contour into object concavities, but it has been shown that strong pressure forces can override weak object edges

[20]. Careful selection is hence necessary so that the pressure force is slightly smaller than the Gaussian potential forces at the desired edges, but larger than the Gaussian potential forces at spurious edges, since the model is inflated or deflated until it reaches the Gaussian potential forces at the desired edges.

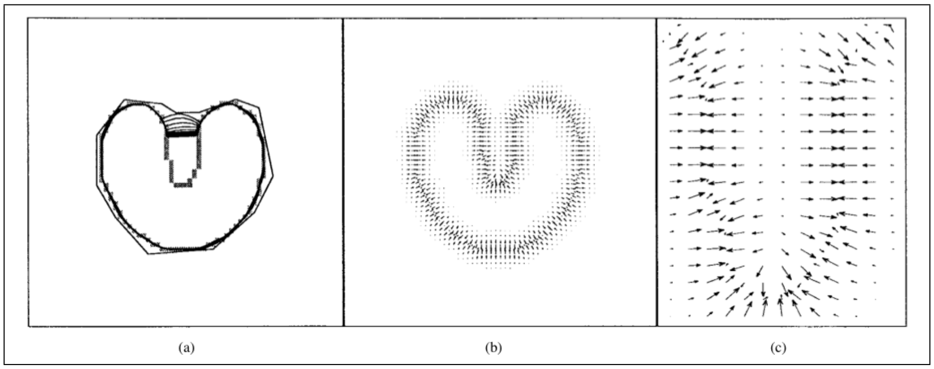

Figure 3.1: (a) Convergence of a snake using (b) potential forces, and (c) shown close-up within the boundary concavity.

Source: http://www.iacl.ece.jhu.edu/ chenyang/research/pubs/p084j.pdf

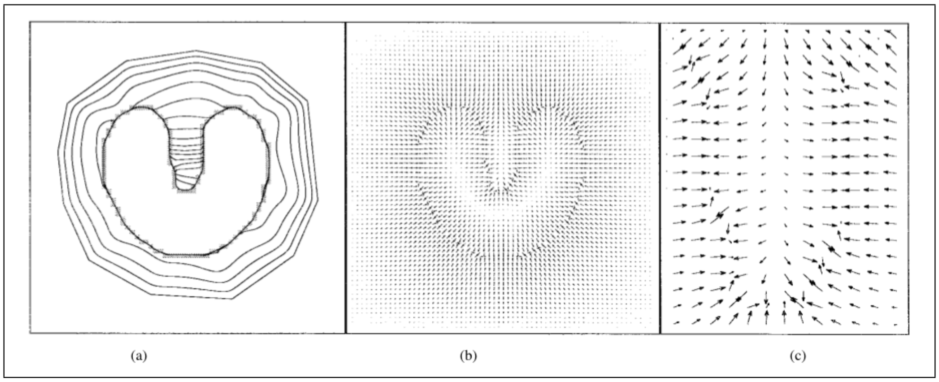

Figure 3.2: (a) Convergence of a snake using (b) GVF external forces, and (c) shown close-up within the boundary concavity.

Source: http://www.iacl.ece.jhu.edu/ chenyang/research/pubs/p084j.pdf

## 3.2 Advantages of GVF Snakes

GVF Snakes have three major advantages over the traditional active contours.

Firstly, GVF has a large capture range as seen in Figure 3.2 (b) and hence, the initial GVF snake contours can be initialized far away from the object boundary, without affecting the final convergence. Secondly, GVF Snakes can progress into boundary concavities [2], as seen in Figure 3.2 (a) without creating new difficulties, which is a major plus point. This is due to the fact that the GVF vectors at the mouth of the concavity point into the concavity and since they also have a large capture range, the snake contour easily gets pulled into the concavity.

Lastly, GVF Snakes can be initialized across boundaries too [1, 2], unlike pressure forces which need to be carefully initialized on one side of the desired edge and not across the boundaries. This is because the energy functional used with pressure forces can either inflate or deflate the model, depending on whether the snake contour is initialized inside or outside the desired object by the user [21] and hence can deform only in one direction.

## 3.3 Limitations of the $k$-means Clustering Algorithm

The $k$-means clustering technique is subject to several limitations as described below:

1. The $k$-means clustering algorithm is slow and scales poorly on time.

2. The $k$-means clustering algorithm requires that the user specify the number of clusters for the value of $k$ beforehand, which is unreasonable in cases where the image objects are not clearly distinguishable from the background. Also, the value of $k$ does not remain the same for every image, which would require human intervention to specify

a different $k$ value every time.

3. The $k$-means clustering algorithm can get stuck in local optima.

4. Performance of the $k$-means clustering algorithm largely depends on the initialization of the cluster means.

## 3.4 Advantages of Adaptive $k$-means Clustering Algorithm

Adaptive $k$-means clustering technique overcomes some of the limitations of the $k$-means clustering technique as listed below:

1. Adaptive $k$-means clustering does not require that the user initialize the number of means. Adaptive $k$-means clustering adapts to the image being processed and results in a different value of $k$ for every image.

2. The performance of an adaptive $k$-means clustering process is not dependent on the initial cluster means.

# Chapter 4

# Implementation of IVUS Image Segmentation and Visualization

## 4.1 Hardware and Software Specifications

The software for the IVUS Image Segmentation and Visualization system, except for the isosurface generation was implemented on a Macintosh computer with OS X version 10.6.5, a 2.4 GHz Intel Core 2 Duo processor, and a 2GB RAM. The isosurface generation was performed on a Linux cluster (rcluster.rcc.uga.edu), comprising 331 compute nodes where each node is a dual-core AMD Opteron(tm) Processor clocked at 2.19 GHz. The programs for running the experiments were written and executed in MATLAB, version 7.8.0.347 (R2009a) 32-bit software. ParaView, version 3.6.1 [31], was used for the volumetric visualization of the 3D reconstruction of the inner walls of the coronary arteries.

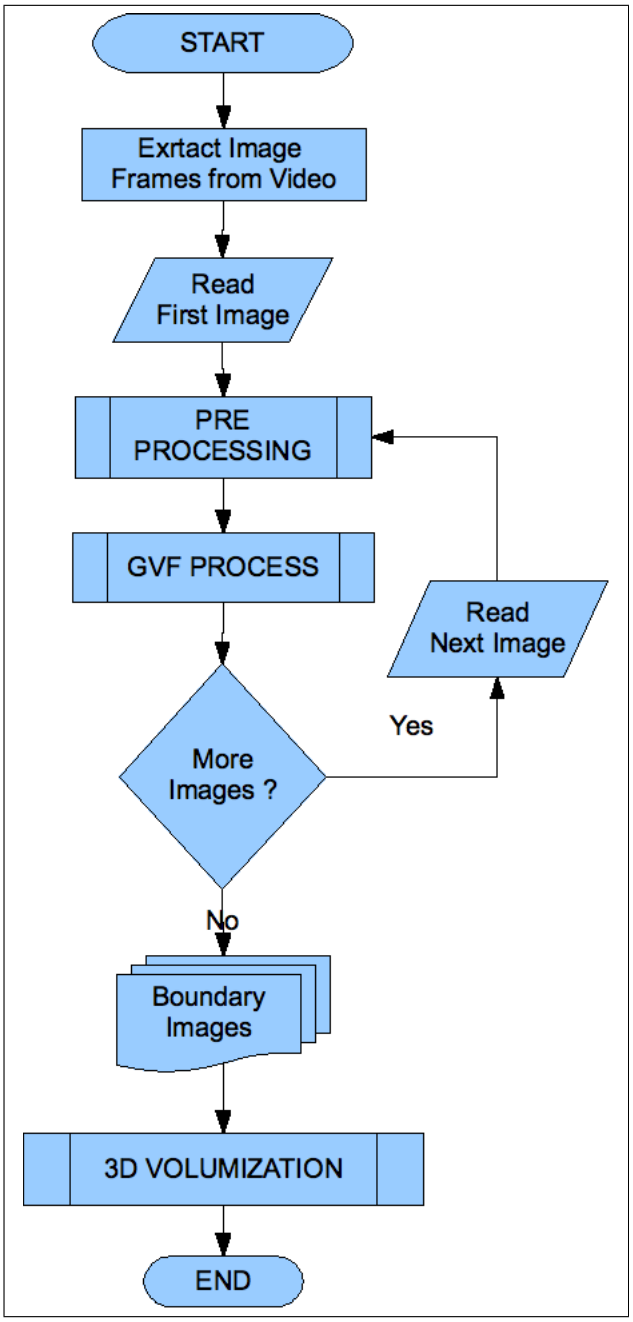

Figure 4.1: Project Implementation Overview

## 4.2 Implementation Overview

The implementation of the IVUS Image Segmentation and Visualization system mainly comprises three phases as shown by the implementation overview flowchart in Figure 4.1:

- Pre-Processing (Detection of the Outer Vessel Boundary and Detection of the Inner Lumen Boundary)

- Gradient Vector Flow Field and Snake Deformation

- 3D Reconstruction and Volume Visualization of the blood vessels

The patient information is obtained in the form of a video comprising several image frames where the images are obtained at regular intervals. The first step in the project is to extract the IVUS images from the video using any video-to-images converter. This process generates several continuous IVUS images. Each of these images is subjected to pre-processing techniques and the GVF process. The resultant processed images are then combined into a single 3D structure and are viewed as a 3D volume using a visualization tool, Paraview, to view the final result.

