

POLYP ENHANCING LEVEL SETS FOR CT COLONOGRAPHY

by

Ender Konukoğlu

B.S., in E.E., Boğaziçi University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronic Engineering

Boğaziçi University

2005

POLYP ENHANCING LEVEL SETS FOR CT COLONOGRAPHY

APPROVED BY:

Assist. Prof. Burak Acar
(Thesis Supervisor)

Assist. Prof. Burak Gürel

Prof. Bülent Sankur

DATE OF APPROVAL: 15.06.2005

ACKNOWLEDGEMENTS

First of all I am thankful to my advisor Assist. Prof. Burak Acar for his guidance and support throughout these two years. It was an honor and privilege to work with him. I really enjoyed the hours of technical discussions as well as the friendship he has offered me.

I am grateful to Prof. Bülent Sankur for being one of the few who has supported me when I needed the most. In last three years, he has shared his wisdom and knowledge in numerous occasions, helping me understand the nature of things.

I would like to thank Assist. Prof. Burak Gürel for helping me in my thesis, being in my jury and both courses I have attended. Apart from these I am grateful to him for being a friend and an elder to me who showed me a righteous path to follow in life.

I would also like to give my special thanks to Celal Eşli for being a comrade, Erdem Yörük and Turgay Şimşek for being my older brothers when I missed the real one, Uğur Bozkaya for being the source of joy, Erhan Durusüt who shared the boredom BUSIM gave one at night time, Barış Özgül for calming me down when I flared up with uncontrollable anger and all others I forget to mention here.

I should not forget to thank David Paik, Sandy Napel and Christopher Beaulieu for their support that made this thesis possible. I am also grateful to everyone at the third floor of the Clark Center and the Lucas Center at Stanford University, CA, USA, for their kind hospitality and friendship.

This thesis was in part supported by grants from TUBITAK KARIYER-DRESS project (104E035) and from NIH (1R01 CA72023).

ABSTRACT

POLYP ENHANCING LEVEL SETS FOR CT COLONOGRAPHY

Computer Aided Diagnosis (CAD) in Computed Tomographic Colonography aims at detecting the locations of colonic polyps, which are the precursors of colon cancer. CAD algorithms increase the performance of radiologists in detecting polyps and decrease the variability of diagnostic accuracy among different readers. Most of the proposed CAD algorithms are based on the assumption that colonic polyps are spherical protrusions on the colon wall attached to the wall by a thin strip. Most colonic polyps deviate from the assumed shape therefore reduce the CAD performance.

In this thesis we propose a polyp enhancement algorithm for improved CAD. The underlying idea is to use geometric characteristics of the colon wall to enhance colonic polyps by evolving the whole wall via level set methods. There are two different level set flows proposed within the algorithm: joint curvature flow and volume maximizing constant surface flow. Both of the flows evolves colonic polyps so to increase their spherical symmetries, while preserving other structures on the colon wall, to increase the detection and identification performance of existing CAD algorithms.

ÖZET

BT KOLONOGRAFİ İÇİN POLİP GELİŞTİREN DÜZEY KÜMELERİ

Bilgisayarlı tomografi ile kolonografide bilgisayar destekli teşhisin (BDT) amacı kalın bağırsaktaki poliplerin bulunmasıdır. Bu poliplerin bir kısmı bağırsak kanserine dönüşebileceği için önceden sezim kritiktir. BDT algoritmaları radyologların polipleri bulmalarını kolaylaştırırken aynı zamanda farklı okuyucular arasındaki teşhis doğruluğundaki değişkenliği azaltır. Önerilen çoğu BDT algoritmaları kalın bağırsaktaki poliplerin, bağırsak duvarına ince bir iple bağlı küresel şekiller oldukları varsayımına dayanır. Halbuki çoğu polip bu varsayıma uymaz ve bu yüzden bu varsayıma dayalı BDT performansını düşürür.

Bu tezde gelişmiş BDT için polip şekillerini geliştiren bir algoritma sunulmuştur. Altta yatan fikir, yüzey kümeleri yöntemiyle, bağırsak yüzeyinin geometrik özelliklerini kullanıp bağırsak duvarını evriltmektir. Bu evrim sayesinde bağırsak poliplerinin şekilleri gelişmektedir. Yüzey kümeleri yöntemi dahilince, iki ayrı akış önerilmektedir: birleşik eğrilik akışı ve sabit yüzey alanıyla hacim enbüyüten akışı. Her iki akışta varolan BDT algoritmalarının performanslarını arttırmak için bir yandan bağırsak poliplerinin küresel bakışımını arttırırken diğer yandan bağırsak duvarındaki diğer şekilleri korur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	viii
LIST OF TABLES	xii
LIST OF SYMBOLS/ABBREVIATIONS	xiii
1. INTRODUCTION	1
1.1. Literature Review on CTC-CAD	1
1.2. Problem Definition and Proposed Solution	4
2. BACKGROUND ON LEVEL SET METHODS	7
2.1. The Deformable Models	7
2.2. Implicit Functions and the Signed Distance Function	9
2.3. Level Set Methods	11
2.3.1. The Evolution Equation	13
2.3.2. Numerical Implementation	15
2.4. Topology Preserving Level Sets	18
2.4.1. Digital Topology	19
2.4.2. Topology Preserving Level Set Algorithm	22
3. COLON WALL DEFORMATION	23
3.1. Segmentation and Preprocessing	23
3.2. Polyp Enhancing Level Set Flows	27
3.2.1. Joint Curvature Flow	27
3.2.1.1. Curvature Calculations	29
3.2.1.2. Directional Smoothing	31
3.2.1.3. Speed Function	35
3.2.2. Volume Maximizing Constant Surface Flow	41
3.2.2.1. Theoretical Derivation	41
3.2.2.2. Application to Polyp Enhancement	45
3.3. Implementation Issues	50

4. EVALUATION AND RESULTS	53
4.1. Methodology	53
4.1.1. Surface Normal Overlap	54
4.1.2. Free-response Receiver Operating Characteristic Analysis	55
4.2. Data Set	56
4.3. Experiments with JCF	57
4.3.1. Results	57
4.3.2. Discussions	63
4.4. Experiments with VMCSF	64
4.4.1. Results	64
4.4.2. Discussions	69
4.5. Comparison of JCF and VMCSF	71
5. CONCLUSIONS	76
APPENDIX A: SIMPLIFYING EULER-LAGRANGE EQUATIONS	78
REFERENCES	81

LIST OF FIGURES

Figure 1.1.	Types of polyps: (a) Sessile (b) Pedunculated	5
Figure 1.2.	2D illustration of the desired motion	6
Figure 2.1.	A circle implicitly represented by its level set function $\Phi(x, y)$. . .	10
Figure 2.2.	Implicit function example: (a) The original interface (b) The signed distance function	12
Figure 2.3.	Gradient vectors of the signed distance function are normal to the interface	12
Figure 2.4.	Smooth corner moving in its normal direction	17
Figure 2.5.	Entropy satisfying weak solutions for 1D interfaces embedded in 2D functions	17
Figure 2.6.	Types of neighborhoods for the point in the middle of the $3 \times 3 \times 3$ cubic grid in 3D digital domain	20
Figure 2.7.	An example of a topologically significant point	21
Figure 3.1.	The PELS flow chart	24
Figure 3.2.	Overlapping method	25
Figure 3.3.	Segmented colon wall example with the red fog representing the colon tissue	26

Figure 3.4.	Phantom colon wall illustrating a polyp and a fold	28
Figure 3.5.	The flow chart for the JCF	29
Figure 3.6.	The principal curvature directions: (a) The first principal vector (b) The second principal vector	33
Figure 3.7.	2D example of a nonsymmetric Gaussian kernel	34
Figure 3.8.	Curvature characteristics around a polyp	36
Figure 3.9.	Local neighborhoods having saddle-like curvature characteristics: (a) polyp neck (b) fold necks (c) junction of folds and (d) bumpy planes	37
Figure 3.10.	The graph of the speed function $F_{prop}(\kappa_1, \kappa_2)$: Line (a) corresponds to haustral fold apex and line (b) corresponds to fold necks	38
Figure 3.11.	Speed distributions (red = 4.0, blue = 0.0): (a) Original segmented surface (b) Speed distribution without directional smoothing (c) Speed distribution with directional smoothing	40
Figure 3.12.	Examples of the flow that evolves 3D shapes towards sphere: (a) Initial shape, a cube (b) After 10 iterations (c) After 25 iterations	45
Figure 3.13.	The red spots on the colon wall are the convex regions	47
Figure 3.14.	Cartoon: (a) The polyps will move towards spherical patches that are part of the smallest sphere they are contained in (b) The bumpy areas will become smoother as local H values will be small leading to a small H_0	49

Figure 3.15.	The flow chart for the VMCSF	51
Figure 4.1.	JCF results: (a) Original subvolume (b) Enhanced subvolume . . .	58
Figure 4.2.	Graph of SNO scores of enhanced polyps vs. original polyps . . .	59
Figure 4.3.	FROC curves for polyps larger than 9.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin	61
Figure 4.4.	FROC curves for polyps larger than 5.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin	62
Figure 4.5.	Problematic structures JCF is having trouble with: (a) PELS with JCF decreases the SNO score for this polyp (b) PELS with JCF introduces this false positive and (c) PELS with JCF increases the score for this false positive	65
Figure 4.6.	VMCSF results: (a) Original subvolume (b) Enhanced subvolume	66
Figure 4.7.	Graph of SNO scores of enhanced polyps vs. original polyps . . .	67
Figure 4.8.	FROC curves for polyps larger than 9.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin	68
Figure 4.9.	FROC curves for polyps larger than 5.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin	70
Figure 4.10.	Problematic structures VMCSF is having trouble with: (a) PELS with VMCSF introduces this false positive and (b) PELS with VMCSF increases the SNO score for this false positive	72

Figure 4.11. Graph of SNO scores of polyps enhanced with JCF vs. polyps enhanced with VMCSF	73
Figure 4.12. FROC curves for polyps larger than 9.00 <i>mm</i>	74
Figure 4.13. FROC curves for polyps larger than 5.00 <i>mm</i>	75

LIST OF TABLES

Table 2.1.	Comparison of parametric and geometric deformable models	8
Table 3.1.	Anisotropic smoothing coefficients	35

LIST OF SYMBOLS/ABBREVIATIONS

a	Iso-level Constant
A	Area
C_i	i^{th} Connected Convex Region
C_n	Set of all n-connected components
C_n^a	Set of all n-connected adjacent components
d	Distance Function
D	Diagonal Eigenvalue Matrix
e	Exponential Function
e_{II}	First Component of the Second Fundamental Form
E_I	First Component of the First Fundamental Form
f	Arbitrary Function
f_{II}	Second Component of the Second Fundamental Form
F	Speed Function
F_I	Second Component of the First Fundamental Form
F_{prop}	Propagation Speed
\vec{F}_{adv}	Advection Vector
F_{curv}	Curvature Speed
g	Spatially Varying Function
g_{II}	Third Component of the Second Fundamental Form
G_I	Third Component of the First Fundamental Form
H	Mean Curvature of a Surface
H_0	Mean Curvature Estimate
\overline{H}	Hamiltonian
H	Hessian Matrix
J	Functional
\hat{J}	Constrained Functional
$k_{\vec{t}}$	Directional Curvature
m	A Point Detected by CAD
n	Discrete Time Index

\vec{n}	Normal Vector of a Curve
\vec{N}	Normal Vector Map of a Surface
\vec{N}_p	Normal Vector of a Surface at a Point
N_m	Set of m-connected points including the center
N_m^*	Set of m-connected points excluding the center
p	Point on a Surface
\mathbf{r}	Position Vector
\mathbf{r}_I	Position Vector in a Set
$\mathbf{r}_{extremum}$	Extremum Point
R_{limit}	Radius of the Smallest Polyp
S	Surface Parameterization
t	Time Variable
\vec{t}	Unit Vector in the Tangent Plane
\vec{t}_i	i^{th} Principal Vector
Δt	Time Step
T_0	Fixed Time
T_m	Topological Number
V	Eigenvector Matrix
w	Solution of the Poisson Equation
X	Set of 26-connected points in the interior region
\bar{X}	Set of 26-connected points in the exterior region
δ	First Variation
ε	Small Number
κ	Curvature
κ_i	i^{th} Principal Curvature
λ	Lagrange Multiplier
λ_i	i^{th} Eigenvalue
ϱ	Sphericity
σ^2	Variance
σ_0^2	Variance Threshold

Γ	Interface
Γ_i^n	Convex Connected Iso-level Patch
Δ	Laplacian Operator
Σ	Covariance Matrix
Φ	Implicit Function
Ω	Subset
Ω^-	Interior of the Subset
Ω^+	Exterior of the Subset
$\partial\Omega$	Boundary of Ω
CAD	Computer Aided Detection/Diagnosis
CT	Computed Tomography
CTC	Computer Tomographic Colonography
CV	Curvedness
DGC	Directional Gradient Concentration
FROC	Free-response Receiver Operating Characteristics
FOC	Fiber-optic Colonoscopy
FP	False Positive
FPF	False Positive Fraction
GC	Gradient Concentration
HDF	Heat Diffusion Fields
HU	Hounsfield Unit
ITK	Insight Toolkit
JCF	Joint Curvature Flow
LDA	Linear Discriminant Analysis
PDE	Partial Differential Equation
PELS	Polyp Enhancing Level Sets
QDA	Quadratic Discriminant Analysis
ROC	Receiver Operating Characteristics
SI	Shape Index
SNO	Surface Normal Overlap

SVM	Support Vector Machines
TP	True Positive
TPF	True Positive Fraction
VMCSF	Volume Maximizing Constant Surface Flow
VTK	Visualization Toolkit

1. INTRODUCTION

Three to five percent of the population in the developed world will eventually be diagnosed with colon cancer. Even now colon cancer is one of the top leading cause of cancer death in the world. The colon cancer grows on the inner surface of the colon and develops from mushroom-like structures, called polyps. The size of polyps may vary greatly, from 0.3 *cm* to 3 *cm* in radius and as the size of the polyp increases it is more likely that it will become cancerous. Most polyps may not become cancerous, however ones that do, grow very quickly, invade, break through the colon wall and eventually spread to other parts of the body. Early detection and removal of colonic polyps improves the survival rate [1]. Computed Tomographic Colonography (CTC), also known as the Virtual Colonoscopy, is a minimally invasive technique that employs X-Ray CT imaging of the abdomen and pelvis following cleansing and air insufflation of the colon. Originally proposed in the early 1980's [2], it became practical in the early 1990's following the introduction of helical CT and advances in computer graphics [3]. Currently available multi-slice helical X-Ray CT scanners are capable of producing hundreds of high resolution ($< 1\text{ mm}$ cubic voxel) images in a single breath hold. Conventional examination of these source images is rather time-consuming and the detection accuracy is unavoidably limited by human factors such as attention span and eye fatigue. As such computer aided detection/diagnosis (CAD) tools have increasingly gained more importance.

1.1. Literature Review on CTC-CAD

Several visualization and navigation techniques have already been proposed to help the radiologists [4–7]. However, computer aided detection (CAD) tools are envisioned to improve the efficiency and the accuracy beyond what can be achieved by visualization techniques alone [9–20]. The CAD problem for CTC bears several challenges summarized in [8]: The difficulty of specifying the polyp location, the fact that there are several structures in the colon wall mimicing colonic polyps, the existence of artifacts and poor colonic distention are among these challenges.

The "global" shape of the polyp, i.e. the mushroom shape, is unique in the whole colon. The algorithms that are explained in this section try to capture the global shape information by various methods. Some of them uses local information to guess the whole shape, while others try to understand it by using extrinsic parameters of the surface.

Summers *et al.* used curvature characteristics of the surface, like mean curvature and defined a dimensionless sphericity ratio using principal curvatures to detect polyps in [9,10]. In [12], Jerebko *et al.* used the same features proposed in [9,10] and added the curvature characteristics of polyp neck to train a support vector machine (SVM) to classify polyp and nonpolyp regions. Jerebko *et al.* have also proposed to use seven geometric features for further identification after an initial detection step. They have used Canny edge detectors and the Radon transform to segment the boundary of initial polyp candidates, find the baseline of polyp candidates and calculate the geometric features like number of boundary pixels, polyp internal area and polyp height. Kiss *et al.* have used the curvature characteristics to find convex regions on the colon wall as an initial detection in [13]. The convex regions however may correspond to either colonic polyps or haustral folds. They have used surface normals and sphere fitting methods to further distinguish between folds and polyps. The surface normals method is based on the observation that the surface normals for a sphere would intersect at the center, on the other hand the normals would intersect on a line for a cylinder. In the sphere fitting method, they have fit spherical patches onto the convex regions since the shape of a colonic polyp would not resemble a whole sphere in most cases. Both of these methods rely on the sphericity assumption of colonic polyps.

In [14,15], Yoshida *et al.* have proposed two measures characterizing the shape, the shape index (SI), and the curvedness (CV) of the colon wall based on principal curvatures. Using these measures, they find the polyp candidates and cluster them using fuzzy c-means algorithm. They also calculate two additional measures for the initially detected polyps, the gradient concentration (GC) and the directional gradient concentration (DGC), which characterizes overall direction of the gradient vectors of CTC data around a point. Combining all these features, they identify polyps with

classifiers using linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). The surface normal overlap (SNO) algorithm, proposed in [16] is similar to the surface normals method used in [13]. Again the intersection of surface normals is used however several problems in the surface normals method have been solved. The length of the normals are kept finite to avoid the intersection of normals coming from other parts of the surface. They also used radially gaussian distributed cylinders instead of lines as surface normals to improve the robustness against small variations on the surface. In [19], Yao *et al.* have proposed to use deformable models to segment colonic polyps. The algorithm starts by finding curvature values for every point on the surface and classifying each surface point as flat, concave or convex. Then from each point in the tissue region close to the colon wall, evenly spaced rays are shot, and a score is given to these points depending on the curvature characteristics of the surface point the rays hit. Intensity adjustment based on this scoring is followed by fuzzy c-mean clustering. Three clusters are formed, air, polyp and nonpolyp. For each polyp cluster a deformable model, as proposed by Cohen [23], is initialized from the center of the class. The deformable model is first applied in 2D to segment the polyp in one slice and then extended to the neighboring slices for 3D segmentation of the polyp.

There is also some research focused on improving the specificity of CTC-CAD algorithms. The proposed methods under this category are post-processing approaches trying to reduce the number of false positives without sacrificing sensitivity of CAD algorithms. Among these post-processing approaches, Göktürk *et al.* in [17] also used SVMs as a secondary identification step. To extract features to be used in SVM, they proposed to use random orthogonal shape sections method. This method uses multiple triple cross sections (three orthogonal cross-sections at a time) of polyp candidates taken at random angles to produce a reliable shape signature. Acar *et al.* modeled the way radiologists use 3D information while they are examining a stack of image slices [18]. They use a predetection step to identify polyp candidates by using the SNO method, in which normals are limited to 5mm from the colon wall to the tissue. Then for each candidate, they create vector fields from the changes in the location of edges in the segmented CT image as one scrolls in each direction. The characteristics of the produced vector fields are used to identify colonic polyps.