Initially each of the IVUS images are subjected to a set of pre-processing techniques, which generate appropriate segmented images or edge maps for the detection of lumen (inner) and vessel (outer) boundaries respectively. The GVF process is then applied separately on these two resultant images to obtain the lumen and vessel boundary contours. The boundary contours are then reconstructed to generate a 3D model. Interpolating contours are generated between each pair of such contours, to obtain a smooth 3D model or a tube like structure which can graphically represent a coronary artery for volumetric visualization of the plaque deposits.

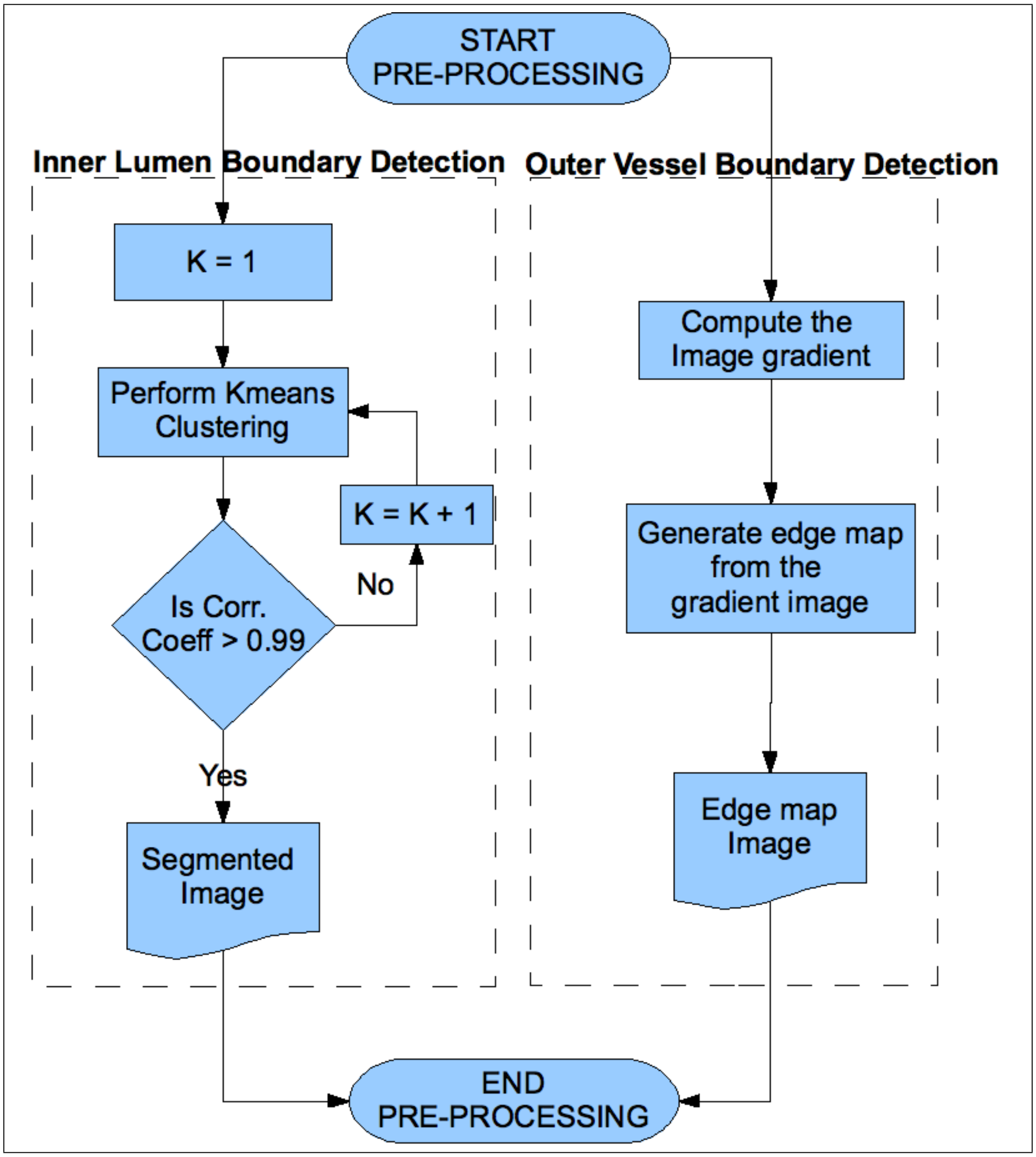
START
PRE-PROCESSING
Inner Lumen Boundary Detection
Outer Vessel Boundary Detection
K = 1
Perform Kmeans Clustering
K = K + 1
No
Is Corr. Coeff > 0.99
Yes
Segmented Image
Compute the Image gradient
Generate edge map from the gradient image
Edge map Image
END
PRE-PROCESSING

Figure 4.2: Pre-Processing Techniques

## 4.3 Pre-processing

In order to detect boundaries in the IVUS images, there is a need to pre-process them to enhance the edge information. However the lumen and vessel boundaries are not easily detected by the same technique, because of the difference in the intensity of the gradient as compared with the image background. The outer vessel boundary is more easily detectable as compared to the inner lumen boundary for reasons explained below. Hence it is necessary to apply two different image processing techniques separately to detect the two boundaries in the image.

Computation of the image intensity gradient easily enhances the vessel boundary in the IVUS images, since it has a high contrast. However, since the lumen boundary has a very low contrast relative to its background, it is necessary to use a more sophisticated technique to segment the lumen region from the rest of the image. Hence an adaptive $k$-means clustering method is employed to extract the lumen region.

The adaptive $k$-means clustering technique cannot however be applied for the detection of the vessel boundary, mainly because there are several other regions detected in the image, inner to the vessel region and ending with the innermost region being the lumen region. These region boundaries along with the lumen boundary generate their respective gradient vector fields which will cause the snake contour to move or deform towards the undesired edges. This is not a desired effect and since the vessel boundary can be detected already by a simpler technique, it is not necessary to apply a complicated procedure to detect it.

A flowchart showing the two techniques used for the boundary detection is shown in Figure 4.2. More details on the pre-processing techniques are presented in Sections 4.3.2 and 4.3.3.

### 4.3.1 Properties of an Edge Map

An edge map for an image is derived from the image itself and is characterized by high image intensity gradient values near the edges (intensity discontinuities).

The following properties of edge maps are exploited in this work.

- Image intensity gradient vectors are oriented perpendicular to the local edge orientation.
- Image intensity gradient vectors have large magnitudes only in the immediate vicinity of the edges or intensity discontinuities.
- In homogenous regions, the magnitude of the image intensity gradient is nearly zero.

### 4.3.2 Detection of the Vessel Boundary

The magnitude of the image intensity gradient between the vessel boundary and its background is typically large and is easily computed using standard edge detection techniques. There are several edge map generation equations according to image processing literature. Equations (2) or (3) are appropriate for gray-scale images and equations (4) or (5) are more suited for binary images. Since, this project operates on gray-scale images, equation (3) was used as re-stated in equation (15) to generate an edge map image. A simple equation for the external force, $E_{ext}$ was used to generate the edge map as described in (15).

$$E_{ext}(x,y) = -| \bigtriangledown [G_{\sigma}(x,y) * I(x,y)]|^2 \tag{15}$$

where,

$E_{ext}(x,y)$ is the external force equivalent

$I(x, y)$ is the input image

$G_\sigma(x, y)$ is a two-dimensional Gaussian function with standard deviation $\sigma$ and

$\bigtriangledown$ is the gradient operator

A Gaussian filter is generated using a $\sigma$ value of 1.5. The image is smoothed using this Gaussian filter and the gradient of the image is calculated using a simple gradient operator. The square of the thus obtained gradient magnitude generates the edge map of the image.

### 4.3.3 Detection of the Lumen boundary

Unlike the vessel boundary, the lumen boundary has a very small contrast with respect to its background and hence cannot be extracted using a simple image gradient intensity computation procedure as described above. Hence an adaptive $k$-means clustering algorithm was used to detect the lumen boundary.

The images were initially subjected to Gaussian smoothing using the same Gaussian filter used in the detection of the outer vessel boundary and then the adaptive $k$-means technique was applied on the images. The smoothed image pixel intensity values define the feature vector for $k$-means clustering. The feature vector for $k$-means clustering is the data on which the $k$-means clustering technique is applied. So, essentially, the Gaussian smoothed pixel intensity values are the values being clustered into multiple clusters in the image.

The basic $k$-means algorithm is described in Section 1.6 and the adaptive $k$-means algorithm is described in Section 1.7. The histogram of the pixel intensities is generated which is then partitioned into $k$ clusters. The initial cluster centroids are generated at equal intervals along the range of the histogram intensities. The adaptive $k$-means algorithm is carried out until the correlation coefficient between the original and the segmented images is greater than 0.99. The high correlation coefficient value signifies that the segmented image is an appropriate fit on the original image.

The value of $k$ is not pre-set, and hence there is no need for human intervention. The algorithm itself obtains an appropriate value for $k$, suited for the image in question. Since the value of $k$ adapts according to the image, it is different for every image and hence the name 'adaptive $k$-means clustering'.

## 4.4 GVF Field and Snake Deformation

The GVF process is shown in Figure 4.3 and is explained in detail in the following sections.

The generation of the GVF field and the GVF snake deformation process are carried out on both the edge map image and the adaptive $k$-means segmented image obtained in the pre-processing step to generate the vessel and lumen contours respectively. Although GVF is a potential external force, it cannot be represented as the negative gradient of a potential function like the other potential forces [1, 2]. The result is that GVF cannot be represented by the standard energy minimization framework and instead is represented by a force balance equation as shown in (6), (7), and (8).

The GVF field is diffused from the edge information into the homogenous regions of the image to create a capture range for the active snake contour to work on. An initial contour is then initialized in the image, which is then deformed towards the edges, because of the effect of the capture range and the external forces. When the deformation stops, typically when the contour reaches the edges, the final contour is obtained, which represents the appropriate boundary.

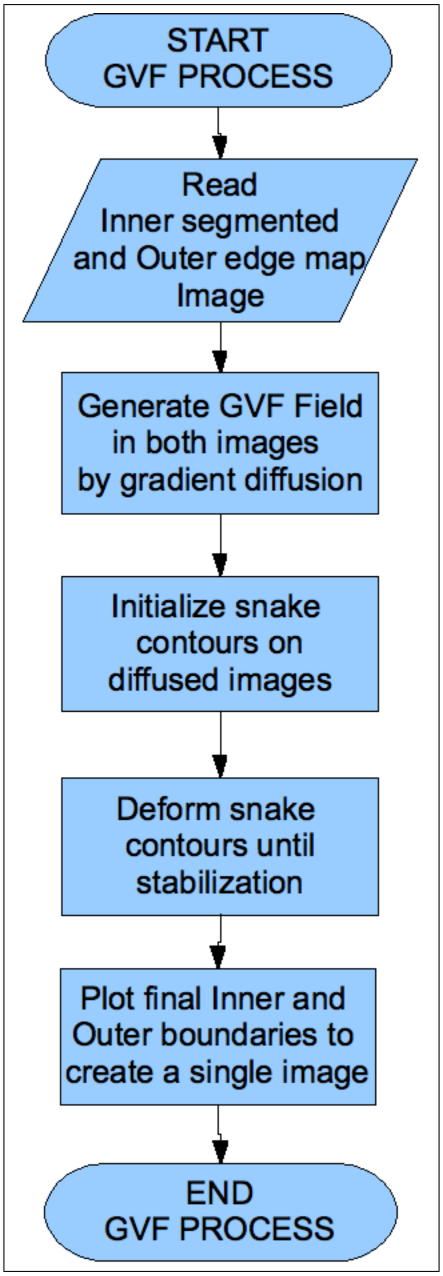

Figure 4.3: GVF Process

### 4.4.1 Gradient Vector Flow Diffusion

The objective of the GVF diffusion process is to retain the property of the edge map that the gradient is large near the edges, and to extend this gradient further away from the edges and into the homogenous regions thus creating a large capture range. A computational diffusion process was used for the diffusion of the gradient information into the homogenous regions. An added benefit of the GVF diffusion process is that the diffused vectors point into boundary concavities. This is important for this project because this causes the GVF snake to deform into plaque boundary concavities, hence generating an appropriate fit for the inner lumen boundary.

GVF fields are dense vector fields derived from images by minimizing a certain energy functional in a variational framework. The minimization is achieved by solving the pair of decoupled linear partial differential equations, (11) and (12). Solving these equations using equations (13), (14) enables the diffusion of the gradient vectors at the object boundaries into the surrounding homogenous regions, thus creating an appropriate capture range for the snake contour.

### 4.4.2 Snake Initialization

Deciding the initial snake contour for the GVF deformation process is a complicated task. Hence, to keep things simple, a circular contour with a small radius at the center of the image as shown in Figure 4.4 was used to initialize the snake contour for GVF deformation. The figure shows the circular initial contour which is then deformed for the generation of the final outer vessel boundary. Similarly, another circular contour with only a slightly smaller radius was created and deformed for the generation of the final inner lumen boundary.

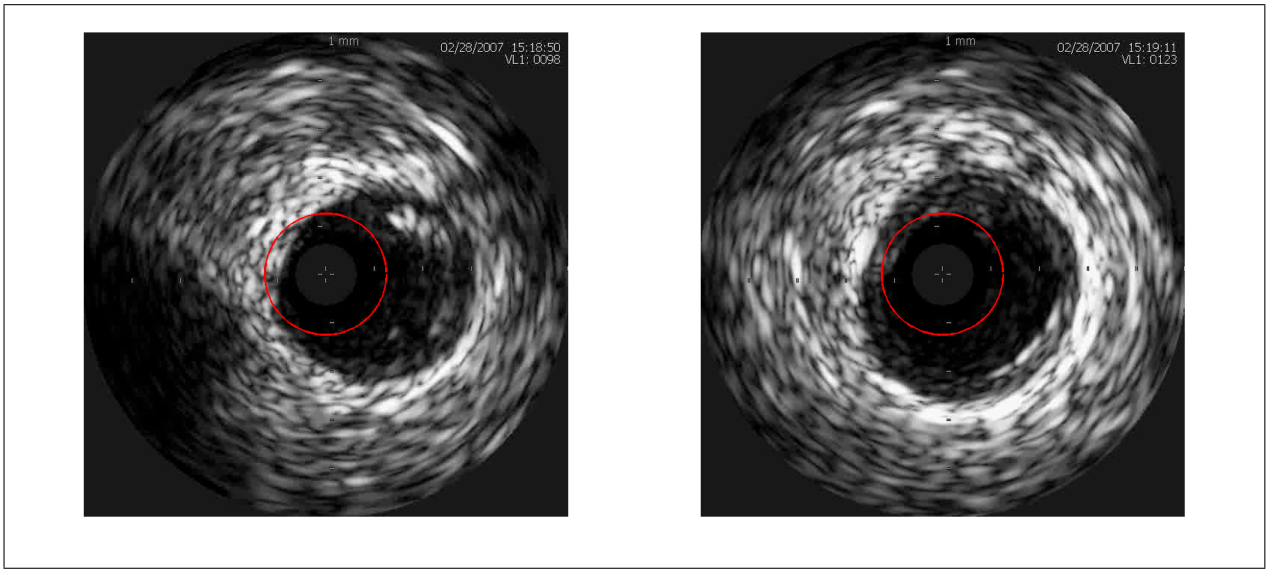

Figure 4.4: Initial circular contour, in red, superimposed upon the original images

### 4.4.3 Snake Deformation and Convergence

The deformation of the snake follows a standard snake deformable model. The initial contour goes through a series of iterations and deforms when it gets pulled by the gradient vector forces towards the boundaries. Since the capture range of the GVF field is very large, the initial position of the snake does not affect the final deformed position as long as there are no large artifacts in the center of the image, which can possibly block the snake deformation or movement of the snake towards the required boundary.

Since, as earlier seen, the GVF field has pulling properties into the boundary concavities, the GVF snake easily deforms even into sharp concavities in the edges of the images. The GVF field obtained from the diffusion process conducted previously replaces **v** in equation (9). Throughout this project, for all the snake contours, $\alpha = 0.6$ and $\beta = 0$. The iterative solution to equation (9) results in the generation of the final snake contour.

## 4.5 3D Reconstruction and Volume Visualization

The 3D volumization process is shown in detail in Figure 4.5. A set of contours is obtained for every image from the GVF deformation process, the outer vessel contour and the inner lumen contour. For 3D visualization, these contours need to be stacked on top of each other at regular intervals i.e., with equal spacing between two contour sets. This is done to replicate the fact that IVUS images are generated at regular intervals using the ultrasound catheter.