The detection algorithm we have previously proposed in [20], Heat Diffusion Fields (HDF), was based on diffusion filters and to the best of our knowledge is the first algorithm that included a built-in polyp enhancing scheme. The algorithm is based on the observation that a heat flow from the lumen air towards the colon tissue would create heat sinks in the middle of hemispherical structures, like polyp centers. Using the curvature characteristics of polyp necks and nonlinear diffusion filters, we have implemented a heat flow initialized by a hot lumen air and cold colon tissue. We intended to use higher diffusion coefficients at polyp necks by making the diffusion coefficient a function of surface curvature. Thus, we have enhanced the spherical symmetry of colonic polyps. The heat flow is captured using optical flow equation and a vector field is formed based on it. The spherically symmetric sinks of the vector field yielded the initial polyp candidates. For each polyp candidate a geometrical measure is computed by shooting rays similar to the one computed in [19], capturing the information about characteristics of the colon wall surrounding it. Two level threshold is applied using the information coming from the vector field and from the ray shooting process to identify colonic polyps.

In all of these works, the performance of the CAD algorithm is evaluated by comparing the results of the algorithm with those of a radiologist. The details of the evaluation methodologies will be given later, however it is important to explain the performance metric used in CTC-CAD briefly at this point. The aim of any CTC-CAD algorithm is to identify all colonic polyps without any false positives. A false positive is a structure identified as a polyp by the CAD algorithm whereas it is not. The performance is measured by the number of false positives at a specified sensitivity level, which denotes the percentage of polyps found. The free response receiver characteristic (FROC) curves provide the required tools for such an assessment [21].

1.2. Problem Definition and Proposed Solution

Most of the CAD algorithms are, either implicitly or explicitly, based on the assumption that all colonic polyps are hemispherical protruding structures with a regular surface having the same geometric properties locally at every point. This does

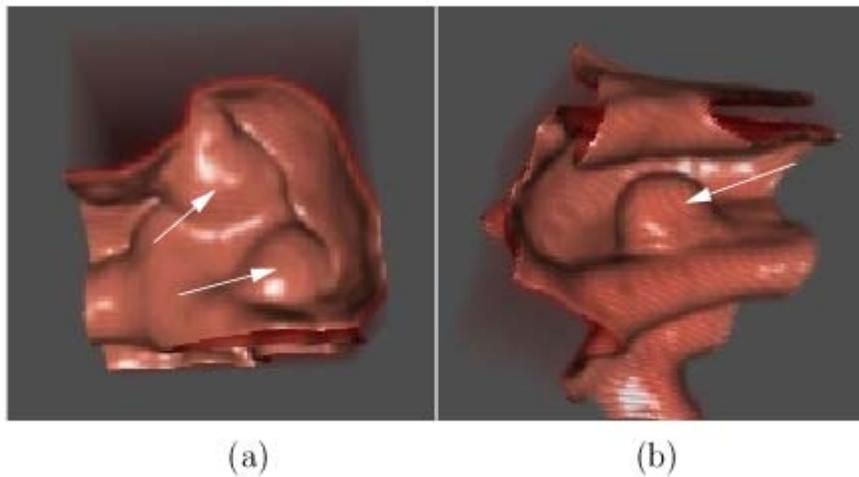


Figure 1.1. Types of polyps: (a) Sessile (b) Pedunculated

not hold in most cases though. Colonic polyps can be divided into two categories: sessile and pedunculated. A *sessile polyp* have a broad base of attachment and is not protruding from the surface much, resembling a big sphere mostly buried under the surface. A *pedunculated polyp* looks like a sphere attached to the end of a stalk with variable length. Examples of each type of polyp can be seen in Figure 1.1. Among these categories pedunculated polyps fit better to the assumption in CAD algorithms and they are relatively easier to detect. Sessile polyps on the other hand, are much harder to detect automatically since their spherical nature is not as visible as pedunculated polyps [8, 11]. This type of polyps decrease the performance of CAD algorithms by either decreasing the sensitivity level for a specific false positive number or increasing the false positive rate at a specific sensitivity level.

In this thesis, we propose the Polyp Enhancing Level Set methods (PELS) as a preprocessing step that would be applied prior to any CAD algorithm based on the assumption that a colonic polyp is a hemisphere residing on the colon wall. As such, it is designed to improve almost all of the previously proposed polyp detectors. This idea of enhancing colonic polyps was first proposed in the HDF based polyp detector in [20]. PELS is an interface evolution method built on the well known Level Set framework. The algorithm extracts and evolves the whole colon wall to increase the difference between the true colonic polyps and other structures that mimic polyps. In other words, the algorithm evolves sessile polyps towards pedunculated polyps, which

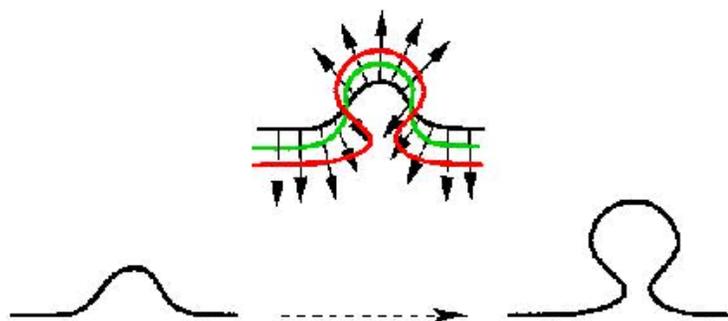


Figure 1.2. 2D illustration of the desired motion

are easier to detect for CAD algorithms. Figure 1.2 illustrates the desired motion of the colon wall around a polyp for a 2D representation.

The algorithm is based on the fact that colonic polyps have distinct "global" shapes, protrusions attached to the colon wall by a stem, as in the case of pedunculated polyps, or flat bumps, as in the case of sessile polyps. Geometric information extracted from local neighborhoods of points on any colonic polyp have specific characteristics, which however are not unique to polyps. In other words there are other structures that mimic polyps locally in the colon. Our method uses local curvature characteristics of the colon wall to evolve the "whole" colon wall such that the spherical symmetry of polyps increase and their attachment to the colon wall gets thinner. Other structures on the colon wall also change during this evolution, however due to their global geometric shapes they do not evolve into a polyp shape. The proposed algorithm can be divided into three different steps segmentation/preprocessing, propagation speed calculations and topology preserving level set evolution. We evaluated PELS in combination with the SNO algorithm [16].

Chapter 2 is devoted to explain briefly the deformable models, the level set theory, numerical implementations of level sets and the topology preserving level sets. In Chapter 3, we give details about the colon wall evolution. Segmentation of the colon wall, detailed derivations of speed functions and implementation issues are given in this chapter. Experiments with the real patient data, details of SNO algorithm and the discussions are given in Chapter 4. In Chapter 5, we conclude with speculations about future studies.

2. BACKGROUND ON LEVEL SET METHODS

The methods explained in this thesis are based on level set methods, which form a subclass of deformable models. In this chapter general information about deformable models, implicit functions, the mathematical theory behind and numerical implementations of level sets and topology preserving level sets are explained.

2.1. The Deformable Models

The motion of the colon wall, which is in more general terms an arbitrary interface, can be implemented by several different methods under the class of deformable models. Deformable models are curves or surfaces that move within 2D or 3D digital images under the influence of internal, external forces and user defined constraints. After Kass introduced them first in [22], many different models have been proposed. These models can be coarsely divided into two different classes, parametric deformable models, [23, 24], and geometric deformable models, [25–27].

The two subclasses of deformable models differ by the way the interface is represented and the way they are implemented. The parametric deformable models represent the interface explicitly as parameterized contours while the geometric deformable models represent the interface implicitly as a level set of a higher dimensional function. There are certain advantages of each model over the other one. The first advantage of the geometric models is that they are independent of the parametrization of the evolving contour and generally the contour is not parametrized until evolution is over. However parametric models depend on the parametrization used and furthermore special care must be paid to the discrete representation of this parameterized interface over the course of evolution. Secondly, the computation of intrinsic parameters of the interface is much easier and accurate in the case of geometric models. As for the parametric models the accuracy of the computation of these parameters depends on the accuracy of the parameterization used and the process is harder. One property of the geometric models that can be viewed as an advantage or disadvantage depending on the applica-

tion is that these models handle topological changes (merging or breaking up) of the evolving interface automatically. Hence topological changes cannot be avoided in the original geometric model formulation proposed in [25,26]. Parametric models are more advantageous when one needs to control topological changes. Another difference between parametric and geometric models is that the geometric models are implemented in a Eulerian framework while parametric models are implemented in a Lagrangian framework. The underlying coordinate system remains fixed in Eulerian framework. In a Lagrangian framework on the other hand, the range of the parametrization defines the motion of the interface hence the position of the points on the interface defines the coordinate system. A summary of this comparison is given in Table 2.1.

Table 2.1. Comparison of parametric and geometric deformable models

Parametric	Geometric
Depends on parametrization.	Independent of parametrization
The accuracy of computations of intrinsic parameters depends on the parametrization and they are generally harder.	Intrinsic parameter computations are easy and accurate.
User controls the topological changes.	Topological changes are handled automatically.
Uses Lagrangian framework.	Uses Eulerian framework.

Due to the difficulties in parameterizing an arbitrary surface and the ease in computing intrinsic surface parameters using implicit functions, we chose to use geometric deformable models to enhance the colon wall. The fact that geometric deformable models handle topological changes of the evolving interface automatically becomes a disadvantage in our application since we want to preserve the topology of the colon wall. This problem is solved by using Topology Preserving Level Sets as explained in [29]. More details about this issue will be given in Section 2.4.

2.2. Implicit Functions and the Signed Distance Function

In an explicit interface representation, we define the interface, $\partial\Omega$, with points that belong to the interface, either by a parameterization or just by listing the points [22]. In the case of implicit representation on the other hand, the interface is defined as an iso-level of a higher dimensional function (the implicit function). The iso-level is a 1D curve for 2D implicit functions and a 2D surface for 3D implicit functions. In order to be able to evolve an interface using the geometric deformable models, the interface should either be closed or not contained in the whole domain entirely. In other words, it divides the whole domain of definition of the implicit function, Φ , into two regions. For a closed interface these two regions can be considered as the interior and the exterior of the interface and for others a similar convention can be used. The only thing that an implicit function should do is to distinguish between these two regions so that the interface can be tracked. A very simple 2D example of an implicit function and implicitly represented unit circle is given with Equation 2.1.

$$\Phi(x, y) = x^2 + y^2 - 1 \quad (2.1)$$

where the $\Phi(x, y) = 0$ iso-level is the unit circle centered at the origin. The interior region is the open disk Ω^- and the exterior region is the open set Ω^+ . Equation 2.1 defines a function in \mathbb{R}^2 and bears the unit circle as its zero level set, which can be seen in Figure 2.1.

Generalizing the idea behind the example above, the $n - 1$ dimensional interface $\partial\Omega$ would be represented implicitly as a level set of a function $\Phi : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ as given in Equation 2.2.

$$\partial\Omega = \{\mathbf{r} | \Phi(\mathbf{r}) = a\} \quad (2.2)$$

where \mathbf{r} is the position vector in \mathbb{R}^n and a is a constant defining which iso-level of Φ , $\partial\Omega$ is. Then $\partial\Omega$ would be called the a -level set of Φ . The parameter a does not have an importance in the formulation, so for simplicity 0 is used for a . Hence $\partial\Omega$ becomes

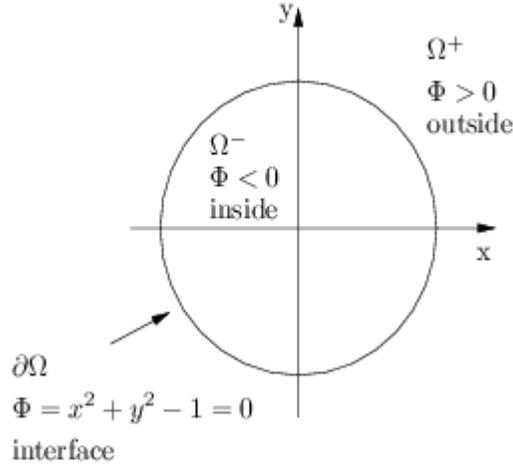


Figure 2.1. A circle implicitly represented by its level set function $\Phi(x, y)$

the zero level set of Φ .

When the underlying interface can be represented analytically, as the unit circle, the higher order function to keep the interface as its zero level set can be found by algebraic manipulations. If the underlying interface is complex and cannot be expressed analytically but through parametrizations then algebraic manipulations will not work. Construction of an implicit function for a general interface is an ill-posed problem since there is no unique solution. Amongst infinitely many implicit functions with identical zero level set, the signed distance function plays a very important role in the theory of level set methods. In order to define the signed distance function let us first define $d(\mathbf{r})$ as:

$$d(\mathbf{r}) = \min_{\mathbf{r}_I} (|\mathbf{r} - \mathbf{r}_I|) \quad \text{for all} \quad \mathbf{r}_I \in \partial\Omega \quad (2.3)$$

$d(\mathbf{r})$ represent the smallest distance to the interface from any point $\mathbf{r} \in \Omega$. Note that for all points in Ω , $|\nabla d| = 1$ [31]. Another advantage of this distance function is that it is a smooth function at every point except on the interface itself.

Using the function defined in Equation 2.3, the signed distance function is defined

as

$$\Phi(\mathbf{r}) = \begin{cases} -d(\mathbf{r}) & \text{if } \mathbf{r} \text{ is in the interior region} \\ d(\mathbf{r}) & \text{if } \mathbf{r} \text{ is in the exterior region} \end{cases} \quad (2.4)$$

We, conventionally, define the signed distance function to be positive in the exterior of the interface, negative in the interior and zero on the interface. Clearly if the point is on the interface then the signed distance function becomes zero. It follows from the properties of $d(\mathbf{r})$ and the definition of $\Phi(\vec{x})$ that

- $|\nabla\Phi(\mathbf{r})| = 1 \ \forall \mathbf{r} \in \Omega$.
- $\nabla\Phi$ is normal to iso-levels.
- $\Phi(\mathbf{r})$ is monotonic and generally differentiable at the zero level set where as $d(\mathbf{r})$ has a cusp there. (The differentiability depends on the initial interface.)

Although the problem of creating an arbitrary implicit function for an arbitrary interface is an ill-posed problem we can cure this by changing the question to creating the signed distance function of the interface. Since the signed distance function is unique, this new problem has a unique solution. The numerical construction of the signed distance function is done by the *fast marching methods*, which is implemented via solving a boundary value PDE. Further details of this method can be found in [30] and in [31]. An example of a 2D signed distance function for a general curve is given in Figure 2.2. Figure 2.3 shows the gradient vectors of the signed distance function Φ on the curve. We see that the vectors are normal to the curves.

2.3. Level Set Methods

The geometric deformable models are implemented using the level set methods proposed in [32]. This section summarizes the mathematical theory behind the level set methods and numerical implementations of these models.

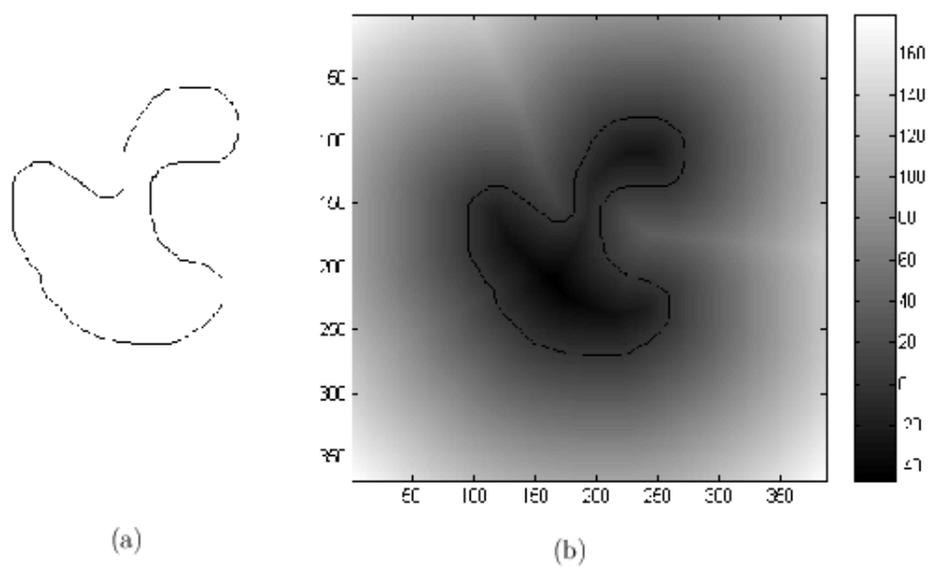


Figure 2.2. Implicit function example: (a) The original interface (b) The signed distance function

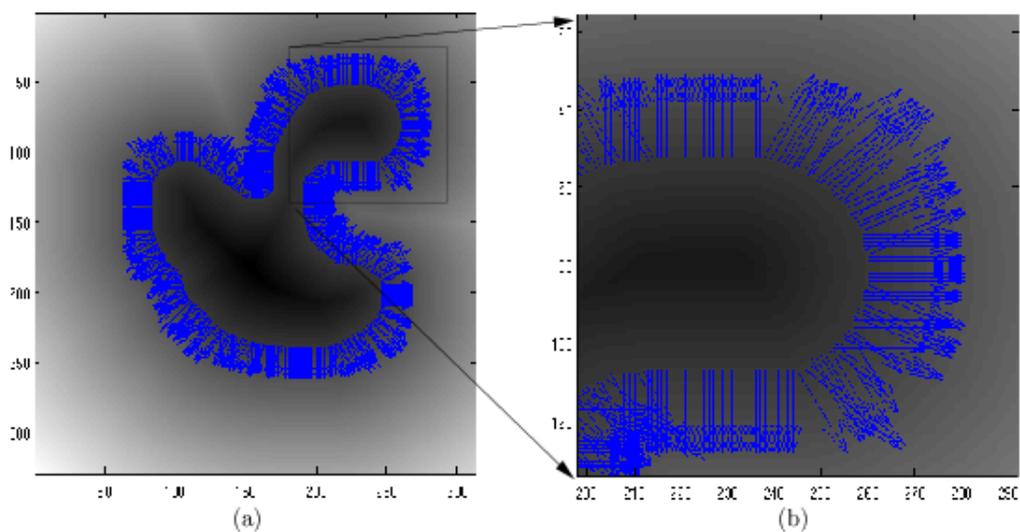


Figure 2.3. Gradient vectors of the signed distance function are normal to the interface

2.3.1. The Evolution Equation

The level set method is based on the manipulations of the level set function, Φ , in order to move the interface, $\partial\Omega$, which will be called Γ in this section. Hence the function Φ is a time varying scalar function in \mathbb{R}^n and can be written as $\Phi(\mathbf{r}, t)$, where \mathbf{r} is the position vector and t is the time variable. As explained in Section 2.2, the initial interface is embedded in Φ as its zero level set. To construct the link between the motion of the interface and the level set function, we assume that $F(\mathbf{r}, t)$ is the speed function acting on the interface in the normal direction. The level set equation arises from the fact that as time evolves the interface should remain at its zero level set, which is given by:

$$\Phi(\Gamma(t), t) = 0 \quad (2.5)$$

Differentiating both sides of Equation 2.5 with respect to t and observing that $\Gamma' \cdot \vec{n} = \Gamma' \cdot \frac{\nabla\Phi}{|\nabla\Phi|} = F(\mathbf{r}, t)$ we obtain the level set equation Osher and Sethian proposed in [32]:

$$\Phi_t + F(\mathbf{r}, t)|\nabla\Phi| = 0, \Phi(\Gamma, t = 0) = 0 \quad (2.6)$$

This equation is an initial value problem and the initial condition is the implicit function constructed from the interface. It only gives the motion in the normal direction defined by F . Other works on level sets, like [26], have improved this formulation by adding several other components. The general form of the level set equation as used in the literature currently is as follows,

$$\Phi_t = F_{prop}|\nabla(\Phi)| + F_{curv}|\nabla(\Phi)| + \vec{F}_{adv} \cdot \nabla(\Phi) \quad (2.7)$$

The speed function F in Equation 2.6 is divided into three different parts. These parts account for different effects on Φ and therefore the interface Γ . The effect of these components on the interface are:

- The first component $F_{prop}|\nabla(\Phi)|$ provides the motion in the normal direction to

the interface. Mathematically, this motion is given by changing Φ in time relative to the norm of the gradient vector at any point multiplied with a scalar function, $F_{prop}(t, \mathbf{r}, \Phi, \nabla\Phi)$. This term is called the propagation speed.