However, to replicate a coronary artery, there is a need to regenerate several contours in between each set of these contours. This is necessary to create a 3D tubular structure which graphically re presents the arterial outer and inner boundaries. A simple linear interpolation technique was used to generate the interpolating contours. The number of interpolating contours between each set of images is kept constant throughout the experiment and is currently set to 100 intermediate sets of contours. This value can however be set to a user-defined parameter $n$.

The linear interpolation method is as follows:

1. Consider two contours $c_1$ and $c_2$.

2. For every point $(x_1, y_1)$ on the contour $c_1$, find the nearest neighboring point $(x_2, y_2)$ on the next contour, $c_2$.

3. Let $i = 1$. For every $i = 1$ to $n$, generate $n$ interpolating points $(x_i, y_i)$ using a simple linear interpolation formula.

   $x_i = \frac{(n-i)x_1}{n} + \frac{ix_2}{n}$ and $y_i = \frac{(n-i)y_1}{n} + \frac{iy_2}{n}$

4. Generate $n$ interpolating points for every point on the contours, $c_1$ and $c_2$.

5. Repeat steps 1 to 4 for every set of contours. This process will create $n$ interpolating contours between each set of contours.

The outer vessel and inner lumen contours obtained from the GVF snake deformation and the linear interpolation are then plotted and saved as binary images showing the lumen contours inside the vessel contours. The area between the lumen and the vessel contours is filled in black. This area represents the plaque volume in that image. These sets of contours are then stacked consecutively to obtain a 3D volume data. An isosurface of the 3D volume data is then generated with an isovalue of 0.5.

An isosurface is a surface reconstruction, which connects all points in the 3D data with a specific isovalue. More specifically, an isosurface retains all the pixels in the data volume where there is a transition around the given isovalue. Since, the set of images being used in this project are binary images, isosurface retains points which have a transition from 0 to 1 and vice versa. Hence an isovalue of 0.1, 0.2 upto 0.9 would all generate the same isosurface in this case. The rest of the points in the images which do not have a transition are ignored. Hence we obtain a 3D tube like structure with inner and outer tubes, which represent the inner lumen and outer vessel boundaries respectively.

The isosurface function from the MATLAB library [30] is used for this purpose. The isosurface function returns the faces and vertices of the 3D volume data, which is written into a MATLAB *'obj'* file. The 3D data model written into a MATLAB *'obj'* file is then converted into a VTK (Visualization Tool Kit) formatted file for enabling volumetric visualization. ParaView (Parallel Visualization Application) [31], is then used to read this VTK formatted file, which in turn reconstructs the model for easy 3D visualization.

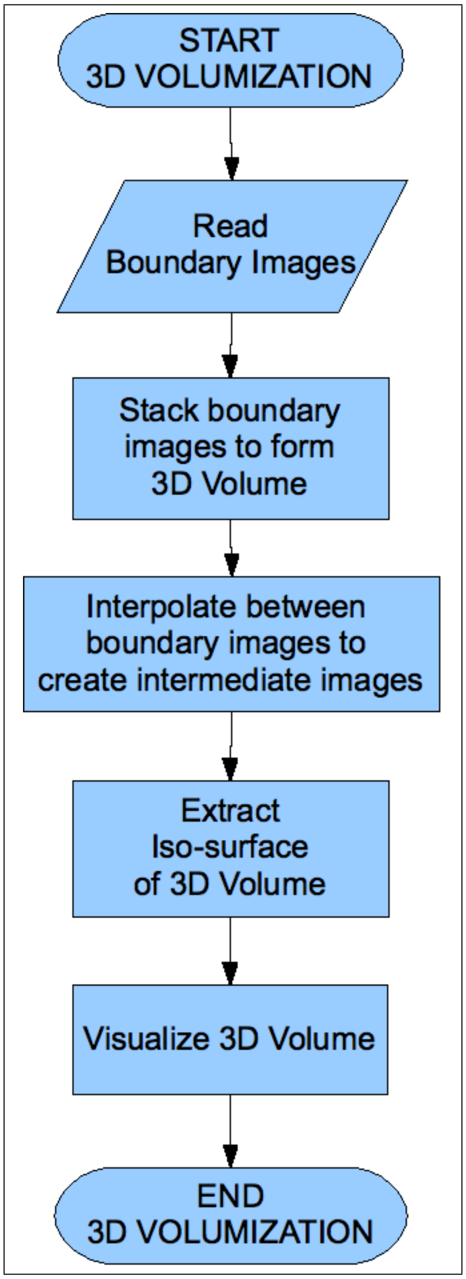

Figure 4.5: Volumization Process

# Chapter 5

# Experimental Results

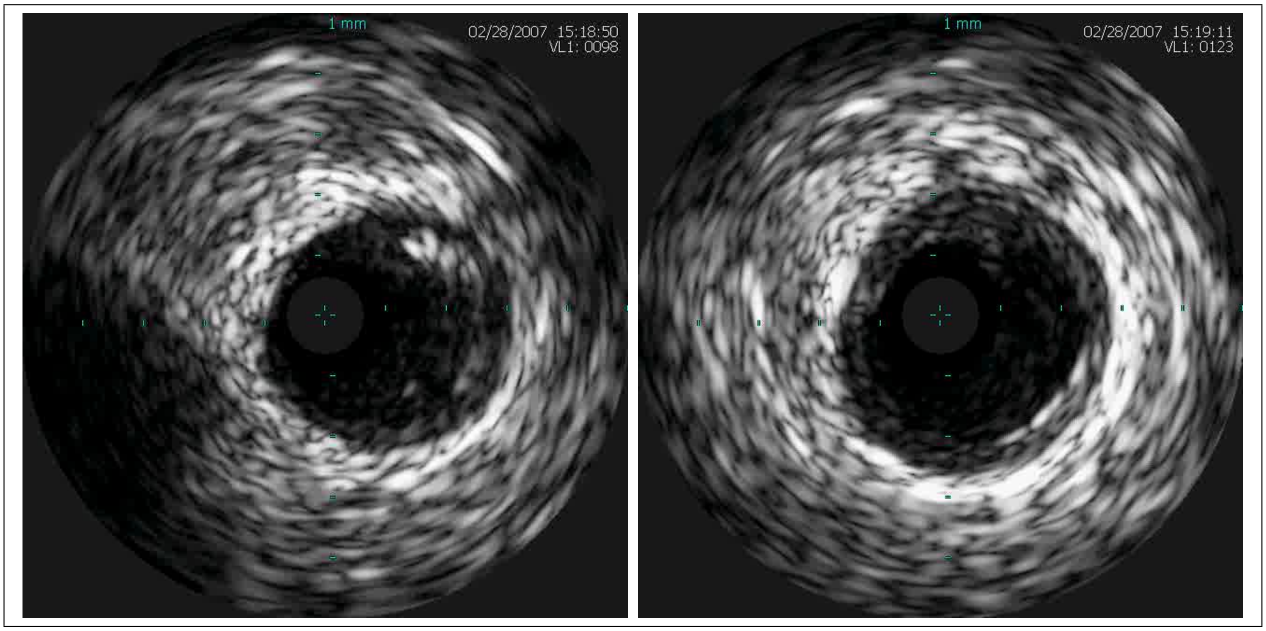

Figure 5.1: Original Images

Figure 5.1 shows the original cross-sectional IVUS images in grayscale. As seen from the above images, the outer vessel boundary is clearly visible, but the inner lumen boundary does not have a good contrast with its background.

## 5.1 Pre-Processing

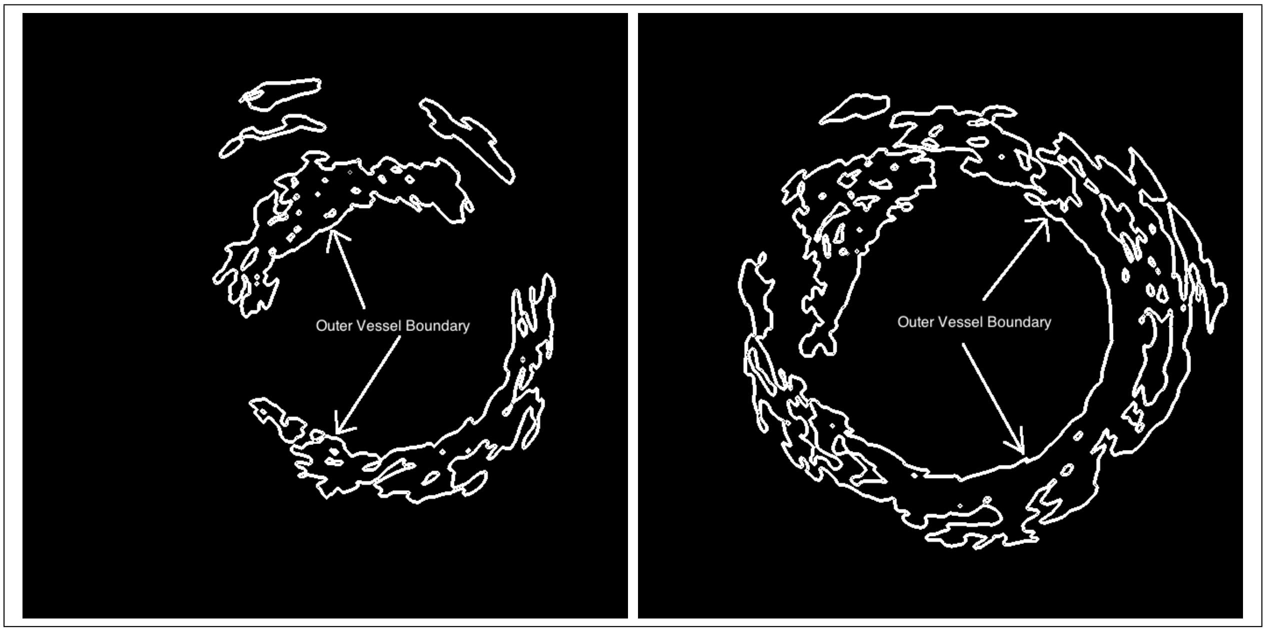

Figure 5.2: Edge Maps derived via image intensity gradient computation

Figure 5.2 shows the edge maps derived from the cross-sectional IVUS images by computing the image intensity gradient function using equation (15). The resultant edge maps show that the outer vessel boundary is easily detectable with a gradient operator, but the inner lumen boundary requires a specialized technique for boundary detection.

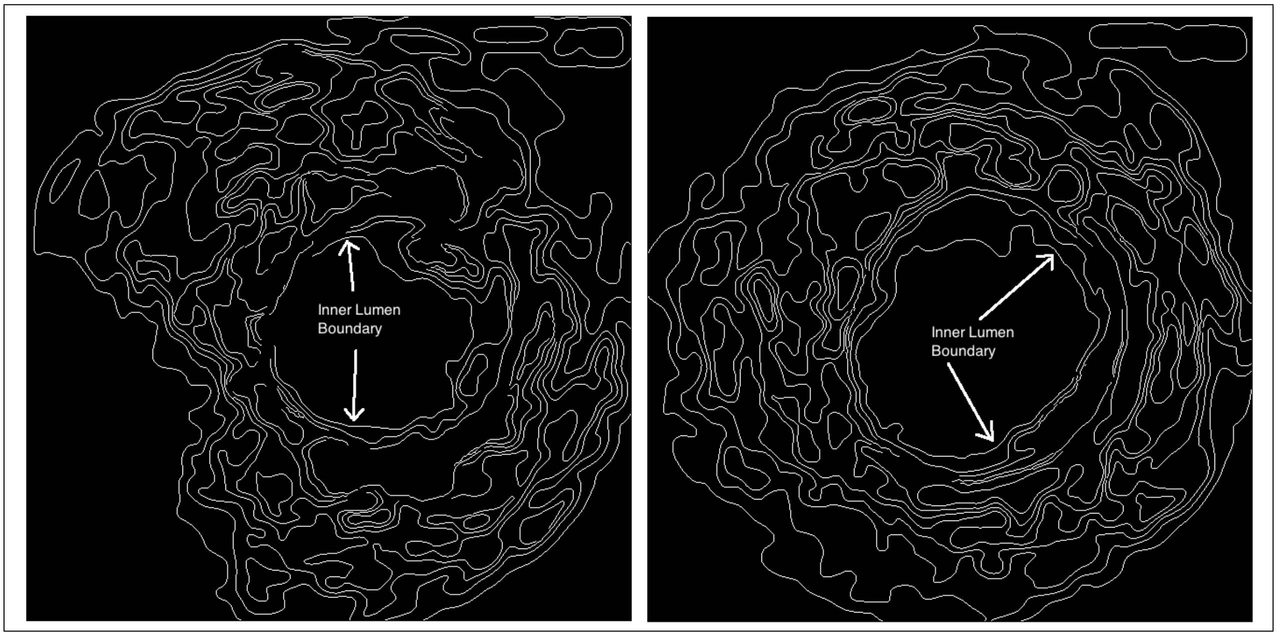

Figure 5.3: Segmented images obtained through adaptive $k$-means clustering (showing boundaries around different segmented regions)

Figure 5.3 shows the adaptive $k$-means clustered images obtained by performing an adaptive $k$-means clustering on two of the cross-sectional IVUS images. The resultant image shows a series of segmented regions in the image each separated by edges. The innermost region is the lumen region and the edge surrounding it is the lumen boundary. The edges are generated using a canny edge detector.

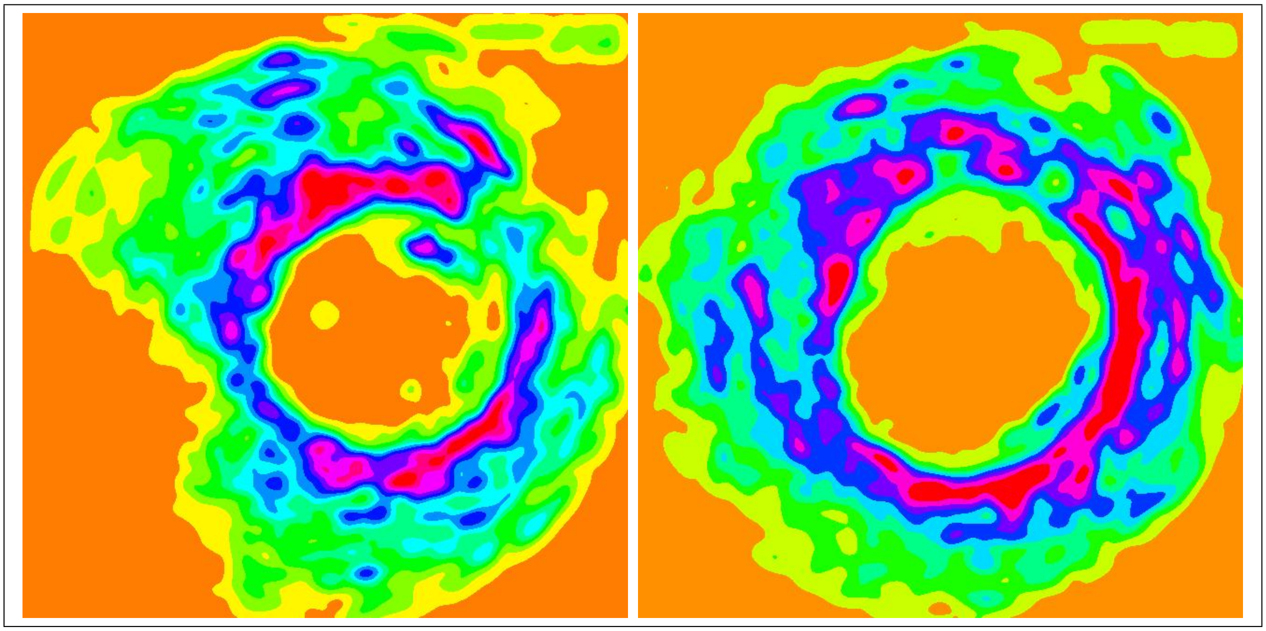

Figure 5.4: Segmented images obtained through adaptive $k$-means clustering (showing different regions in different colors)

Figure 5.4 shows the segmented images obtained from the adaptive $k$-means clustering on two IVUS images. The segmented images show each region in the segmented image in a different color. Similar to Figure 5.3, the innermost region represents the lumen area. Adaptive $k$-means clustering generates a different number of segmented regions for different images.