- The second component $F_{curv}|\nabla(\Phi)|$ is simply the curvature flow, which has a smoothing effect on the interface under motion. F_{curv} is proportional to the curvature of the interface (mean curvature in 3D case), in other words $F_{curv} = g(\mathbf{r})\kappa$, where $g(\mathbf{r})$ is a spatially varying function and κ is the curvature. This term makes sense when curvature can be defined in the space the interface is living in, as a curve in 2D and a surface in 3D.
- $\vec{F}_{adv} \cdot \nabla(\Phi)$ gives us the effect of the external vector field \vec{F}_{adv} on the motion of the interface and is called the drift velocity. This term is similar to the first one however it is a vectoral term.

The level set theory is built to move an interface under the effect of the speed function F . It defines the motion of the interface. We can impose our aim, enhancing colonic polyps, to the level set formulation through F . Our research is concentrated on finding a speed function that would enhance colonic polyps. The creation of the speed function, the function itself and the motion defined by the speed function will be explained in Section 3.2.

In almost all examples in the literature the speed terms are created considering the interface alone. However, these terms are not only defined for the points on the interface, they are defined on the whole domain of the implicit function. This is where the Eulerian framework of level set methods differs from parametric deformable models where the speed values are assigned only to the points on the interface. The level set equation acts on the value of the implicit function Φ at every point, hence even though we create speed terms considering the interface alone we should assign suitable speed values to all points in the domain of Φ . Once speed values for points corresponding to the interface are assigned, there are infinite ways to assign speed values to all other points, as in the case of implicit function construction. While using the signed distance function as the implicit function, it is preferable to preserve its nice properties such as $|\nabla\Phi| = 1$ for all time. In [30], methods of speed extension to the whole domain, which

keeps the implicit function as signed distance function for all time values are explained.

2.3.2. Numerical Implementation

In a domain where both time and space are discrete we have to use numerical methods to solve Equation 2.7. The discretization of the left hand side and the right hand side of Equation 2.7 are handled differently. The time derivative in the right hand side is computed by a very simple first order accurate forward differencing method. For a given time step Δt the level set equation is

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} = F_{prop}|\nabla(\Phi^n)| + F_{curv}|\nabla(\Phi^n)| + \vec{F}_{adv} \cdot \nabla(\Phi^n) \quad (2.8)$$

where $n \in \mathbb{N}$ is the time index. To find the solution of the level set equation at a specific time $t = T_0 = n_0 * \Delta t$, we solve the Equation 2.8 iteratively n_0 times starting from the initial $\Phi|_{t=0}$. In practice however, rather than finding the solution of the level set equation at a specific time, we are interested in the steady-state solution the equation will converge to. Hence we would iterate the solution until the right hand side of Equation 2.8 would be smaller than some pre-specified small value ε .

The left hand side of Equation 2.8 contains spatial derivatives and the discretization of these derivatives require more attention. In Section 2.3.1 we have seen that we can separate the right hand side of the level set equation into three parts having different effects on the motion of the interface. In terms of numerical methods and partial differential equations, the right hand side can be separated into two different components.

$$\Phi_t = \underbrace{F_{prop}|\nabla(\Phi)| + \vec{F}_{adv} \cdot \nabla(\Phi)}_{-\bar{H}(\mathbf{r}, \Phi, \nabla\Phi)} + g(\mathbf{r})\kappa|\nabla\Phi| \quad (2.9)$$

The first component shown in Equation 2.9 is a combination of the first and the third terms of Equation 2.7. If the signed distance function is used as the implicit function then the second term in Equation 2.9 is equivalent to a nonlinear diffusion term. Hence

we can rewrite Equation 2.9 in two parts as,

$$\Phi_t^I = -\overline{H}(\mathbf{r}, \Phi, \nabla\Phi) \quad (2.10)$$

$$\Phi_t^{II} = g(\mathbf{r})\Delta\Phi \quad (2.11)$$

$$\Phi_t = \Phi_t^I + \Phi_t^{II}$$

Equation 2.10 is in the form of a Hamilton-Jacobi equation and Equation 2.11 is a nonlinear diffusion equation. The numerical schemes for these two parts are different because the diffusion part does not develop discontinuities during the evolution [30,31] while Hamilton-Jacobi equations can develop discontinuities. Thus discretization of these two parts are handled separately.

Diffusion equations are studied extensively in the computer vision and image processing literature under the topic of diffusion filters [34]. Weickert *et al.* have proposed efficient and reliable methods for solving nonlinear diffusion equations in [35]. Central differencing is commonly used to approximate spatial differentiation in these techniques. The discretization of the diffusion term in the level set formulation is also done using the central differencing.

Equation 2.10, which is in the form of a Hamilton-Jacobi equation, is more interesting. The interface evolution governed by these types of equations may develop discontinuities like corners in finite time, even if the flow is very simple. Assume that the initial Φ is a smooth corner and that we evolve in the normal direction with constant speed. The following equation is the equation for this flow with constant speed A and Figure 2.4 shows the evolution of such an initial curve.

$$\Phi_t = A|\nabla\Phi| \quad (2.12)$$

At a finite time a corner develops, and at that point the normal direction is not defined and the flow becomes ambiguous. To continue the flow after this point, the level set equation should be solved numerically for a weak solution. A weak solution of

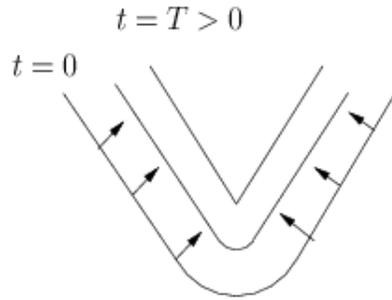
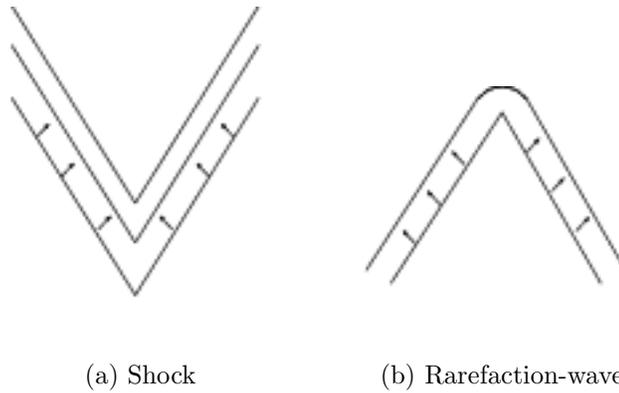


Figure 2.4. Smooth corner moving in its normal direction



(a) Shock

(b) Rarefaction-wave

Figure 2.5. Entropy satisfying weak solutions for 1D interfaces embedded in 2D functions

a differential equation satisfies the integral formulation of the PDE. The integral form does not require the same degree of differentiability as the differential form, hence may allow discontinuous solutions. The problem with solving for a weak solution is that even if the differential equation has a unique solution for the given initial and boundary conditions the weak solution is not unique. So the question arises: Which weak solution should we solve for? The answer comes from fluid dynamics and physical systems. It says that Equation 2.10 should be solved for the physically meaningful weak solution, which Sethian called "the entropy satisfying" weak solution [30]. Once a discontinuity is formed, the entropy satisfying weak solution takes one of the two forms shown in Figure 2.5 depending on the direction of the motion.

The vectors in Table 2.5 shows the direction of motion. In the case of shocks, as the interface moves corners are developed. The entropy satisfying solution preserves the shock as the interface evolves. In the case of a rarefaction wave a discontinuity

is to be separated and the two end points of the discontinuity are combined with a smooth curve. The numerical scheme applied to solve Equation 2.10 should be able to evolve the interface towards the entropy satisfying weak solution so that we can have discontinuous, physically meaningful solutions. Sethian and Fedkiw give details about such schemes in their books [30, 31]. These numerical schemes are motivated from the methods proposed to solve hyperbolic conservation laws for which the details can be found in [33]. The general class of these methods are called the *upwinding schemes*. These schemes are based on the *method of characteristics* that is used to solve partial differential equations [36]. The method of characteristics tells us which points in the domain of Φ influences the value of Φ at point \mathbf{r} . In other words, the method tells us the direction of information flow at point \mathbf{r} . The upwinding schemes use this information flow direction to update the value of Φ at every point in the domain. One of the methods in the class of upwinding schemes, which is commonly used in level set applications, is the Godunov's method [33]. The numerical implementation of Equation 2.10 is done by this method. More details about the consistency and the stability of numerical methods using upwinding schemes and forward differencing can be found in [33].

2.4. Topology Preserving Level Sets

The PELS algorithm evolves the colon wall so as to enhance the geometric structures of colonic polyps. One thing to be careful about in this process is the topological structure of the colon wall. Creating new holes, merging the colon wall patches or breaking up the colon wall into two or more parts at some location during the evolution would change the topological characteristics of structures around that location. These topological changes might destroy existing polyps and/or introduce additional structures on the colon wall.

As explained in Section 2.1 geometric deformable models handle topological changes automatically. While we are moving the interface via solving the level set equation with respect to the implicit function, we do not have direct access to the interface embedded. With parametric models, one always works on the interface thus,

one can easily detect topological changes and explicitly prevent them. In the geometric deformable models, the topological changes can be detected in the level set formulation with some modifications on the algorithm. Han *et al.* in [29] have proposed a modified level set model, which will be briefly explained in this section, that would detect the topological changes as they occur and prevent them.

The topology of an interface is the relation between two different regions, which are separated by the interface. In our case, the colon wall is the boundary separating the colon tissue and the lumen air. During the evolution, the interface (the zero level set) moves when the implicit function changes sign at some point. Thus the topological changes are related to such sign changes. Some of these sign changes do not alter the topology of the interface depending on the point the sign change is happening at. Such points are called *simple* points. Points for which the sign change of the implicit function at that point changes the topology are *non-simple* points and detecting these points before changing the sign enables us to preserve topology of the interface [29]. To detect these points we need to define concepts like neighborhoods and topological numbers for digital topology [37].

2.4.1. Digital Topology

Before we begin defining a simple point, we must define neighborhoods in a cubic grid. The neighborhood relation in digital domain is defined in terms of connectivity. In a 2-dimensional domain we have two different connectivity types, 4-connected and 8-connected. In 3-dimensions we have three different connectivity types and these can be seen in Figure 2.6.

Topology of an interface is defined in terms of the relation between regions, which are separated by the interface. We can call one of these regions “inside” ($\Omega^- = \{\mathbf{r} \in \Omega | \Phi(\mathbf{r}) < 0\}$) and the other one “outside” ($\Omega^+ = \{\mathbf{r} \in \Omega | \Phi(\mathbf{r}) > 0\}$) with respect to the interface. Using this convention we can define the following sets of points connected to a given point. These sets will be useful in classifying between simple points and non-simple points. For an arbitrary point \mathbf{r} ,

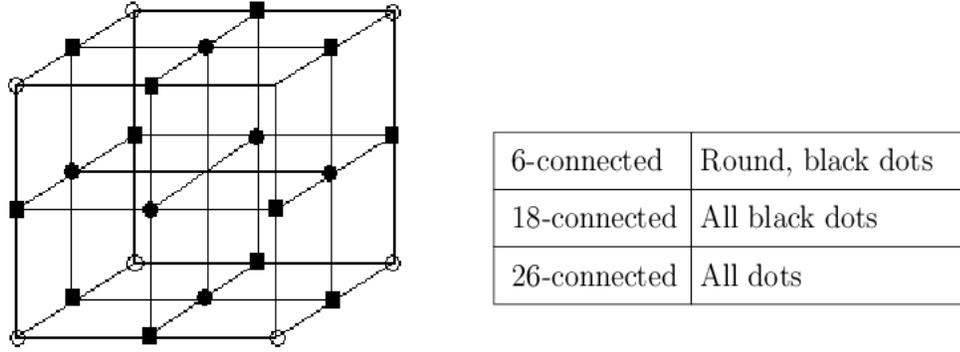


Figure 2.6. Types of neighborhoods for the point in the middle of the $3 \times 3 \times 3$ cubic grid in 3D digital domain

- $N_m(\mathbf{r})$: The set of points m -connected to \mathbf{r} , including \mathbf{r} .
- $N_m^*(\mathbf{r})$: The set of points m -connected to \mathbf{r} , excluding \mathbf{r} .
- X : $N_{26}(\mathbf{r}) \cap \Omega^-$
- \bar{X} : $N_{26}(\mathbf{r}) \cap \Omega^+$
- $C_n(X)$: The set of all n -connected components in X .
- $C_n^a(\mathbf{r}, X)$: The set of all n -connected components of X n -adjacent to the point \mathbf{r} .

A n -connected component in the last two definitions resembles a set for which any two elements can be connected with a n -connected path.

Observe that not all points in a region are topologically important. The significant points are the ones that are on the boundary between two regions. In other words, all points 26-connected to a boundary point are not in the same region as the boundary point. Such points on the boundary may move from one region to the other one during the evolution and cause topology to change.

In Figure 2.7 white points are in Ω^- and black points are in Ω^+ and the point in the middle is on the boundary. This point has a topological significance because the location of the surface depends on its membership. To detect topological changes in the surface we need to check the boundary points. If the boundary point is a simple point then topologically the surface remains equivalent when that point moves from one region to the other, and if the point is non-simple then topologically the surface

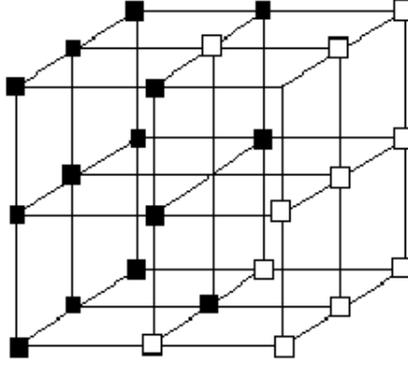


Figure 2.7. An example of a topologically significant point

will change.

The topological numbers are determined based on simple relations between the connectivity of boundary points on both sides of the interface. These relations will help us classify boundary points into simple and non-simple points. The topological numbers of a point x relative to the set X are defined in [37] as,

$$T_6(\mathbf{r}, X) = \#C_6^a[\mathbf{r}, N_{18}^*(\mathbf{r}) \cap X]$$

$$T_{6+}(\mathbf{r}, X) = \#C_6^a[\mathbf{r}, N_{26}^*(\mathbf{r}) \cap X]$$

$$T_{26}(\mathbf{r}, X) = \#C_{26}^a[\mathbf{r}, N_{26}^*(\mathbf{r}) \cap X]$$

$$T_{18}(\mathbf{r}, X) = \#C_{18}^a[\mathbf{r}, N_{18}^*(\mathbf{r}) \cap X]$$

Similarly these values could be defined for any set, for example for \overline{X} . In simple terms, the first quantity represents the number of 6-connected components in the 18-neighborhood of a point \mathbf{r} that are in set X (inside the interface) and the others follow similarly. For a boundary point we need to use a conjugate pair of topological numbers of it (one relative to X and the other relative to \overline{X}) to understand whether it is a simple point or not. In [37] four different conjugate pairs are given (6,26), (26,6), (6+, 18) and (18,6+). Any one of these pairs may be used, we have used (6,26) pair because of the ease in implementation. Using this conjugate pair a boundary point is simple if $T_6(\mathbf{r}, X) = T_{26}(\mathbf{r}, \overline{X}) = 1$ and non-simple otherwise.

2.4.2. Topology Preserving Level Set Algorithm

In the level set implementation the topology preservation is accomplished by restricting the evolution of the interface using the ideas of simple points. In our case, as explained in Section 2.2, the colon tissue is set to have positive values and the lumen air to have negative values. So a point on the boundary has both negative and positive valued points in its 26-neighborhood. During the evolution, we should detect such points and check whether they are changing their sign during iteration or not. For an arbitrary point the level set algorithm is modified as follows:

- Update the current value at the point using the level set method explained in Section 2.3.
- If the sign of the function at that point changes, check whether the point is a simple point or not.
- If the point is a simple point update the value as suggested by the original algorithm.
- If the point is a non-simple point then update the value at this point as:

$$\Phi_{n+1}(x) = \text{sign}(\Phi_n(\mathbf{r}))\varepsilon \quad (2.13)$$

where n is the iteration number and ε is a small positive number.

With this modification in the evolution we prevent non-simple points to move from one region to the other. Only the simple points are allowed to change region. Hence the topology of the interface is preserved.

3. COLON WALL DEFORMATION

In this chapter we are going to give the overview of the algorithm used to deform colon wall. Then we are going to explain each step in detail and give implementation issues at the end.

Two interface propagation flows are proposed for PELS. For both flows the algorithm starts with the segmentation step. The output of the segmentation step is subvolumes of size $39 \times 39 \times 39 \text{ mm}^3$ containing parts of the colon wall embedded in an implicit function Φ which is a signed distance function. Our sign convention assigns positive distance values to the points on the colon tissue and negative values to the points in the lumen air.

After creating the implicit function, we compute the principal curvatures and principal curvature direction, which will be explained in Section 3.2.1.1. These features will be later used in calculating the speed terms. In the level set formulation we have seen that there were three different terms, propagation term, curvature driven term and the drift velocity. For both of the flows defined in this thesis, we will only use the propagation term. Using the principal curvatures, we compute propagation terms, F_{prop} . The details of these computations will be given in Sections 3.2.1 and 3.2.2. Then we evolve the colon wall using level set methods explained in detail in Section 2.3. The flowchart of the overall algorithm is given in Figure 3.1.

3.1. Segmentation and Preprocessing

The first step in enhancing colonic polyps is the segmentation of the colon wall in CT images. The segmentation is done in two different steps. The initial segmentation step is based on the algorithm explained in [16]. This step separates the colon lumen from the rest of the body. The density of air is less than water and tissue, thus the Hounsfield Unit (HU) in the CT images is significantly lower for air regions. The air regions in the body are separated from the tissue by simple thresholding at 350 HU.

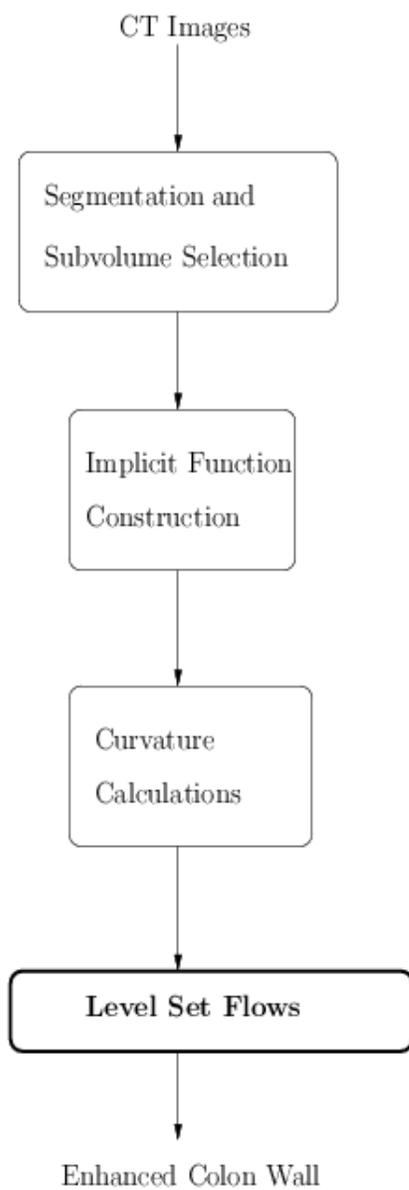


Figure 3.1. The PELS flow chart

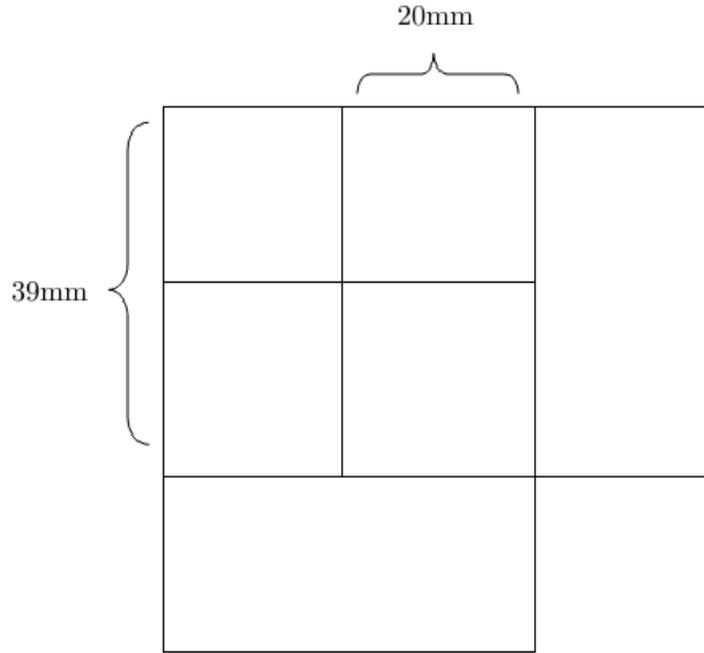


Figure 3.2. Overlapping method

However in CTC data, the colon lumen is not the only air region, the inferior parts of lungs are also air filled in the images. The air in the lungs has to be segmented and excluded. 3D region growing is used with an initial seed in an air region with a width or depth of greater than 60 mm in the most superior axial slice to segment out lungs. Once lungs are found and excluded, we are left with the colon lumen. We mark the tissue points in contact with the colon lumen and hence obtain a point cloud consisting of voxels on the colon wall.

Using the point cloud found in the initial segmentation step we form the colon wall. However, the whole colon wall is too big to process at once. We wish to run the enhancing algorithm on smaller subvolumes containing patches of the colon wall. Prior to choosing the center points of subvolumes, the colon wall is dilated towards the tissue using a 10 mm cubical structural element to guarantee the coverage of colonic polyps of size 20 mm or less (whose center will be 10 mm away from the colon wall). The center points for the subvolumes are chosen from the dilated colon wall. The chosen subvolumes are of size $39 \times 39 \times 39\text{ mm}^3$ and the centers are chosen such that the subvolumes overlap by 20 mm in the manner shown in Figure 3.2.

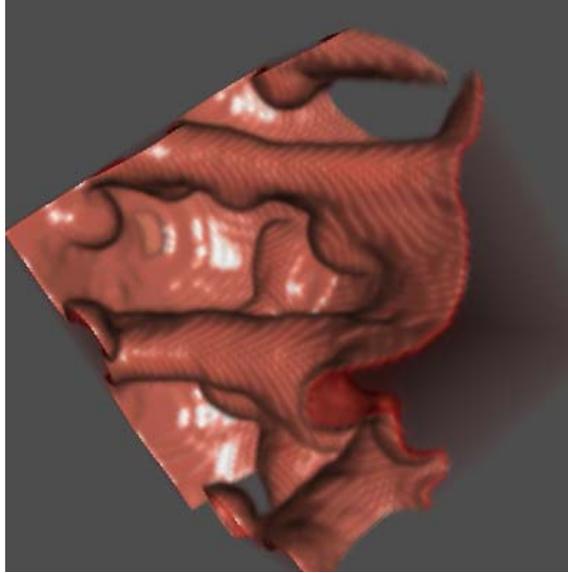


Figure 3.3. Segmented colon wall example with the red fog representing the colon tissue

Using this overlapping method we can be sure that we are not leaving any part of the colon wall out since the maximum distance between the set of subvolume centers and set of points on the colon wall is less than 20 mm . Moreover the amount of overlap guarantees that each polyp of diameter less than 20 mm is contained in at least one subvolume and most polyps are contained in more than one subvolume.

Some subvolumes selected may contain no parts of the colon wall due to dilation. After removing these ones the colon wall is segmented once more for each subvolume. All subvolumes are around the colon wall thus we can confidently assume that the air regions in each subvolume are part of the colon lumen. To segment the colon wall we first subtract 350 HU from the whole subvolume. Then we find voxels closest to the zero crossings by detecting sign differences among neighbors using 6-connectivity. Collection of these points constitute the segmented colon wall in the current subvolume. An example of a final segmented colon wall patch contained in a subvolume is given in Figure 3.3.

Next the colon wall is embedded in an implicit function in order to be used in the level set formulation. The colon wall patches contained in subvolumes do not have to be closed surfaces, and most of the time they are not. Hence, the notion of interior and

exterior regions, as explained in Section 2.2, is not directly applicable. However, the whole surface is not contained in a single subvolume and the colon wall divides each subvolume into two regions. The distinction between these regions is rather physical, the tissue and the lumen. We embed the colon wall patch in each subvolume in a signed distance function Φ constructed by the fast marching method. As explained in Section 2.2 the sign of the signed distance function is used to distinguish between different regions. We construct Φ such that voxels in the colon lumen will have negative distance values and voxels in the colon tissue will have positive distance values. Naturally voxels on the colon wall itself will have the value zero. As a result for each subvolume we obtain an implicit function Φ which bears the surface patch contained in the subvolume.

3.2. Polyp Enhancing Level Set Flows

The speed function in the level set formulation is the most important part of the theory from our view point. As explained in Section 2.3, the speed function defines the motion of the interface, in our case the colon wall. Creating a speed function is therefore equivalent to defining a flow. In this chapter, we are going to propose two different flows, attacking the problem of colonic polyp enhancement from two different views. The first flow is called the Joint Curvature Flow (JCF) and is created using a geometric approach. For the second flow, we have taken a variational approach and derived the Volume Maximizing Constant Surface Flow (VMCSF). These flows are explained in Sections 3.2.1 and 3.2.2 respectively.

3.2.1. Joint Curvature Flow

The joint curvature flow (JCF) is based on geometric observations on the colon wall and the polyp region, thus it is a pure geometric approach. Figure 3.4 shows a phantom polyp and a phantom fold. Observe that the polyp resembles an ideal sphere partially buried under the colon wall. In order to evolve it towards a pedunculated shape (ideally a mushroom shape) we have to pull the sphere off the colon wall. The illustration of this motion is given in Figure 1.2. The approach we took to achieve this, is to dig polyp necks towards the colon tissue while pulling polyp apices towards

the lumen air. Thus, the flow that will enhance the polyp should be sensitive to the neck of the polyp and the polyp apex. Moreover the polyp neck should sink, the polyp apex should rise and all other regions should stay unchanged. As the flow evolves the surface, polyp will become apparent on the colon wall and take an ideal mushroom shape.

The intensity information will not be useful to distinguish between the polyp neck and all other regions, since the HU values of the polyp and the rest of the colon are the same. Instead the specific geometric characteristics of these regions will become useful in this task. These characteristics can be captured by curvature characteristics of the surface around those regions. Distinguishing between the polyp neck and the other regions is in fact the main aim of any CAD algorithm since that would tell us the location of the polyp. This, on the other hand, cannot be done perfectly because although "global" shape of a polyp is unique we are only able to look at local neighborhoods around points on the surface to understand if it can be a part of a neck, an apex or neither. The local information can be deceiving since there may be neighborhoods not on the polyp neck having very similar local shapes. Therefore we will only be able to distinguish between local neighborhoods having similar geometric characteristics as

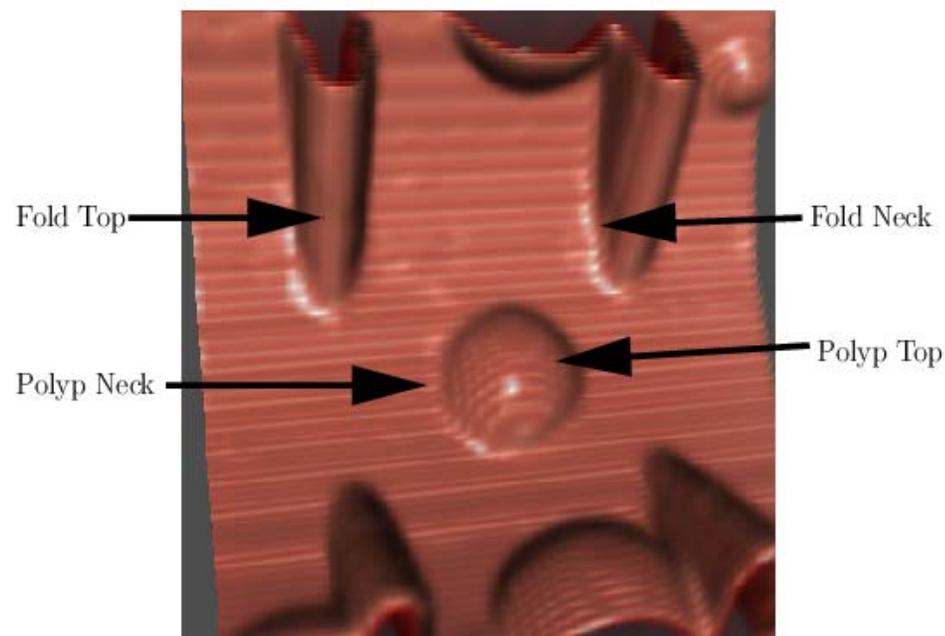


Figure 3.4. Phantom colon wall illustrating a polyp and a fold

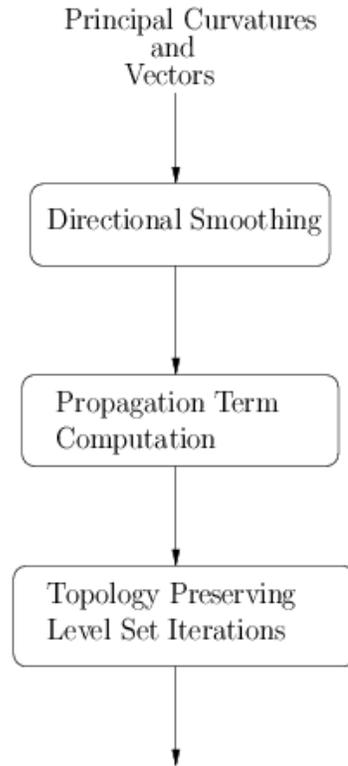


Figure 3.5. The flow chart for the JCF

polyp apex, neck, or other structures.

This flow consists of computing curvature maps, anisotropic smoothing of these curvature maps, the speed computation and evolving the colon wall using the level set formulation. Figure 3.5 shows the flowchart of the Joint Curvature Flow summarizing the relations between these. We will explain each block of the flowchart in detail in this section.

3.2.1.1. Curvature Calculations. In JCF, we are going to use the curvature characteristics to capture the geometric characteristics of local neighborhoods on the colon wall. As explained in Section 1.1 curvature characteristics of the colon wall have been used by many researchers in the detection process. Unlike previous works, we do not build our speed function on the mean curvature or the gaussian curvature. We used principal curvatures and principal vectors in creating the desired flow because of the geometrical insight they provide. The details for the theory of surfaces, curvatures and equations to compute principal curvatures and vectors using an arbitrary parametriza-

tion can be found in any introductory textbook on differential geometry, such as [38]. However, we use geometric deformable models and embed the colon wall in a higher dimensional implicit function. Thus, to compute the principal curvatures and vectors, we use the implicit function, not a parametrization. In [39], Monga and Benayoun derived the equations to compute several intrinsic and extrinsic parameters of a surface using the partial derivatives of the implicit function, in which that surface is embedded in.

Consider the surface Γ defined as an iso-level of an implicit function Φ . Let \vec{t} be a unit vector in the tangent plane of Γ at a point p , and \vec{N}_p be the normal vector of the surface at p . The Hessian matrix \mathbf{H} of the implicit function is given by,

$$\mathbf{H} = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} & \Phi_{xz} \\ \Phi_{xy} & \Phi_{yy} & \Phi_{yz} \\ \Phi_{xz} & \Phi_{yz} & \Phi_{zz} \end{bmatrix} \quad (3.1)$$

where the subscripts denotes partial differentiation in the respective dimension. Using \mathbf{H} , the curvature of Γ at the point p in the direction of \vec{t} is given by:

$$k_{\vec{t}} = -\frac{\vec{t}^T \mathbf{H} \vec{t}}{\|\vec{N}_p\|} \quad (3.2)$$

The principal vectors are the unit vectors in the tangent plane of Γ for which the directional curvature attains an extremum. The extremum values attained are called principal curvatures. In order to find the principal curvatures and the principal directions we have to search for the direction \vec{t} for which $k_{\vec{t}}$ is an extremum. This optimization problem, when solved by simple calculus, leads to arctangent computations. In order to avoid this computation, Monga and Benayoun used Lagrange multipliers technique

and obtained simple equations:

$$\kappa_i = \frac{h^T \mathbf{H} h + f^T \mathbf{H} f \pm \sqrt{(h^T \mathbf{H} h - f^T \mathbf{H} f)^2 + 4(h^T \mathbf{H} f)^2}}{2 \|\vec{N}_p\|} \quad (3.3)$$

$$\vec{t}_i = \begin{pmatrix} h_1 + f_1 \frac{K_i \|\vec{N}_p\| - h^T \mathbf{H} h}{f^T \mathbf{H} h} \\ h_2 + f_2 \frac{K_i \|\vec{N}_p\| - h^T \mathbf{H} h}{f^T \mathbf{H} h} \\ h_3 + f_3 \frac{K_i \|\vec{N}_p\| - h^T \mathbf{H} h}{f^T \mathbf{H} h} \end{pmatrix} \quad \text{with } i=1,2$$

where κ_i and \vec{t}_i are the i th principal curvature and principal vector respectively. In addition, \vec{h} and \vec{f} are defined as:

$$\begin{pmatrix} \frac{\vec{N}_p}{\|\vec{N}_p\|} & \vec{h} & \vec{f} \end{pmatrix} = \begin{pmatrix} \frac{\Phi_x}{\delta} & \frac{\Phi_y}{\gamma} & \frac{\Phi_x \Phi_z}{\gamma \delta} \\ \frac{\Phi_y}{\delta} & -\frac{\Phi_x}{\gamma} & \frac{\Phi_y \Phi_z}{\gamma \delta} \\ \frac{\Phi_z}{\delta} & 0 & -\frac{\gamma}{\delta} \end{pmatrix} \quad (3.4)$$

$$\gamma = \sqrt{\Phi_x^2 + \Phi_y^2} \quad \text{and} \quad \delta = \sqrt{\Phi_x^2 + \Phi_y^2 + \Phi_z^2}$$

The two principal curvatures correspond to the two extrema of Equation 3.2. For every smooth surface, Equation 3.2 has a minimum and a maximum, moreover the direction giving the minimum is orthogonal to the direction giving the maximum. This proof can be found in [38]. As can be seen in Equation 3.3, the second principal curvature is numerically less than the first principal curvature ($\kappa_1 > \kappa_2$), thus it is the minimum.

These computations are not only valid for the zero level set (the colon wall) but for any level set in the domain of Φ . They are valid for all iso-surfaces of Φ and computed for all the points in the domain of Φ . Using these computations, we compute four functions (maps): $\kappa_{1,2} : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $\vec{t}_{1,2} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. These maps will be used to construct the speed function for the desired flow.

3.2.1.2. Directional Smoothing. The computed principal curvature maps suffer from two different problems, due to discrete differentiation and noise. The first problem is discontinuities seen in these maps. The second problem is that the regions that should have homogeneous curvature characteristics are broken and interrupted by discontinu-

ities. For example, although all the points on a polyp neck should have similar principal curvature values, they differ in reality.