## 5.2 Snake Initialization

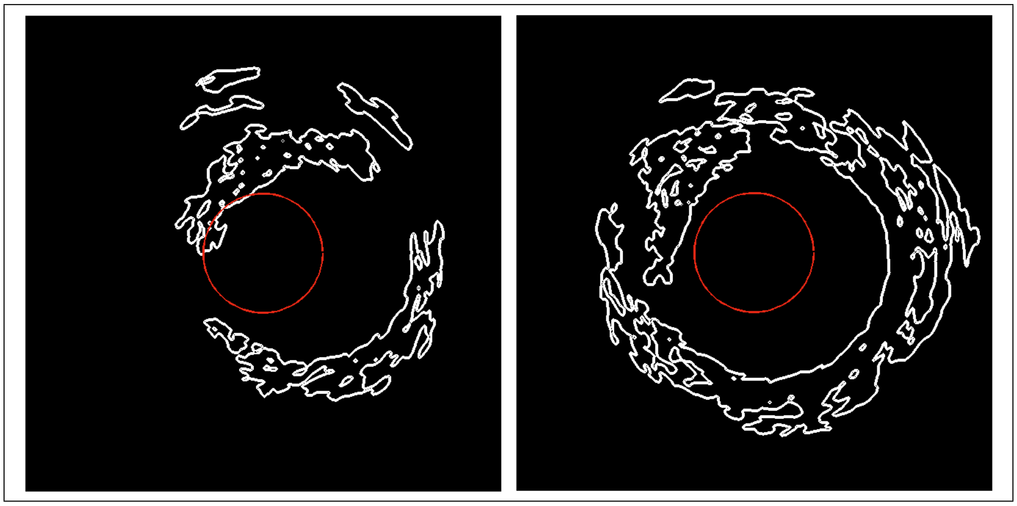

Figure 5.5: Initial contours, in red, superimposed on simple gradient edge map images

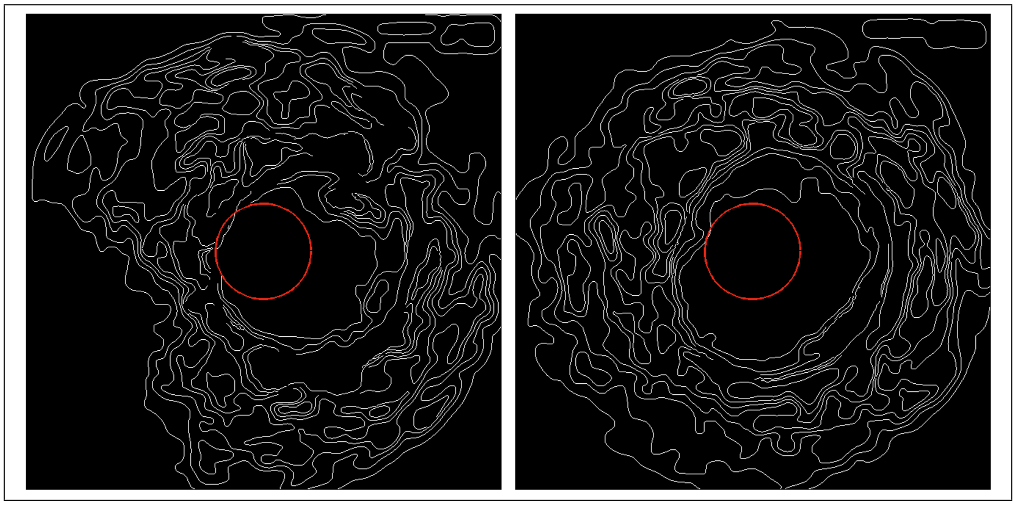

Figure 5.6: Initial contours, in red, superimposed on adaptive $k$-means clustered images

Figures 5.5 and 5.6 show the initial circular snake contours, as described in Section 4.4.2 superimposed on the pre-processed images obtained from the pre-processing step. As seen from the images, in some cases, the initial contours are overlapping over the desired edges, but can still deform to a correct final result as seen in later results.

## 5.3 GVF Field Diffusion

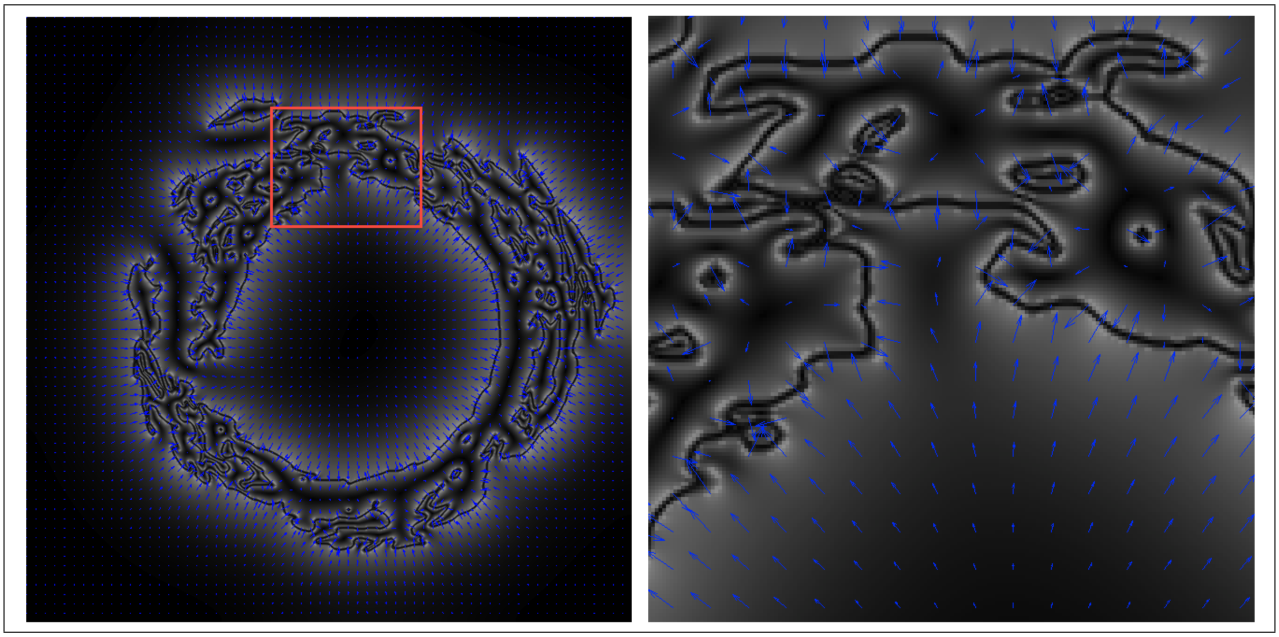

Figure 5.7: Diffused Gradient Vector Field (left) and its close-up view (right)

Figure 5.7 shows the GVF field diffused into the homogenous regions and the close-up view of one of the concavities in the image (the red box in the image on the left), which shows the GVF vectors pointing into the concavity (image on the right).

## 5.4 GVF Snake Convergence

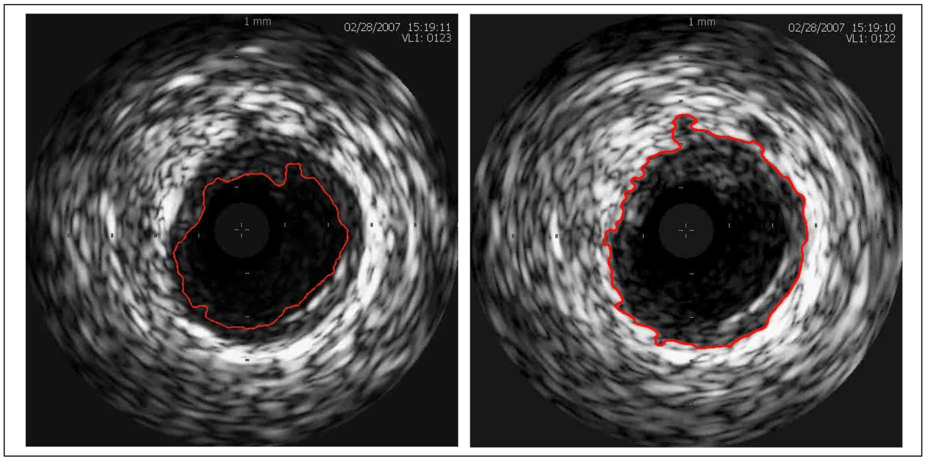

Figure 5.8: Lumen (left) and Vessel (right) contours, in red, superimposed upon original images

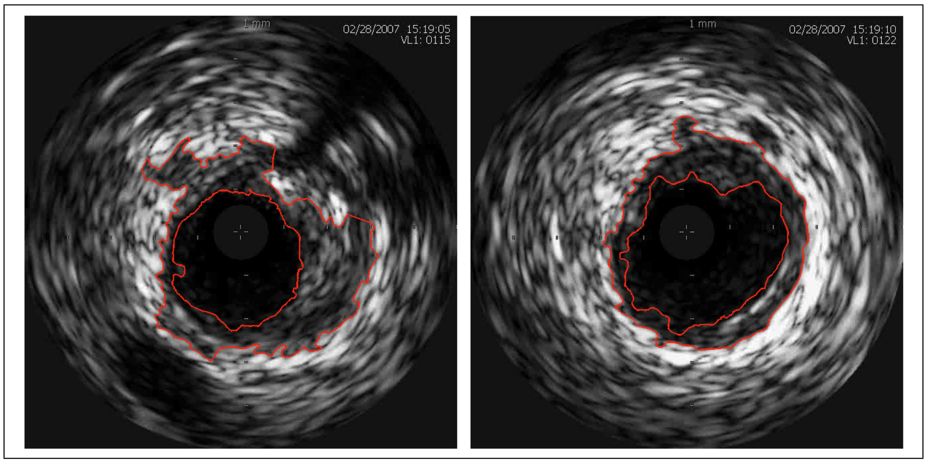

Figure 5.9: Lumen and Vessel contours, in red, superimposed upon original images

Figures 5.8 and 5.9 show several of the final GVF snake contours (in red) of both the inner lumen boundary and the outer vessel boundary, generated from the deformation of the initial contours and superimposed upon the respective original cross-sectional IVUS images.