The discontinuities in curvature maps can be overcome with a suitable smoothing operation. On the other hand, isotropic smoothing operators without directional preferences, like Gaussian filtering, mean filtering and so on, would distribute the speed values through all neighborhood points equally. So they will not be useful in solving the second problem mentioned. We use non-symmetric Gaussian filters, which are very similar to anisotropic diffusion [34] to connect disconnected regions with similar curvature characteristics. In order to complete the disconnected "iso-curvature" regions, we will define the kernel of the Gaussian operator using principal vectors at every point, which form an orthonormal basis for the tangent plane of the colon wall. Although completing isocurvature regions at every point on the surface is important, our primary aim is to complete the ones on polyp necks. The observations on the real data have shown that on the neck of a colonic polyp, one of the principal vectors is parallel to the direction of the neck while the other one is perpendicular to this direction pointing towards the polyp apex. Due to our sign assignment in the construction of the implicit function Φ , explained in Section 2.2, the principal vector pointing in the neck direction is the second one, \vec{t}_2 . On the other hand a polyp apex ideally consists of umbilical points (an umbilical point is a point for which $\kappa_1 = \kappa_2$) thus the principal curvature directions do not have a specific meaning as in the case of the polyp neck. Figure 3.6 illustrates this on a phantom polyp.

The multidimensional Gaussian filtering, on the other hand, is totally characterized by the covariance matrix of the corresponding filter kernel. The eigenvalues of this matrix determine the amount of smoothing done in respective eigenvector directions. In order to create a Gaussian filter with the desired amount of smoothing in the desired direction we have to formulate the covariance matrix. Remembering that the covariance matrix is a positive definite matrix we can write it as follows,

$$\Sigma = VDV^{-1} \tag{3.5}$$

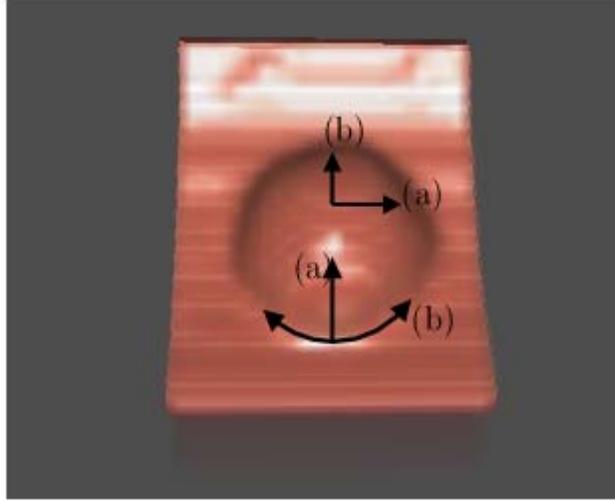


Figure 3.6. The principal curvature directions: (a) The first principal vector (b) The second principal vector

where Σ is the covariance matrix, V is the matrix of eigenvectors and D is the diagonal matrix of eigenvalues. We have mentioned that the principal curvature directions are orthogonal to each other and they form a basis for the tangent plane of the surface, which is also orthogonal to the normal vector of the surface. Thus, the eigenvectors of Σ form an orthonormal basis for \mathbb{R}^3 and the matrix V becomes orthonormal. In this case the equality $\Sigma = VDVT^T$ holds. By changing the n th entry in the matrix D we can set the amount of smoothing in the desired direction, which is the n th column of the matrix V . We obtain the Gaussian filter by using this covariance matrix in a n -dimensional Gaussian function as

$$P(\mathbf{r}) = \frac{e^{-\frac{1}{2}\mathbf{r}^T\Sigma^{-1}\mathbf{r}}}{\sqrt{(2\pi)^n \det(\Sigma)}} \quad (3.6)$$

where \mathbf{r} denotes the position vector. A two dimensional example of such a Gaussian filter is given in Figure 3.7. Observe that a Gaussian filter using the kernel shown in Figure 3.7 will smooth more in the $y = x$ line and less in the perpendicular direction. In this example the matrices D and V are 2×2 matrices since the dimension of the problem is 2. For an n -dimensional system the same matrices will be $n \times n$.

We exploit the same idea in completing the iso-curvature regions on the colon

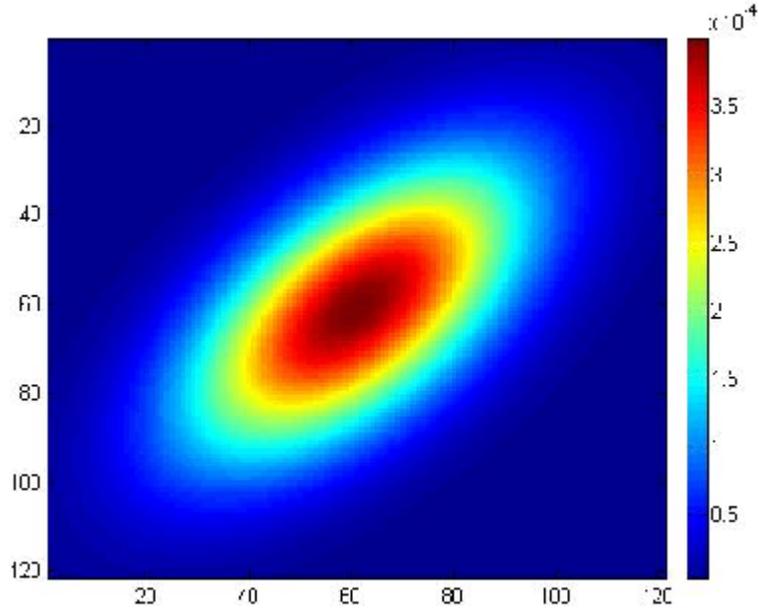


Figure 3.7. 2D example of a nonsymmetric Gaussian kernel

wall. The problem is simply directional smoothing on a surface, which is embedded in a 3D function. As we previously discussed, we use the principal vectors to create the necessary matrices. At a point on the polyp neck the second principal vector is along the neck and the first one is towards the polyp apex. So the desired Gaussian filter should smooth more in the second principal vectors direction and less in the other one.

We calculate a covariance matrix hence for each point. The covariance matrices are formed using the principal vectors and the normal vector. The diagonal matrix is used to set the amount of smoothing along each direction. We calculate the covariance matrix as follows,

$$\Sigma = \begin{bmatrix} \vec{t}_1 & \vec{t}_2 & \vec{N} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \vec{t}_1 \\ \vec{t}_2 \\ \vec{N} \end{bmatrix} \quad (3.7)$$

where \vec{t}_i for $i = 1, 2$ are the principal vectors and \vec{N} is the normal vector, ($\vec{N} = \vec{t}_1 \times \vec{t}_2$). As a convention we will call \vec{t}_2 as the second principal vector whose direction is along the polyp neck. λ_i for $i = 1, 2, 3$ are the smoothing coefficients along the corresponding directions. Since we want to smooth more in the \vec{t}_2 direction, λ_2 will

Table 3.1. Anisotropic smoothing coefficients

Coefficients	Values
λ_1	1.5
λ_2	6
λ_3	4

be the highest. (Table 3.1 shows the values for these parameters to be used at every point smoothing will take effect.)

The first diagonal element λ_1 is set to 1.5 which is the lowest of all. The reason for this is not to distribute the curvature information on the necks towards polyp apices and thus prevent different regions from merging. The second element sets the amount of smoothing in the second principal direction (along the polyp neck) and thus it has the highest value to complete the disconnected neck regions. Notice that the third element, which sets the amount of smoothing in the normal direction to the surface at that point, is significantly high. By setting this element to a relatively high value, we achieve smoothing along neighborhooding iso-levels of the implicit function. So that we smooth out the discontinuities on each iso-level as well as across iso-levels. The effect of directional smoothing on the principal curvature maps can be better observed on the speed distribution images given at the end of Section 3.2.1.3.

3.2.1.3. Speed Function. In Section 2.3, we have seen that the speed function in the level set equation can be divided into three different parts. The second and the third components, the curvature driven component and the drift velocity, will not be useful to us in creating the desired flow. The desired flow sinks polyp necks towards the tissue and pulls polyp apices towards lumen air. To create this motion we are only using the propagation speed term, which gives the speed in the normal direction. Due to our sign assignment in the construction of the implicit function Φ the direction of the normal vector at every point on the surface is towards the colon tissue. In light of this, the propagation speed assigns, depending on the curvature characteristic of a

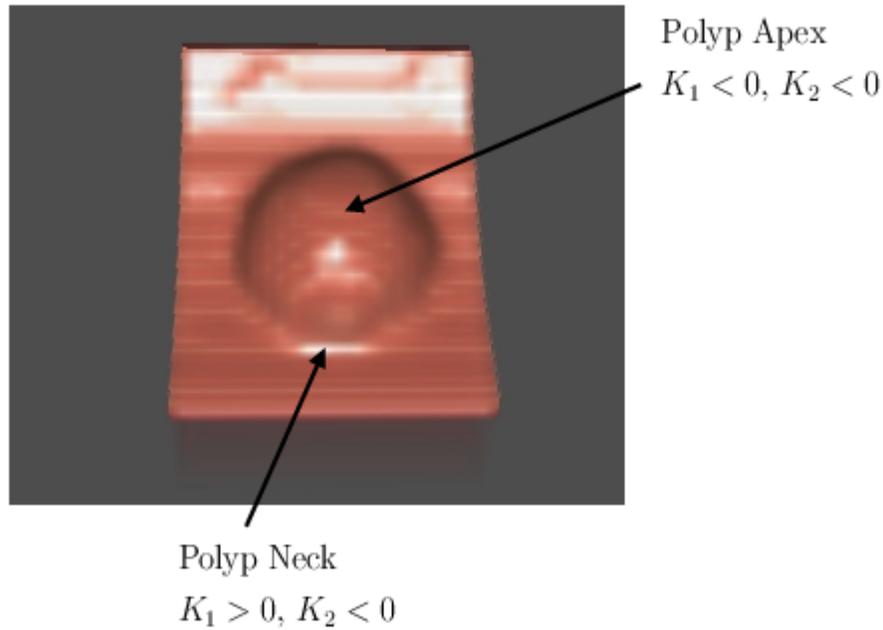


Figure 3.8. Curvature characteristics around a polyp

neighborhood, positive values if the point is similar to point on a polyp neck, negative values if the point is similar to a point on a polyp apex and zero otherwise.

In order to be able to assign correct speed values to neighborhoods we should understand the curvature characteristics of local neighborhoods. Initial observations on the real data show that the polyp apex has a convex structure and it should have both principal curvatures smaller than zero (due to our sign assignment). The polyp neck on the other hand has negative second principal curvature and positive first principal curvature, in other words it has a saddle shape. Figure 3.8 illustrates this on a phantom polyp. Although the convexity characterizes points on polyp apices, characterizing polyp necks as saddle is not enough. As there are other structures on which local neighborhoods behave like a saddle. These structures can be roughly classified into three different categories, junction of folds, fold necks and noisy bumps in planar regions. Examples of these points can be seen in Figure 3.9. However, these three categories differ from the polyp neck in several aspects. The difference between points on the polyp neck and points on the bumpy planes is the magnitude of principal curvatures. Bumps on planar regions are not visible and even if they are not cleared by the smoothing, they have small curvature values. Points on a fold neck also have the

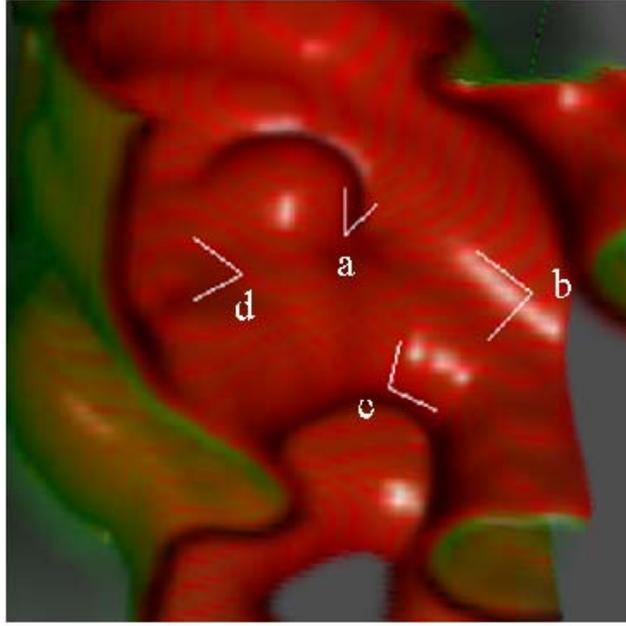


Figure 3.9. Local neighborhoods having saddle-like curvature characteristics: (a) polyp neck (b) fold necks (c) junction of folds and (d) bumpy planes

same property. Even though the magnitude of one of the principal curvature values is high, the other one is small. The difference between junction of folds and polyp necks is of a different nature. The magnitude of the first principal curvature is higher than the second one on polyp necks, hence the mean curvature at those points are positive. The situation is reversed in the case of a junction of folds, i.e. the mean curvature is negative.

We based our propagation speed function on these observations. It is defined in a piecewise manner in order to provide different motions on polyp necks and on polyp apices. Our propagation speed function is

$$F_{prop}(\kappa_1, \kappa_2) = \begin{cases} \frac{4}{(1+e^{300\kappa_2})(1+e^{-300(\kappa_1+\kappa_2)})(1+e^{-40\kappa_1+7.5})} & \kappa_1 > 0 \\ -0.05 & \kappa_1 < 0 \end{cases} \quad (3.8)$$

where κ_1 is the first principal curvature and κ_2 is the second principal curvature. The propagation term assigns -0.05 to convex regions making them move towards lumen air. For all other parts that are not convex the propagation term assigns an output of a nonlinear function of principal curvatures. The graph of the propagation term

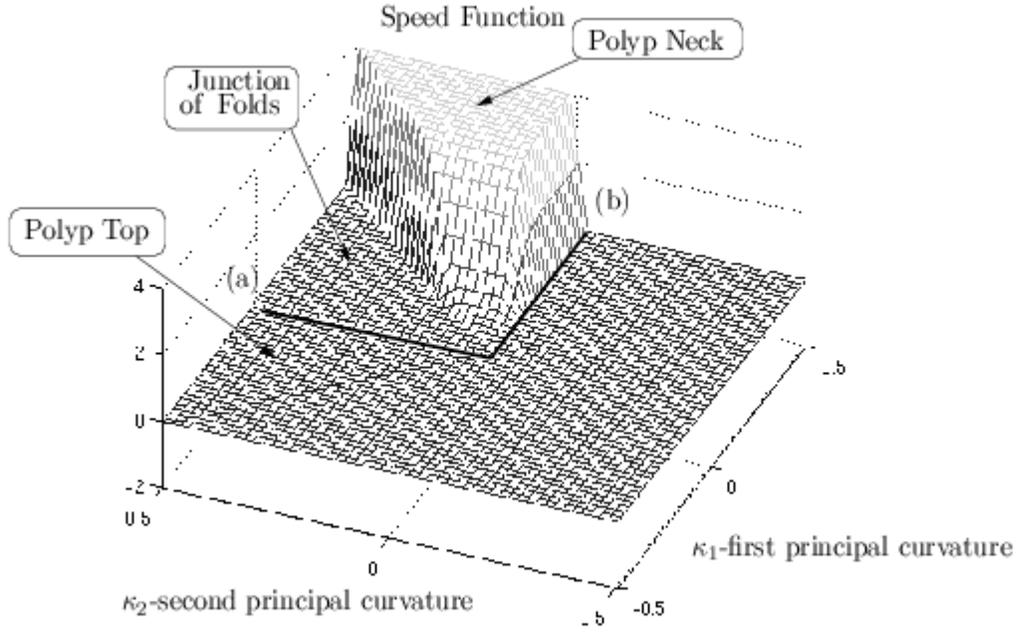


Figure 3.10. The graph of the speed function $F_{prop}(\kappa_1, \kappa_2)$: Line (a) corresponds to haustral fold apex and line (b) corresponds to fold necks

and different regions on this graph corresponding to geometric structures on the colon wall is given in Figure 3.10. This function is very intuitive and another function with similar properties could be used as well. The numeric values used in this function are set empirically. The value 300 in the first two terms is used to separate different regions shown in Figure 3.10. The parameters 40 and 7.5 together with the third component of the denominator are used to obtain a smooth transition near the point $(\kappa_1, \kappa_2) = (0, 0)$ so that we do not evolve structures with very small κ_1 and κ_2 values since they correspond to bumpy areas on the colon wall. The value -0.05 is set to a small value not to grow the polyp too much and change the location of its center.

The speed term explained in this section is created by only considering the colon wall, in other words points corresponding to the zero level set of Φ . However, we should assign appropriate speed values for all other points in the domain of Φ as explained in Section 2.3.1. Rather than extending the speed values computed for the colon wall into the complete domain, we compute it for every point (with respect to the corresponding level set) as given in Equation 3.8. This way we include the information, brought in

by the signed distance function of the surface into the complete flow.

Figure 3.11 shows the speed distribution for one of the subvolumes. In the same figure we also provide the speed distribution calculated without applying directional smoothing to the curvature maps prior to speed calculations. Thus, the effect of directional smoothing on the speed computations can be observed. In this figure the same polyp is shown from two angles. We see that the neck part of the polyp on the right is captured better when we apply the directional smoothing.

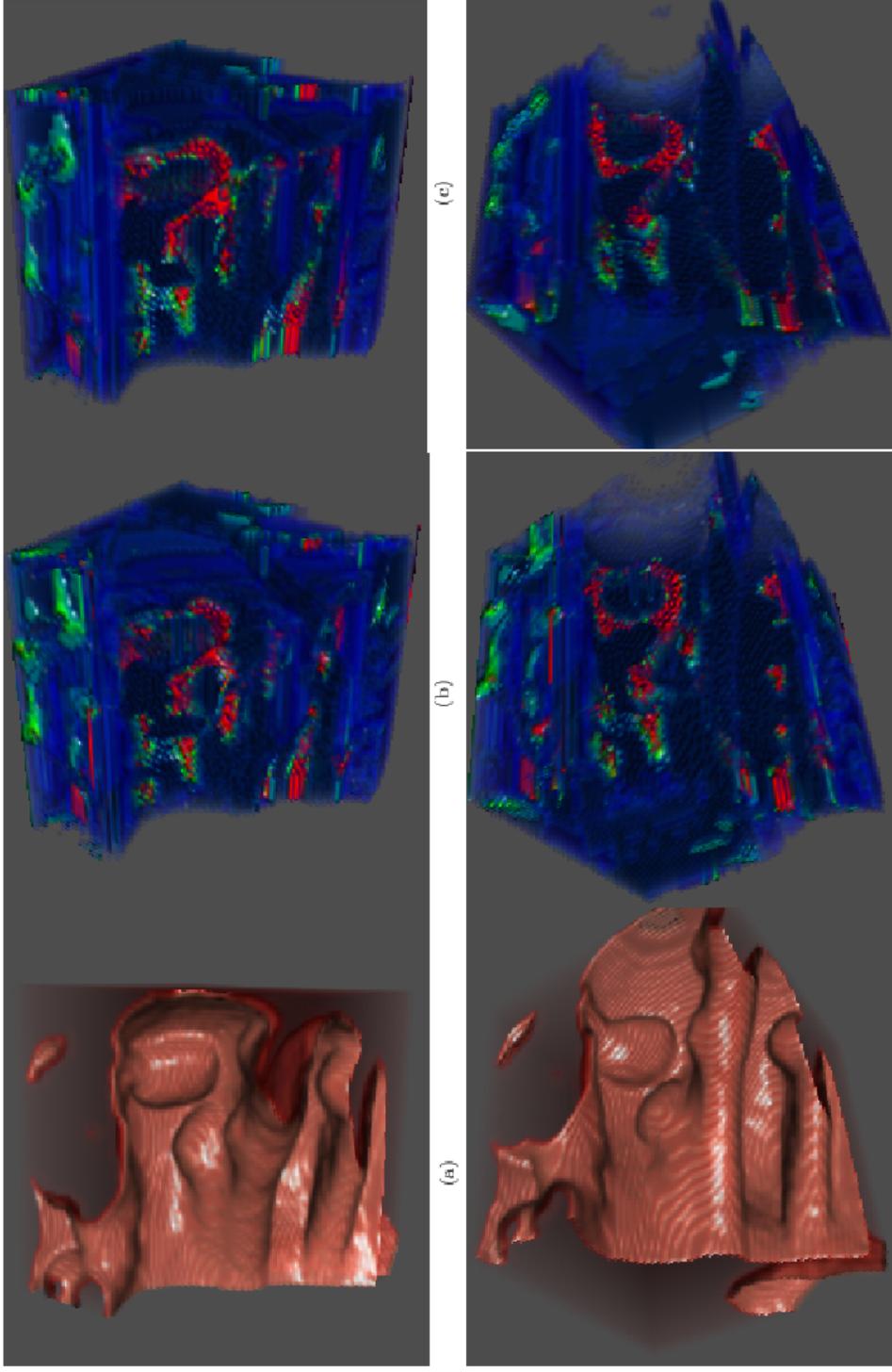


Figure 3.11. Speed distributions (red = 4.0, blue = 0.0): (a) Original segmented surface (b) Speed distribution without directional smoothing (c) Speed distribution with directional smoothing

3.2.2. Volume Maximizing Constant Surface Flow

The volume maximizing constant surface flow (VMCSF) aims to enhance colonic polyps using a variational approach. The desired flow in this section moves the colon wall so as to increase the spherical symmetry of polyp apices, and as a result to increase the performance of CAD algorithms (since the spherical symmetry of colonic polyps is the main assumption under almost all CAD algorithms). The problem of increasing the spherical symmetry of certain parts of a surface, like polyps on the colon wall, is a hard problem to tackle at first. Instead, in order to build the variational framework for this problem we start from a simpler one. We derive the necessary flow that would increase the spherical symmetry of the whole of any closed surface, using variational methods in Section 3.2.2.1. In Section 3.2.2.2 we explain how to apply the derived results to the colon wall and solve the original problem.

3.2.2.1. Theoretical Derivation. Increasing the spherical symmetry of a closed surface is equivalent to evolving the surface towards a sphere. Thus we initially tackle the problem of finding the flow that evolves any closed surface towards a sphere. Observe that a sphere has the maximum volume for a given surface area. So a flow that maximizes the volume while keeping the surface area constant will evolve a given closed surface towards a sphere and increase its spherical symmetry. To find this flow we are using the constrained maximization problem given as

$$\begin{aligned} J(\Omega) &= \int_{\Omega} dv \\ A &= \int_{\partial\Omega} da \end{aligned} \tag{3.9}$$

where $\Omega \subset \mathbb{R}^3$ is a bounded open set with regular boundary $\partial\Omega$, $A \in \mathbb{R}$ is the surface area and J is the volume functional we want to maximize. This form of the problem requires us to solve regional integrals, however all the information required by the volume functional is contained in the enclosed surface. Hence, instead of dealing with the volume integral we can transform the region functional into a boundary functional, using the method explained in [40]. Let $S(u, v) = \{x(u, v), y(u, v), z(u, v)\}$ be a differ-

entiable parametrization of $\partial\Omega$ and w be the unique solution of the Poisson equation in Ω given as

$$\begin{aligned}\Delta w &= -1 \text{ in } \Omega \\ w|_{\partial\Omega} &= 0\end{aligned}\tag{3.10}$$

By saying that $S(u, v)$ is differentiable we mean it has continuous partial derivatives of all orders. Since $\partial\Omega$ is regular it can be shown that w has continuous partial derivatives up to second order. Using the parametrization, $S(u, v)$, and w we can rewrite the constrained maximization problem in terms of surface integrals as

$$\begin{aligned}J(S) &= \int_u \int_v \nabla w \cdot (S_u \times S_v) dudv \\ A &= \int_u \int_v |S_u \times S_v| dudv\end{aligned}\tag{3.11}$$

where S_u and S_v are tangent vectors of the surface $\partial\Omega$ [40].

In order to build the link between the maximization problem and the surface evolution, we use the *gradient descent* algorithm. This algorithm iteratively finds the extremum of a function or a functional by moving in the reverse direction of the gradient, starting from an initial guess. Let us briefly explain the algorithm for a function. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, be the differentiable function we want to find the extremum of, and let $\mathbf{r}_0 \in \mathbb{R}^n$ be the initial guess. Starting from \mathbf{r}_0 we search for the steady-state solution of

$$\frac{d\mathbf{r}}{dt} = -\nabla f\tag{3.12}$$

to find the extremum of f , where ∇ is the gradient operator. This equation can be deduced from the Taylor series expansion of f around $\mathbf{r}_{extremum}$. Observe that Equation 3.12 reaches a steady-state solution when \mathbf{r} reaches an extremum of f . In the gradient descent algorithm we use discrete differentiation to find the steady-state solution of the differential equation 3.12. After replacing the time derivative with forward differencing

we obtain the following update rule, where Δt is the time step used in discretization.

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \nabla f \Delta t \quad (3.13)$$

Using the same idea as in Equations 3.12 and 3.13, we can write the equation that is to be solved in order to find the surface $\partial\Omega$, which extremize the functional \widehat{J} .

$$\frac{\partial S}{\partial t} = -\delta \widehat{J} \quad (3.14)$$

where S is a parametrization for the surface and δ denotes the first variation. This equation gives us the evolution equation that will move any closed surface to a sphere with the same surface area. Notice that the functional in Equation 3.14 is not J but \widehat{J} , which also bears the constraint information. When the extremum is reached the right hand side of this equation, first variation of \widehat{J} , becomes zero, and solves the *Euler-Lagrange* equations for the constrained maximization problem given in Equation 3.11. As explained in [41, 42], using the Lagrange multipliers technique the Euler-Lagrange equation for this problem is written as:

$$\begin{aligned} F_x - \frac{\partial}{\partial u} F_{x_u} - \frac{\partial}{\partial v} F_{x_v} &= 0 \\ F_y - \frac{\partial}{\partial u} F_{y_u} - \frac{\partial}{\partial v} F_{y_v} &= 0 \\ F_z - \frac{\partial}{\partial u} F_{z_u} - \frac{\partial}{\partial v} F_{z_v} &= 0 \end{aligned} \quad (3.15)$$

where $\{u, v\}$ are the independent variables, $\{x, y, z\}$ are the dependent variables, subscripts denote partial differentiation and F is defined using integrands in Equation 3.11 and a Lagrange multiplier λ as,

$$F = \nabla w \cdot (S_u \times S_v) - \lambda |S_u \times S_v| \quad (3.16)$$

Putting F into Equation 3.15 and simplifying with algebraic manipulations, shown in Appendix A, we showed that the Euler-Lagrange equations given in Equation 3.15 are

equivalent to

$$-(1 - 2\lambda H)(S_u \times S_v) = 0 \quad (3.17)$$

where λ is the Lagrange multiplier regarding the area constraint and H is the mean curvature of the surface $\partial\Omega$ parametrized by $S(u, v)$. We mentioned that at the extremal, Equation 3.17 is satisfied. In other words at the extremal, for every point on the surface $H = \frac{1}{2\lambda}$, a constant value. Sphere is a solution for this constrained problem since it is one of the constant mean curvature surfaces [38]. Using this information and the constraint we set $\lambda = \frac{1}{2H_0} = 4\sqrt{\frac{A}{\pi}}$ where H_0 is the mean curvature of a sphere with surface area A . Putting it all together we can rewrite evolution equation given in Equation 3.14 as

$$\frac{\partial S}{\partial t} = \left(1 - \frac{H}{H_0}\right)(S_u \times S_v) = \left(1 - \frac{H}{H_0}\right)|S_u \times S_v|\vec{N} \quad (3.18)$$

Notice that the term $S_u \times S_v$ is equal to $\vec{N}|S_u \times S_v|$, where \vec{N} is the outwards normal vector of the surface $\partial\Omega$. The term $|S_u \times S_v|$ depends on the parametrization of S . However, this evolution equation can also be formulated using level sets by embedding the surface $S = \partial\Omega$ into an implicit function Φ such that $\partial\Omega = \{(x, y, z) | \Phi(x, y, z) = 0\}$. Since the parametrization will not be of any importance in this case, we can think of $|S_u \times S_v|$ as a constant. The level set equation of the flow, which we were trying to construct, is

$$\frac{\partial \Phi}{\partial t} = \left(1 - \frac{H}{H_0}\right)|\nabla \Phi| \quad (3.19)$$

Observe that the only parameter that includes "global" information about the surface in this equation is H_0 , which has the initial surface area information. The mean curvature H and the gradient magnitude $|\nabla \Phi|$ are found using local neighborhoods, hence contains only local information about the surface around a point. When this flow is applied to a closed surface, all points evolve in order to make their mean curvature

value H_0 , independent of all other points on the surface except their own local neighborhoods. In other words, if H is less than H_0 at a point \mathbf{r} , then $1 - \frac{H}{H_0}$ will be positive and the point \mathbf{r} will move outwards and vice versa. As the mean curvature of a point gets closer to H_0 it slows down and totally stops when it reaches H_0 . When all points on the surface stop we obtain a sphere. Figure 3.12 shows the evolution of a cube under this flow. We see that the corners, where the mean curvature value is higher than H_0 , move inwards and sides of the cube, where the mean curvature is lower than H_0 , move outwards. These observation will be important when we want to apply this flow to colon wall for polyp enhancement, in Section 3.2.2.2.

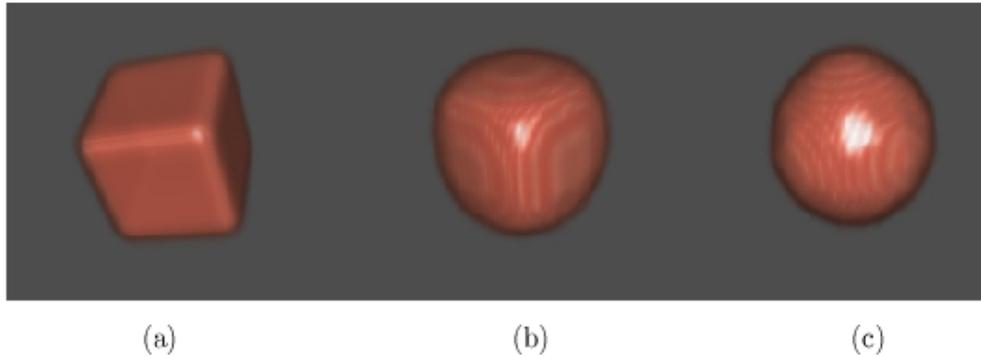


Figure 3.12. Examples of the flow that evolves 3D shapes towards sphere: (a) Initial shape, a cube (b) After 10 iterations (c) After 25 iterations

3.2.2.2. Application to Polyp Enhancement. In applying the above derivations to the case of colonic polyps we are going to use the propagation speed to evolve the surface using level set flow given in Equation 3.19. In Section 3.2.1.3 our speed function was derived based on geometric observations hence applying to the colon wall was straightforward. Applying the theory given in Section 3.2.2.1 derived using variational approaches is a bit more tricky and bears several questions.

In the previous section, we derived the level set flow that evolves closed surfaces towards a sphere with the same surface area as the initial surface. This flow was constructed for closed surfaces with known area. In order to apply it to the colon wall we need to solve several problems arising from the differences between closed

surfaces and the colon wall. The initial difference is that the colon wall does not have to be closed in an arbitrary subvolume. This does not create a big problem though since we know that the colon wall patch contained in a subvolume divides it into two distinct regions and the level set formulation is applicable for this case. The second problem is that in the previous section while we were constructing the optimization problem we have considered all the points of the initial surface. Hence in order to solve the optimization problem we need to apply the resultant flow to every point on the surface. For the colon wall on the other hand we do not wish to apply this flow to the whole colon wall, since that would damage any CAD algorithm. We want to increase the spherical symmetries of polyp apices and leave other parts unchanged as much as possible. As we have discussed before, fortunately in the level set flow defined in Equation 3.19, every point moves independent of all other points of the surface as long as the "global" parameter H_0 is given. Therefore in the case of the colon wall, applying the described flow only to some parts of the surface is not a problem, if we can define an H_0 parameter for every point to be evolved. As a result of these differences two problems arise:

1. How to choose the points that should be evolved using the flow?
2. How to set the H_0 parameter for these points?

Most authors have used the assumption that points on polyp apices are spherical points in their algorithms. We do not want to use this assumption since points on sessile polyps do not have to be spherical. Instead, we are using the assumption that points on polyp apices are convex points, which is much more general. So we solve the first problem above by choosing all convex points on the colon wall to evolve using the derived flow. The set of all convex points include points on polyp apices and points from other structures like folds and bumpy planes as can be seen in Figure 3.13.

The second problem needs extra care. In Section 3.2.2.1, we have always assumed that we know the surface area of the initial shape. The surface area information is used in finding the H_0 parameter. Observing Equation 3.19, we notice that if we set the H_0 parameter as the mean curvature of a sphere with smaller surface area than the initial

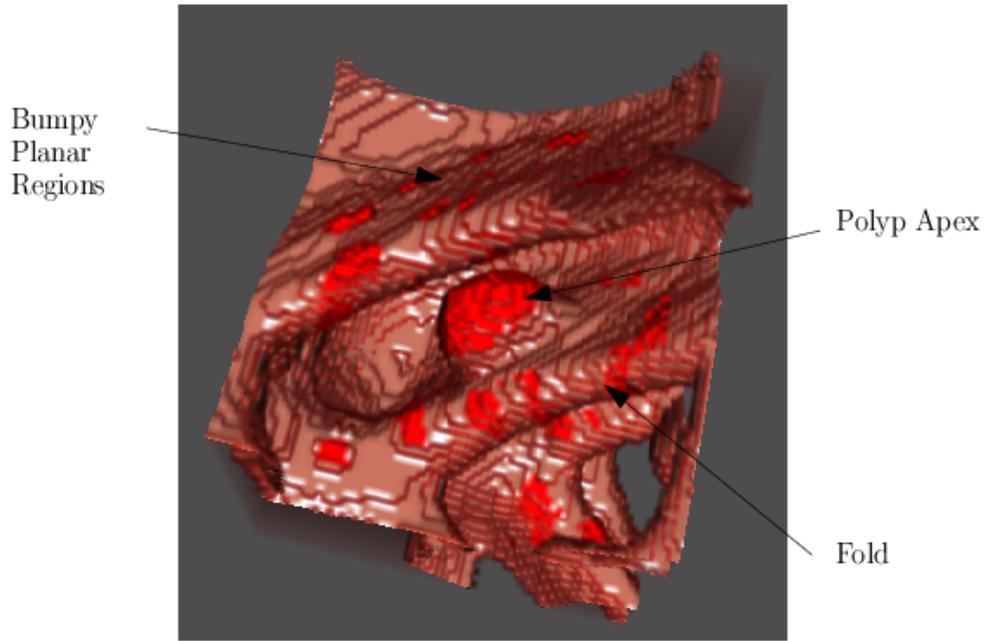


Figure 3.13. The red spots on the colon wall are the convex regions

surface, the evolution will first create a bigger sphere with some $H_1 < H_0$ (because of speed magnitude differences). After that point this sphere will grow indefinitely since $1 - \frac{H}{H_0}$ term will always be positive (positive speed means outwards growth). Hence the evolution will never reach the sphere with mean curvature H_0 and diverge. Similarly, if we set the H_0 parameter as the mean curvature of a sphere with greater surface area then the surface will shrink to a point. To overcome this problem, we use the fact that the surface initially evolves to a sphere with some mean curvature not equal to H_0 . While the surface evolves to a sphere the variance of mean curvature values on the surface decreases. So instead of waiting until we get the sphere with mean curvature H_0 if we end the evolution when the variance of mean curvature values on the surface drops below some small value then we can stop surfaces from growing indefinitely or shrinking to a point. Thus using the variance of mean curvatures as a stopping criteria, we can evolve surfaces towards spheres regardless of the H_0 parameter.

The H_0 parameter is related to the area of the surface that will be evolved to a sphere, however we cannot know surface areas of colonic polyps without detecting them. Even if we detect them prior to applying the enhancing algorithm, we will still

have estimates about areas, not exact values. Thus the fact that we can control the flow without the exact knowledge of H_0 is very important for the case of colon. On the other hand we cannot choose H_0 arbitrarily for every convex point on the colon. Setting H_0 to a small value would make polyps shrink and setting it to a large value would make small bumps on the colon wall grow. So, although we do not need the exact knowledge of H_0 , we still need an estimate for it.