## 5.5 Final Contour Plotting and Reconstruction

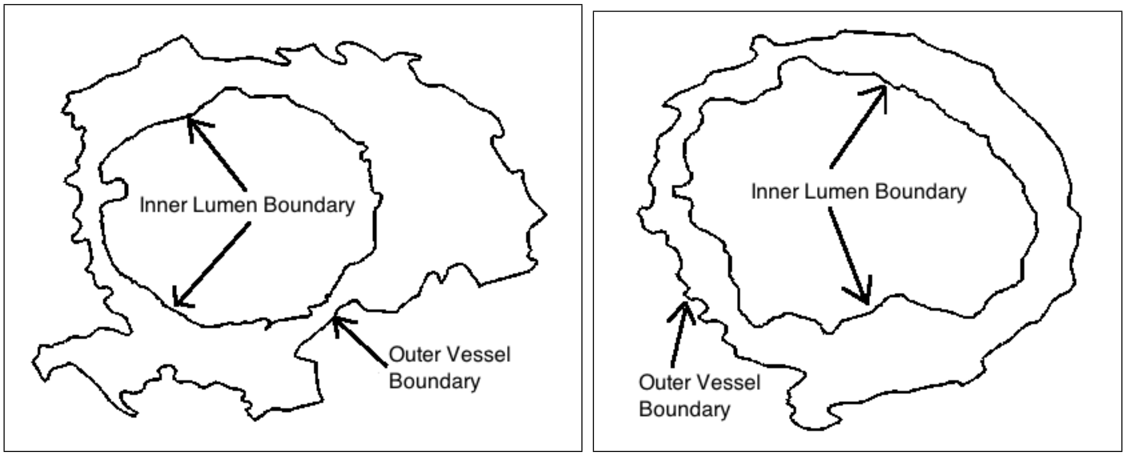

Figure 5.10: Contours extracted and plotted in black

Figure 5.10 shows the final GVF contours plotted separately as black contours on a white background. These contours have further been used for 3D reconstruction of the coronary arteries and their volumetric visualization as described in Section 4.5.

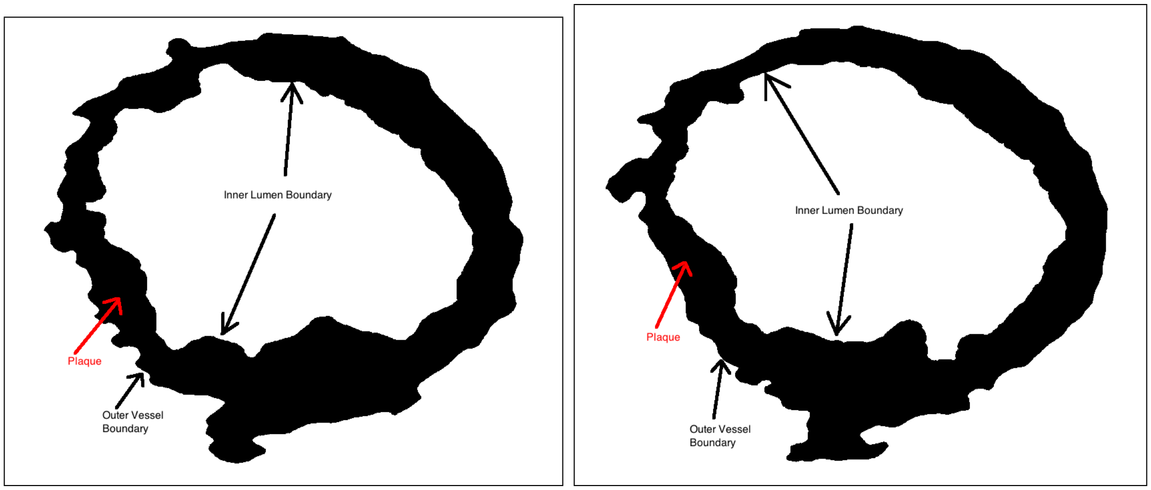

Figure 5.11: Contours extracted and plotted and filled to show the plaque area

Figure 5.11 shows the final GVF contours plotted separately as black contours on a white background. The area between the two contours is filled with black to represent the plaque deposits. These images and then used in the 3D reconstruction of the tubular volume for appropriate visualization.

## 5.6 3D Reconstruction and Volume Visualization

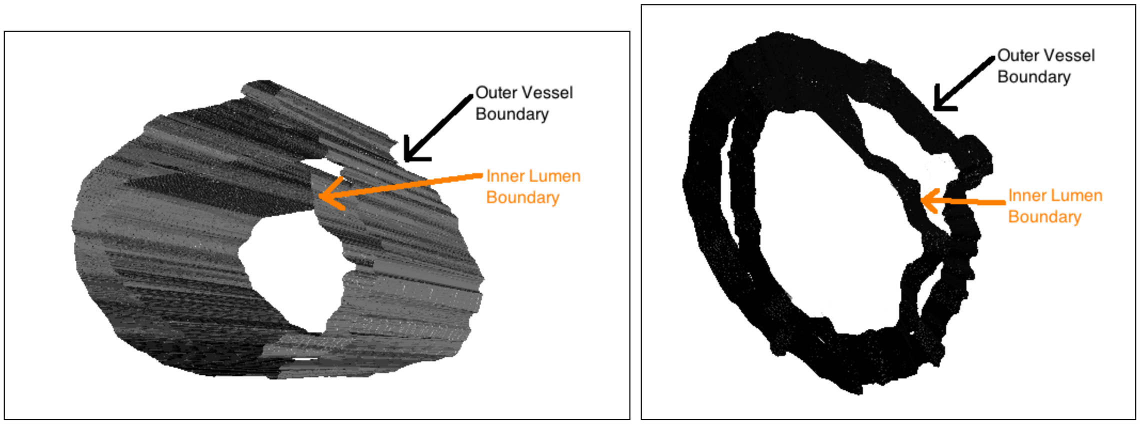

Figure 5.12: Different Views of the same reconstructed 3D isosurface volume

Figure 5.12 shows two different views of the regenerated 3D isosurface volume with the outer vessel and inner lumen boundaries. We can see that there is an area in between the two contours, which is empty. This area represents the plaque deposits.

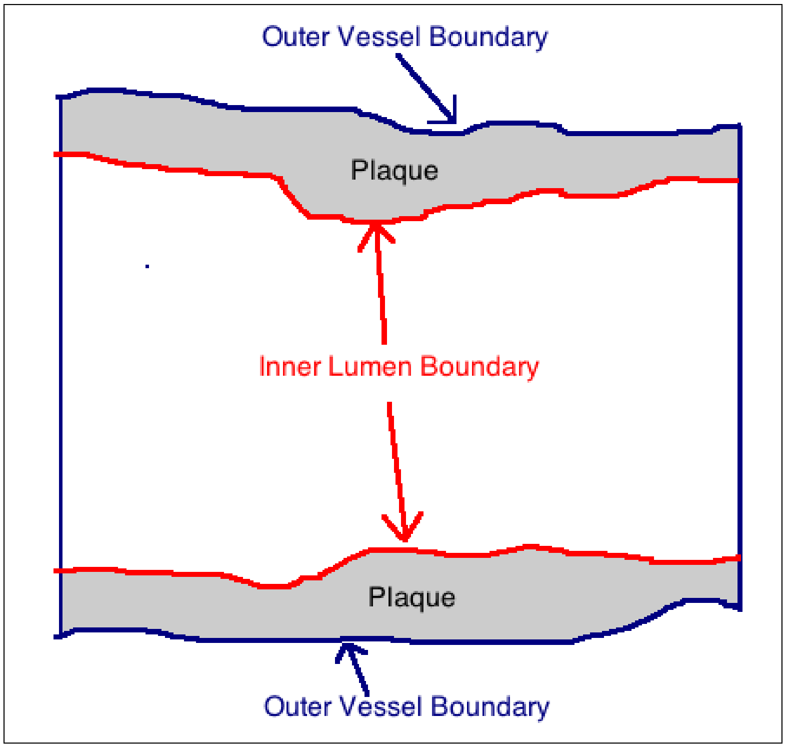

Figure 5.13: Transverse section of regenerated 3D volume

Figure 5.13 shows a transverse section of the 3D regenerated volume. The transverse section shows the outer vessel (in dark blue) and inner lumen (in red) boundaries. The filled grey area represents the plaque deposits.