We based our estimation scheme on the natural assumption that convex points on the colon wall that are connected to each other belong to the same structure. We start by smoothing the principal curvature maps computed in Section 3.2.1.1 using a Gaussian filter to smooth out any discontinuities. Then we find the connected convex regions throughout the whole implicit function by locating connected regions with both curvature values negative. The regions that do not intersect the zero level set of the implicit function, or intersect it with less than 30 voxels (an empirically set value) are ignored, since we are interested in structures on the colon wall. For every remaining connected convex region C_i we find different iso-level patches contained in the region. Let Γ_i^n represent the n^{th} iso-level (n level set) patch contained in the i^{th} region. We define Γ_i^n as follows,

$$\Gamma_i^n = \{\mathbf{r} | \mathbf{r} \in C_i, \Phi(\mathbf{r}) \in (n - 0.5, n + 0.5]\} \quad (3.20)$$

Following this we mark the spherical points on each Γ_i^n . We use the dimensionless coefficient of sphericity proposed by Summers in [9]. This coefficient is defined as

$$\varrho = \frac{|\kappa_1 - \kappa_2|}{|H|} \quad (3.21)$$

where κ_i is the respective principal curvature and H is the mean curvature. For a perfectly spherical point, ϱ becomes 0 and it is strictly positive otherwise. A point is assumed to be spherical if $\varrho < 1.0$. We estimate $H_0 = H_0(\mathbf{r})$ for the convex point \mathbf{r} in the i^{th} region, either on the zero level set or on any other iso-level, as the mean of 60 largest mean curvature values selected from spherical points in $\Gamma_i^0 \cup \Gamma_i^{-1} \cup \Gamma_i^1$. There are certain points that should be explained in this scheme. The reason why we take the

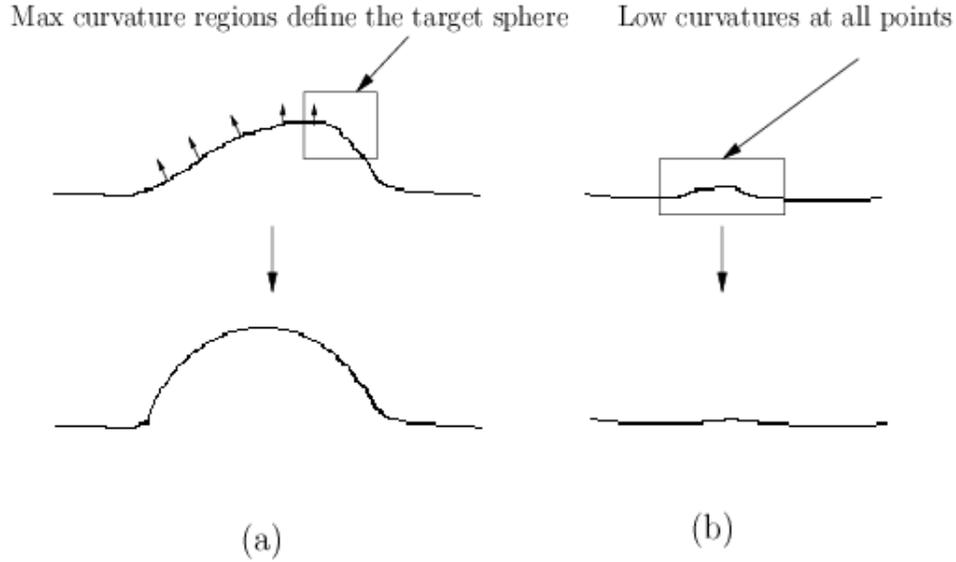


Figure 3.14. Cartoon: (a) The polyps will move towards spherical patches that are part of the smallest sphere they are contained in (b) The bumpy areas will become smoother as local H values will be small leading to a small H_0

mean of 60 mean curvature values is to get rid of any discontinuities in the curvature map, and to prevent small bumpy regions from growing. The reason why we take spherical points in computing the mean is to prevent sharp ridges from increasing this mean unnaturally. Last of all, the reason why we use the largest values of the regions to estimate H_0 , is because we want to evolve colonic polyps to spherical patches that are on the smallest sphere that contains the polyp, so that we do not shrink polyps. This idea is demonstrated in the cartoon in Figure 3.14.

Since we are using the mean of largest H values, the region will grow indefinitely if we do not use another constraint than the fixed area. Controlling the variances of mean curvature values on the surface we can stop the growth. For a connected region C_i either contained in the colon wall or spanning other iso-levels, the propagation term is set to zero if the variance of mean curvatures of points $\mathbf{r} \in \Gamma_i^1 \cup \Gamma_i^0 \cup \Gamma_i^{-1}$ at that iteration is less than a pre-specified value σ_0^2 . Once the variance limit is reached by every connected region, iterations stop and we obtain the enhanced colon wall.

Combining the solutions proposed we can write the complete propagation term as

$$F_{prop}(\mathbf{r}) = \begin{cases} 1 - \frac{H(\mathbf{r})}{H_0(\mathbf{r})} & , \sigma^2 > \sigma_0^2, \mathbf{r} \in \bigcup C_i \\ 0 & , otherwise \end{cases} \quad (3.22)$$

where \mathbf{r} is a voxel (position vector) in the domain of Φ , σ^2 is the variance of the connected convex region and σ_0^2 is set empirically as 0.0001. In this flow, as in the previous one, we do not compute speed values for the voxels on to the colon wall and then extend these values to all other voxels instead F is computed throughout the domain. Notice that the parameter H_0 and the variance is computed using only the colon wall and the immediate neighboring iso-surfaces of it. However, propagation speed values for all voxels in the domain of Φ are set using Equation 3.22. The reason for computing σ_0^2 only around the zero level set is because we are only interested in the shape of the colon wall. The reason for computing H_0 in the same manner throughout the domain is because, as the implicit function is evolved using Equation 3.19, the zero level-set (the colon wall) moves towards the other voxels in the connected region and the surface should encounter the same H_0 values at those points to continue its motion towards the estimated sphere. Figure 3.15 gives a flowchart of the whole method, we used to apply the derived level set flow to polyp enhancement.

3.3. Implementation Issues

In previous sections of this chapter we have talked about the segmentation algorithm and the speed functions we have used. In this chapter we wish to give details about the implementation of these flows.

One may choose to update the propagation terms during the level set algorithm, since the colon wall changes its location during the iterations and propagation terms at the voxels corresponding to the colon wall changes as well. We take different approaches in updating the propagation term for the two flows explained. The JCF does not care about the location of the colon wall while assigning propagation values to voxels and assigns values to all iso-levels of the signed distance function. Remembering the

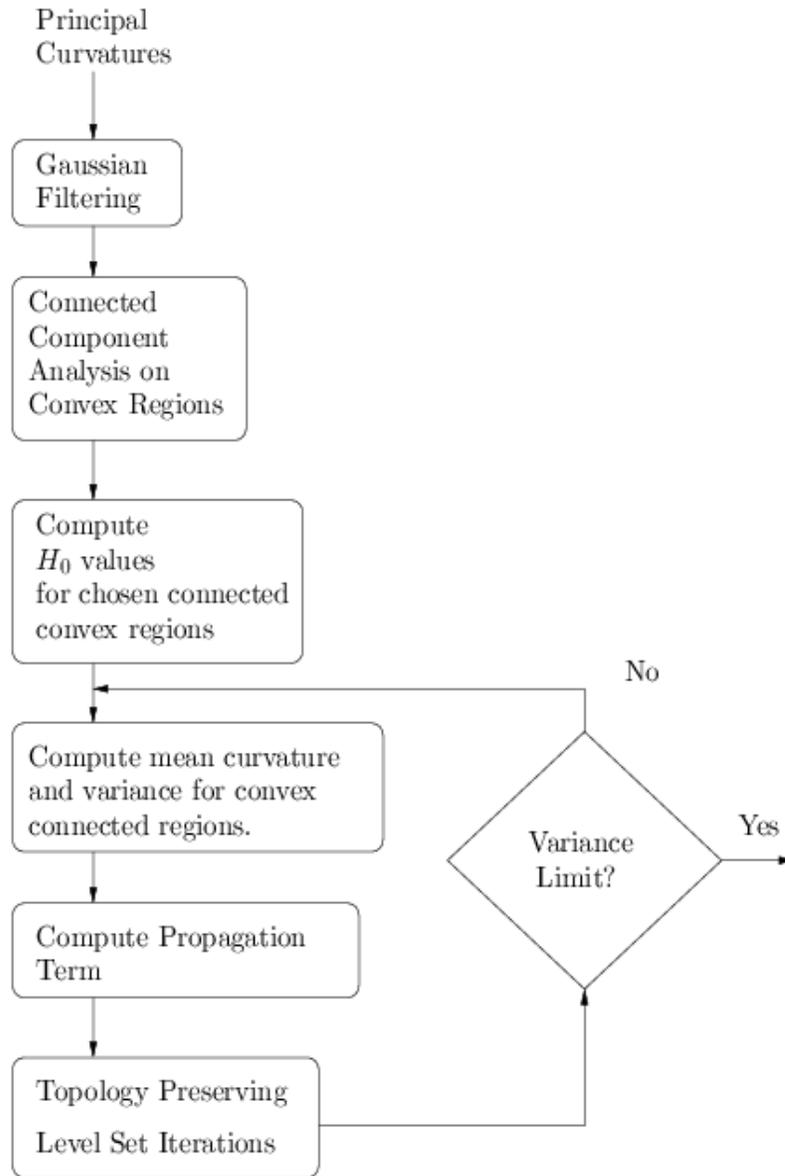


Figure 3.15. The flow chart for the VMCSF

fact that the initial signed distance function contains information about the initial colon wall, the problem of updating the surface curvatures in a flow that increases the curvedness of a surface and the computational cost of the speed computation, we chose not to update the propagation term during the level set iterations for JCF.

On the other hand, VMCSF uses the location of the colon wall while assigning propagation values to voxels. Moreover, the flow at any voxel is based on the ratio between the mean curvature value at that voxel, H , and the H_0 parameter defined for the connected convex region the voxel belongs to. If the H estimate of a point on

an iso-surface is not good enough then the flow may not evolve the surface towards a spherical patch. Hence, we update the mean curvature values and the propagation term at each voxel during level set iterations. However, because of computational cost we update the curvature values once in every ten iterations. This is observed to be acceptable for the desired flow.

The stopping criteria is also an important part of the level set formulation. For the JCF, there are two mechanisms that stop the surface evolution. The first one is the fact that as you move through iso-surfaces the curvature characteristics die off. Hence, as the colon wall moves, it arrives at regions with low propagation terms. As the speed decreases, the colon wall slows down and comes to a rest. Thus, the colon wall converges in time for JCF. In other words, the change in the values of the implicit function becomes less than a preset value (0.008). The other effect that limits/stops the motion of the surface is the topology preservation in the algorithm, since that restricts the motion of the colon wall.

In Section 3.2.2.2 we have talked about the stopping criteria for the VMCSF since that was an important issue in explaining the flow. Recall that, we use the variance of mean curvatures for every convex connected region to stop the VMCSF level set iterations.

The algorithm seen in these flowcharts was coded in C++ using Insight Toolkit (ITK) and Visualization Toolkit (VTK) libraries. It was implemented in an *Intel PentiumTM* IV machine with 2.4 GHz CPU speed and 1GB ram. To enhance a sub-volume of size $39 \times 39 \times 39 \text{ mm}^3$ took on the average 30 seconds using JCF and 10 seconds using VMCSF. Further optimizations on the code would increase the speed of the algorithm. To reduce the computation time for the whole colon, several subvolumes could also be processed in parallel with appropriate hardware.

4. EVALUATION AND RESULTS

The polyp enhancing level set algorithm (PELS) is proposed to increase the performance of existing CTC-CAD algorithms, which are based on the assumption that colonic polyps are hemi-spherical protrusions from the colon wall. Although the algorithm may seem to enhance colonic polyps visually, proper evaluation of its effects on the performance of CAD algorithms is required. To assess the enhancing algorithm we must use an existing CAD algorithm. In this chapter we explain our evaluation methodology, give details about the data set used and present experiments.

4.1. Methodology

The evaluation method we used is to compare the results of a previously proposed CAD algorithm applied to the original colon and to the enhanced colon. As the CAD algorithm we used the Surface Normal Overlap (SNO) proposed by Paik *et al.* in [16]. A brief explanation of the SNO algorithm was given in the Introduction, and more details are given in Section 4.1.1. In order to understand the effects of the enhancing scheme on the CAD performance we ran the SNO algorithm first on the original data set, then on the same data set enhanced by PELS. PELS algorithm, as explained above, runs on small subvolumes of size $39 \times 39 \times 39 \text{ mm}^3$ due to computational limitations. Thus, to perform a valid comparison, we segmented the colon wall and chose the subvolumes prior to applying the SNO algorithm. For both runs SNO was applied to segmented colon wall patches contained in these small subvolumes.

We have used two different criteria to assess the PELS algorithm. The first one is to compare the SNO scores of colonic polyps, which will give us a hint about spherical symmetries of them, found by the SNO algorithm before and after enhancing the colon wall. Although comparison of SNO scores gives us some insight on the performance of enhancing algorithm on colonic polyps, it does not give us information about the effect of the algorithm on other structures. The performance of a CAD algorithm depends not only on its response to polyps, i.e. its sensitivity, but also on its response to non-

polyp structures, i.e. its specificity. Thus, the effect of the enhancing algorithm on other structures than polyps in the colon is very important. In order to understand this effect, we measured the performance of the SNO algorithm using free-response receiver operating characteristic (FROC) analysis for both runs of SNO. Then we compared FROC curves obtained from the enhanced surface and the original surface. Details for the FROC analysis is given in 4.1.2.

4.1.1. Surface Normal Overlap

The surface normal overlap (SNO) algorithm uses the observation that for spherical and hemispherical structures surface normals intersect at the center points, [16]. In the algorithm each voxel in the tissue accumulates a score proportional to the number of surface normals passing through or near it. Colonic polyps tend to have convex regions on their surfaces, thus the inward pointing surface normal vectors near these regions intersect or nearly intersect within the tissue. Haustral folds also have convex regions on them, however since they have a dominant curvature along a single direction the score for folds is generally lower than polyps.

SNO counts the number of surface normals that pass through or near to each voxel. In order to limit the contributions from normal vectors coming from very distant structures, the length of projected normals is limited to 10 *mm*. Since in the real patient data colonic polyps may deviate from a perfect hemisphere, robustness to these variations is crucial. The length of projected surface normals provides robustness against objects with nonconstant surface points to center. The other problem arising from the variations is that surface normals of a polyp may never intersect inside the tissue but pass near each other. To deal with these kind of skewed surface normals SNO uses weighted cylinders of finite width to project normals, in which the weight decreases radially with a gaussian distribution of some specified variance. Weights passing from a voxel are aggregated and the final sum of these weights becomes the SNO score for that voxel. As a result spherically symmetric structures, like polyps, would yield a higher SNO scores than other structures in the colon.

4.1.2. Free-response Receiver Operating Characteristic Analysis

Receiving operating characteristics (ROC) analysis is commonly used in evaluation of CAD algorithms, [21]. In an ROC experiment the algorithm is applied to many different images, some of them containing abnormalities and some do not. Each image is rated into 5 different classes according to the confidence that the image is normal or abnormal. As an example, 5 means high confidence that the image contains abnormal structures and 1 means high confidence that the image is normal. Intermediate classes represent the levels of confidence that image is abnormal in the increasing order. The ROC curve is a plot of true positive fraction (TPF) vs. false positive fraction (FPF). It is obtained by counting first the 5-responses, then 5 plus 4 responses and so on. The area under the ROC curve is used for objective measurement on the performance of the CAD algorithm.

As explained in [21], there are several problems with the ROC analysis if we want to apply it to the case of CT-colonography, in which localization of lesions is important. The ROC analysis can be used effectively in applications where the localization is not of importance because it rates the whole image. But it cannot be used if the position of the lesion is important. One other problem is the fact that the ROC analysis does not care about the multiple responses of the CAD algorithm to a single image. The last problem is that there exists an ambiguity in the ROC analysis: Assume an image contains an abnormal structure in it and the CAD algorithm fails to detect it. If the CAD algorithm detects a false positive on the same image, then the ROC analysis will not be able to distinguish this failure. Because of all these problems ROC analysis will not be effective in evaluating our algorithm.

We are using free-response receiver operating characteristic (FROC) analysis in order to measure the performance of the SNO algorithm. For each colonic polyp, the FROC analysis finds the number of false positives with a score higher than the polyp. A colonic polyp is considered to be detected, a true positive, if the CAD algorithm detects the polyp to within a certain distance from the center of it. This distance is commonly set as the radius of the polyp as reported by a radiologist. The FROC curve

is a plot of fraction of true positives vs. the number of false positives detected until all the true positive in that fraction are found. CAD algorithms may produce multiple hits for the same polyp. Hence in order to use the true fraction of polyps while doing the FROC analysis we need to cancel multiple hits coming from the same polyp. One way of doing this, is to subsample points detected by the CAD algorithm. We first set a size limit, R_{limit} , which is the radius of the smallest polyp that will be included in the analysis. Then we sort points detected by the SNO algorithm according to their score in an descending order. Starting from the top, for each detected point m , we remove any other detected point that is closer to m than R_{limit} and has a lower score than m . This way we remove multiple detections of the same structure and keep the detection with the highest SNO score.

The FROC analysis takes into account the localization of the abnormality by the CAD algorithm and it allows multiple responds to a single image. Thus, it would be more effective in evaluating a CAD algorithm in the case of CT colonography.

4.2. Data Set

In the evaluation of PELS we have used two different real patient datasets. In the first dataset a total 8 CT colonography exams performed at either Stanford University or at the San Francisco VA hospital were used to include a reasonably large number of colonic polyps and to balance the number of patients with and without large polyps. These 8 patients were given rectal air contrast and scanned in the supine position with single- or multidetector helical CT (GE HiSpeed/CTi or LightSpeed, General Electric Medical Systems, Milwaukee, WI) with effective section width oof $2.5 - 3.75 \text{ mm}$ and 50% overlapping reconstruction. Immediately following CT scanning, each patient also underwent fiber-optic colonoscopy (FOC). These results were correlated to the CT images with a total of 7 polyps ($\geq 10 \text{ mm}$) found in 4 of 8 patients and a total of 11 small polyps ($5 - 9 \text{ mm}$) found in 3 of 8 patients. A wide range of polyp shapes were present in the dataset.

In order to create the gold standard for the dataset a study coordinator with

extensive experience in CTC and blinded to CAD results carefully reviewed the CTC data and recorded the location and diameter of polyps found by FOC. Only one significant polyp (measured as 15 *mm* by FOC) was unable to be located in the CT images, most likely due to retained water. A total of 10 small polyps (1 was 8 *mm* and 9 were 5–6 *mm* measured by FOC) were unable to be located in the CT images. These polyps are not included in the evaluation since they are not the fault of the CAD algorithm.

The second database we used was acquired from a 56 year old female patient for evaluation. The patient was scanned in the prone position in an 8 slice multi-detector row CT scanner (GE Lightspeed Ultra, Milwaukee, WI) in the 4 slice helical mode (slice width=2.5 *mm*, pitch=0.75, slice spacing=1.25 *mm*, FOV=36 *cm* reconstructed on a 512×512 matrix, kV=120, mA=120). She underwent fiberoptic colonoscopy immediately after the CT scan 47 polyps were detected - 16 smaller than 5 *mm*, 24 between 5 *mm* and 8 *mm*, 7 larger than 8 *mm*. The gold standard was generated by a radiologist with 8 years of experience in CTC, who marked the centers of colonoscopy-confirmed polyps and measured their diameters using a custom built computer program.