# Chapter 6

# Conclusions and Future Work

The work in this project has been performed on cross-sectional IVUS images to enable the 3D reconstruction of the coronary arteries, mainly the outer vessel and inner lumen boundaries to enable physicians to easily study the inner arterial walls for plaque deposits and to recommend appropriate treatment. To summarize the work in this project:

1. Cross-sectional IVUS images were pre-processed to obtain edge maps and segmented images suitable for the detection of the vessel and lumen boundaries respectively.

2. GVF field in the images was generated by the diffusion of the gradient in the pre-processed images resulting from the gradient computation and adaptive $k$-means clustering techniques.

3. Initial snake contours were deformed using the GVF field which resulted in the final vessel and lumen boundaries upon suitable convergence.

4. Final contours obtained from snake deformation were reconstructed as a 3D model and were visualized as a 3D tubular structure using an appropriate 3D visualization software.

The experimental results obtained are almost the same as the manual boundary detection carried out by experts in the field. The results from the automated process are more accurate than the contour detection by experts because, the GVF snake contour moves into boundary concavities more accurately than can be traced by the experts. This makes the GVF snake-based detection process automated as well as efficient since it does not require human intervention and it also gives a better quality result.

The salient features the GVF active contours are as follows:

- External forces and distance potential forces do not point into boundary concavities. The GVF technique has a major advantage over them, because it progresses into concavities, allowing the active contour to model the boundary concavity.

- GVF field has a large capture range and does not distort the boundary.

- Pressure force contours need to be initialized either inside or outside the boundary because they can push the contour only in one direction. GVF, however can be initialized across boundaries as well.

- GVF is robust against image noise because it can smooth out weak gradients and retain stronger ones.

However, there are also problems associated with this technique. Extensive experiments carried out in the duration of this project show that, if there are large noisy artifacts blocking the movement of the contour, then the snake contour stops deforming and gets conformed to the boundary of these artifacts.

## 6.1 Future Work

From the experiments carried out in this project, it can be seen that GVF is a robust technique and gives excellent results for all images, even with the same initial contour being used in every single one of the cross-sectional images. Hence, GVF snakes do not require human intervention for defining the position of the initial contour. However, the only problem discovered with GVF is that it cannot progress to the desired edges if there are noisy artifacts like shadows from the plaque blocking its way. Hence, a robust pre-processing technique is necessary to remove noise from the images. Again, this should be a totally independent technique and hence, we propose a learning-based technique which can detect noisy artifacts in the images and eliminate them.

For purposes of simplification, a generalized linear interpolation technique was used to generate the intermediate contours during 3D reconstruction. Also, the number of intermediate contours has been assumed to be a user-defined parameter, which might not be the case with actual coronary arteries. This can however be remedied by learning the exact distance between two consecutive images for a more realistic feel to the visualization and the interpolation technique can be adapted accordingly. Also a more sophisticated interpolation technique might result in a volumetric visualization which is aesthetically appealing to the eye.

# Bibliography

[1] C. Xu and J.L. Prince, *Gradient Vector Flow: A New External Force for Snakes*, Proc. IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR), Los Alamitos: Comp. Soc. Press, pp. 66-71, June 1997

[2] C. Xu and J. L. Prince, *Snakes, Shapes, and Gradient Vector Flow*, IEEE Transactions on Image Processing, 7(3), pp. 359-369, March 1998.

[3] C. Xu and J. L. Prince, *Generalized Gradient Vector Flow External Forces for Active Contours*, Signal Processing — An International Journal, 71(2), pp. 131-139, December 1998.

[4] C. Xu, D. L. Pham, and J. L. Prince, *Medical Image Segmentation Using Deformable Models*, SPIE Handbook on Medical Imaging – Volume III: Medical Image Analysis, edited by J.M. Fitzpatrick and M. Sonka, May 2000.

[5] C. Xu and J. L. Prince, *Gradient Vector Flow Deformable Models*, Handbook of Medical Imaging, edited by Isaac Bankman, Academic Press, September, 2000.

[6] P. Radeva, *On the Role of Computer Vision in Intravascular Ultrasound Image Analysis*, Chapter in Angiography and Plaque Imaging: Advanced Segmentation Techniques, Editorial: CRC, Boca Raton, FL, USA, 2003, ISBN: 0849317401, pp.397-450.

[7] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models* Int. J. Comput. Vis., vol. 1, pp. 321331, 1987

[8] V. Caselles, F. Catte, T. Coll, et al., *A Geometric Model for Active Contours in Image Processing*, Numerische Mathematik, 66(1): 1-31 October 1993.

[9] K. Djamel, W. Puech and B. Rossetto, *Geometric Active contour model using level set methods for objects tracking in images sequence*, Int. Conf. Setit, Tunisie, 2004.

[10] C. Xu, A. Yezzi, Jr., and J. L. Prince, *On the Relationship between Parametric and Geometric Active Contours*, in Proc. of 34th Asilomar Conference on Signals, Systems, and Computers, pp. 483 -489, October 2000.

[11] C. Xu and J. L. Prince, *Global Optimality of Gradient Vector Flow*, Proc. of 34th Annual Conference on Information Sciences and Systems (CISS'00), Princeton University, March 2000

[12] C. Xu, *Deformable Models with Application to Human Cerebral Cortex Reconstruction from Magnetic Resonance Images*, PhD Dissertation, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltiomre, MD, 21218, USA.

[13] W. F. Ames, *Numerical Methods for Partial Differential Equations*, Boston: Academic Press, 3rd ed., 1992.

[14] C. Xu, A. Yezzi, Jr., and J. L. Prince, *A Summary of Geometric Level-Set Analogues for a General Class of Parametric Active Contour and Surface Models*, in Proc. of 2001 IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM 2001), pp. 104-111, July 2001.

[15] D. Terzopoulos and K. Fleischer, *Deformable models*, The Visual Computer, 4:306331, 1988.

[16] T. McInerney and D. Terzopoulos, *Deformable models in medical image analysis: a survey*, Med. Imag. Anal., 1(2):91108, 1996.

[17] T. McInerney and D. Terzopoulos, *Topologically adaptable snakes*, In Proc. Intl Conf. Comp. Vis., pages 840845, 1995.

[18] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Intl J. Comp. Vis., 22:6179, 1997.

[19] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.

[20] H. Tek, B.B. Kimia, *Image segmentation by reaction-diffusion bubbles*, pp.156-162, Fifth International Conference on Computer Vision (ICCV'95), 1995.

[21] L. D. Cohen and I. Cohen, *On active contour models and balloons*, CVGIP: Image Understanding, 53(2): 211-218, March 1991.

[22] M. Butenuth, Topology-preserving network snakes, *International Archives of Photogrammetry*, Remote Sensing and Spatial Information Sciences, Beijing, XXXVII (Part B3a), 2008, pp. 229234.

[23] A. K. Mishra, P. W. Fieguth, and D. A. Clausi, *Decoupled Active Contour (DAC) for Boundary Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence 99.

[24] N. Ray, S. T. Acton, T. Alten and E. E. de Lange, *MRI ventilation analysis by merging parametric active contours*, Proceedings of IEEE ICIP 2001, pp.861-864, Thessaloniki, Greece, October 2001.

[25] Bovenkamp, E.G.P., Dijkstra, J., Bosch, J.G., Reiber, J.H.C, *Multiagent IVUS image interpretation*, SPIE Proceedings: Medical Imaging 2003: Image Processing, San-Diego, California, USA, vol 5032, 2003, pages 619630.

[26] Brathwaite, P.A., Chandran, K.B., McPherson, D.D., Dove, E.L, *Lumen Detection in Human IVUS Images Using Region-Growing*, IEEE Computers in Cardiology, pages 3740, 1996.

[27] Sonka, M., Zhang, X., Siebes, M., Bissing, M.S., DeJong, S.C., Collins, S.M., McKay, C.R, *Segmentation of Intravascular Ultrasound Images: A KnowledgeBased Approach*, IEEE T Med Imaging, vol. 14 (4), pages 719732, 1995.

[28] A. Martelli, *An application of heuristic search methods to edge and contour detection*, CACM, vol. 19, p.73, 1976.

[29] http://www.mathworks.com/products/matlab/

[30] http://www.mathworks.com/help/techdoc/ref/isosurface.html

[31] http://www.paraview.org