In evaluation of the PELS algorithm using the scheme explained in Section 4.1, we combined these databases obtaining one with 9 patients. Adding the numbers above we get 12 "clinically significant" polyps (≥ 10 *mm*) and 37 small polyps (5 – 9 *mm*) in our database.

4.3. Experiments with JCF

4.3.1. Results

Results for the non-enhanced data given in this section are obtained by combining the outputs of SNO algorithm applied to original colons of 9 patients. Similarly, the results for the enhanced data are obtained by combining the outputs of SNO applied to enhanced colons. Before giving results for the whole dataset, in Figures 4.1 we give some examples of enhanced subvolumes, and results on these. These figures illustrates the motion we obtained by applying the JCF to polyps and non-polyp structures, as

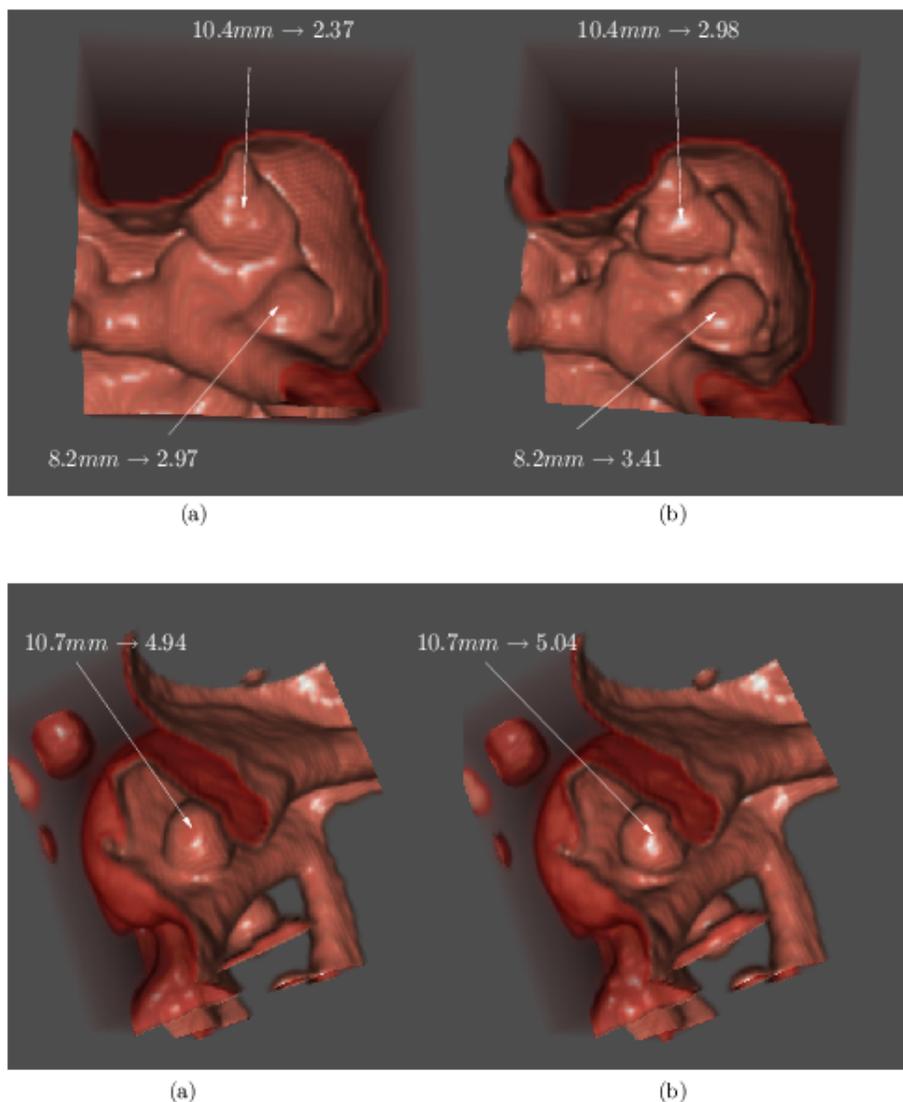


Figure 4.1. JCF results: (a) Original subvolume (b) Enhanced subvolume

well as SNO scores of two sessile and one pedunculated polyps in these subvolumes. Observe that the effect of enhancing polyps using JCF is more apparent in the case of sessile polyps. SNO scores also confirm this observation. The score for one of the sessile polyps of size 10.4 mm increased from 2.37 to 2.98 and the score for the other one, which is of size 8.2 mm increased from 2.97 to 3.41. On the other hand the score for the pedunculated polyp of size 10.7 mm remained nearly constant, it went from 4.94 to 5.04.

In Figure 4.2 we compare the SNO scores for all polyps detected by the CAD algorithm in the original and the enhanced dataset. The $y = x$ line is also drawn in the

figure to be able to make a good comparison between two scores of a polyp. If a point in Figure 4.2 is above the $y = x$ line this means that the PELS algorithm increased the SNO score for the polyp corresponding to this point. In the opposite case, the PELS algorithm decreased its SNO score. In the SNO score plot there are some points on the x or the y axis. These points correspond to polyps, which was found in one dataset and not found in the other. For example if a polyp is not detected on the original subvolume but detected on the enhanced subvolume using SNO then we mark the SNO score for that polyp on the y axis. Although we have designed the algorithm without any size considerations, it may work different on polyps of different size. Thus, in the figure we also discriminate between polyps of different size by the type of the marker used. We see that almost all marks are above the $y = x$ line, in other words

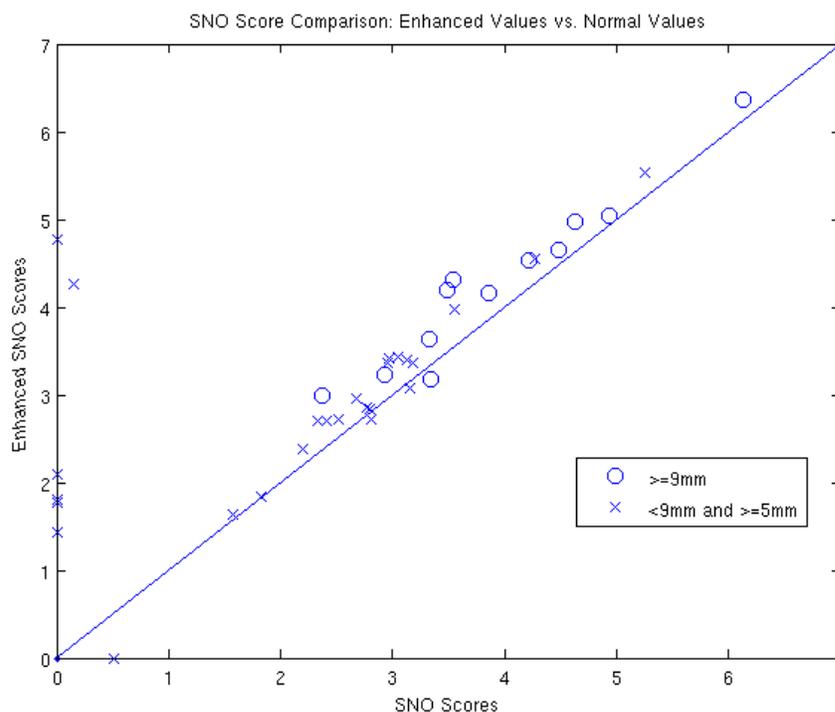
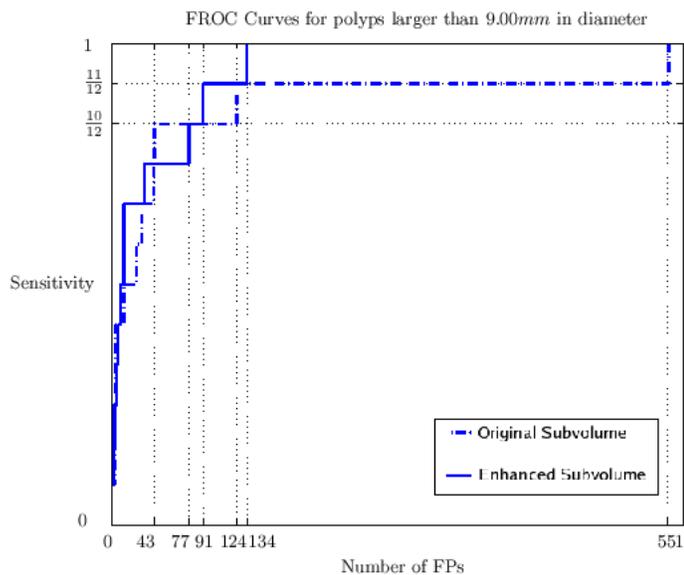


Figure 4.2. Graph of SNO scores of enhanced polyps vs. original polyps

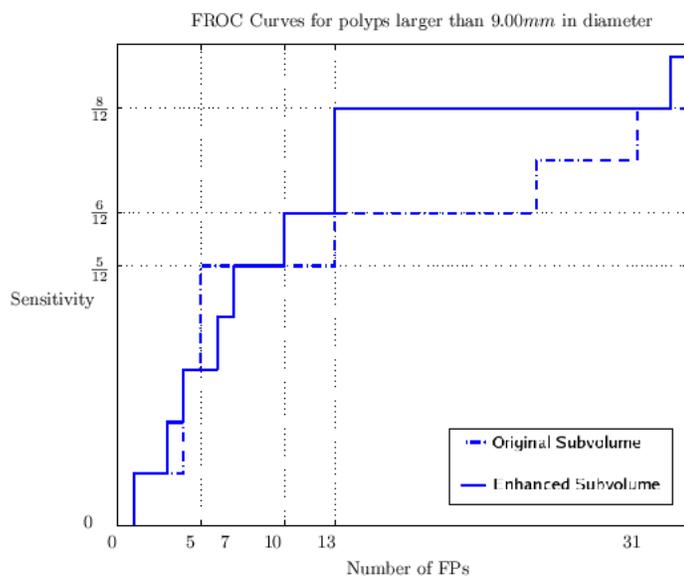
for almost all polyps SNO score increases when we enhance the colon. There are only 3 small polyps and one "clinically significant" ($\geq 9\text{mm}$) polyp below the $y = x$ line. There are 5 polyps that were not detected on the original colon but detected when the colon was enhanced. On the other hand there is one polyp that was detected in the original colon but not detected in the enhanced one.

In Section 4.1 we have mentioned the importance of the effect of the enhancing algorithm on non-polyp structures and the method to assess this. We did FROC analysis on both the original and the enhanced colons' results to evaluate the PELS algorithm with JCF and observe the change in performance of the SNO algorithm introduced by enhancing the colon. The FROC analysis is done twice, once for "clinically significant" polyps by setting the size limit we use in FROC analysis, R_{limit} , to 4.5 mm , and once for polyps larger than 5 mm by setting R_{limit} to 2.5 mm . In Figures 4.3 we provide FROC curves for "clinically significant" polyps and in Figures 4.4 we give FROC curves for all polyps larger than 5 mm . The solid lines in these figures are FROC curves for the results of SNO applied to the dataset enhanced by PELS using JCF, and the dashed lines are FROC curves for the original colon.

In Figure 4.3(a) we observe that the solid line is most of the time to the left of the dashed line. This means that less FPs are detected on the enhanced colon wall for the same sensitivity level. Numerically speaking until SNO detects all the polyps greater than 9.00 mm (12/12 sensitivity), 551 FPs were detected on the original colon while only 134 FPs were detected on the enhanced colon wall. At the 11/12 sensitivity level 124 FPs were detected on the original colon and 91 FPs detected on the enhanced colon. However there are several points where the dashed line falls to the left of the solid line. At the 10/12 sensitivity level 77 FPs were detected on the enhanced colon while 43 FPs were detected on the original colon. Similarly in Figure 4.3(b) at the 5/12 sensitivity level SNO performs better on the original colon wall. For the case of all polyps larger than 5.00 mm , the difference between the solid line and the dashed line is more apparent. The SNO algorithm does not reach 49/49 sensitivity level neither on the original colon or on the enhanced colon. On the original colon wall SNO reaches 33/49 sensitivity and detects 14792 FPs for that level, on the enhanced colon it reaches 37/49 sensitivity and detects 4201 FPs for that level. Figure 4.4(a) does not show the whole FROC curve of the original colon for visualization purposes.

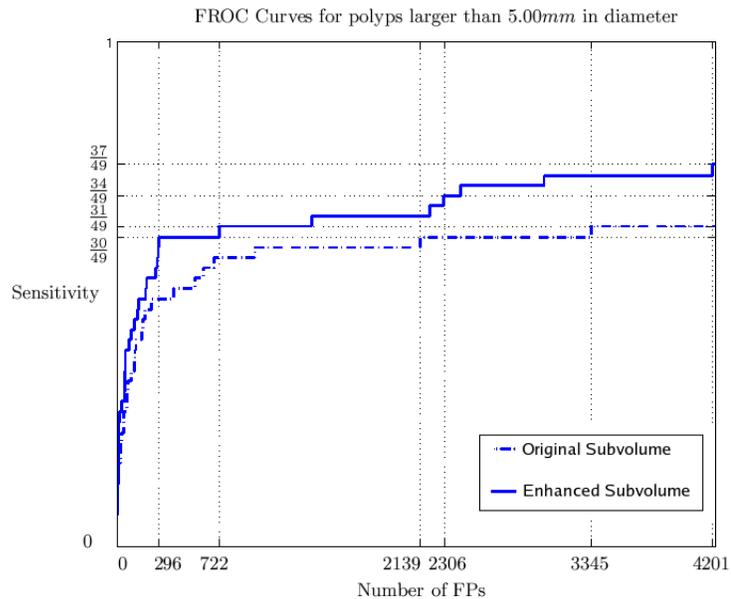


(a)

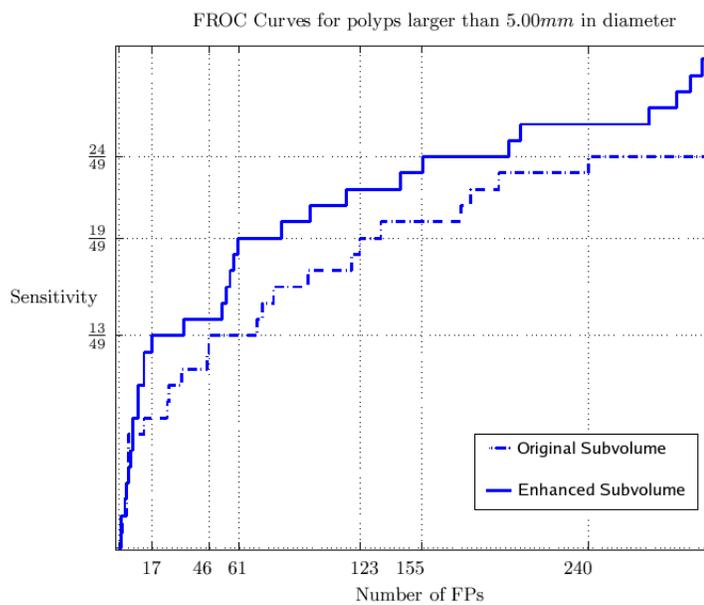


(b)

Figure 4.3. FROC curves for polyps larger than 9.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin



(a)



(b)

Figure 4.4. FROC curves for polyps larger than 5.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin

4.3.2. Discussions

Observing FROC curves in Section 4.3.1 we see that the Joint Curvature Flow (JCF) increases SNO scores for most polyps larger than 5 *mm* and decreases number of false positives for almost all sensitivity levels. The FROC curves in Figure 4.3 show that for 100% sensitivity the false positive numbers decrease from 551 to 134, which is less than a quarter of the original number of FPs. In the case of all polyps larger than 5 *mm* the improvement is even more visible. The FROC curve for the enhanced colon remains above the original one for the whole time. Moreover several polyps that were not detected before become visible to the CAD algorithm when enhanced.

Although the enhancing algorithm applied with JCF seems to improve the overall performance there are some problematic structures for which enhancing algorithm is not successful. In Figure 4.5(a) we see a polyp of size 14.0 *mm* in diameter. The SNO score for this polyp before enhancing was 3.34, but this score drops to 3.17 after enhancing it with the JCF. The reason for this is because the enhancing algorithm captures only a small part of the polyp neck, and evolving only that part decreases the intersection of the surface normals. In Figure 4.5(b) we see a false positive that was not present on the original colon wall. This false positive arises from the segmented surface patches those are not part of the colon wall but they are very close to the surface. Evolving these parts and changing their shapes introduce them as false positives. These parts are enhanced because they have similar curvature characteristics as polyp necks. The SNO score for this false positive after enhancing it is 3.14. Last of all, in Figure 4.5(c) we see a false positive whose SNO score is increased by the enhancing algorithm. The score of this thickened fold was 2.9 on the original colon, and it became 3.1 after enhancing the colon. Since the thickened fold resembles a colonic polyp geometrically, when the neck of this fold is enhanced its score increases.

These observations show us some important facts about the PELS algorithm using JCF. In order to be able to enhance the shape of the colonic polyp the JCF should be able to capture most of the neck of the polyp. For all polyps used in this work except the one shown in Figure 4.5(a) JCF captured enough of the neck of polyps so

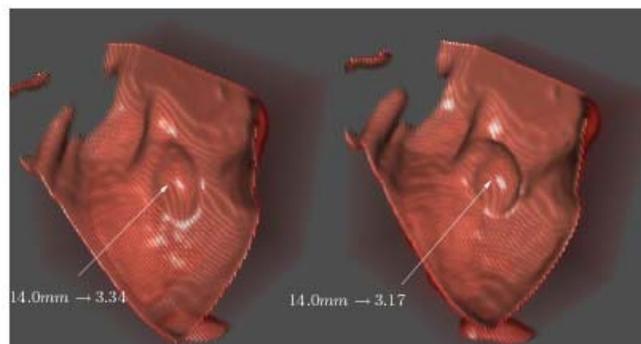
to increase their scores. The other fact is that the JCF was designed considering only the colon wall. Motion at segmentation artifacts is unpredictable since their curvature characteristics are unpredictable. Thus, these artifacts may create problems as the one seen in Figure 4.5(b).

4.4. Experiments with VMCSF

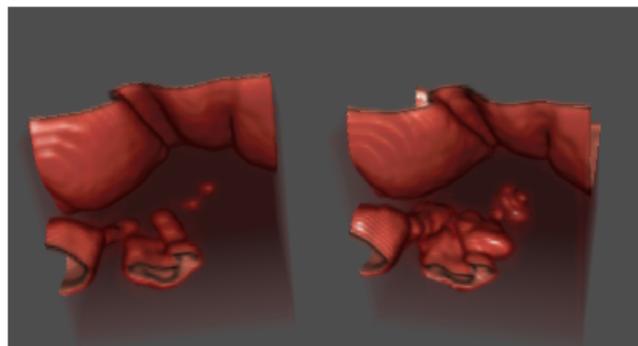
4.4.1. Results

Results in this section are obtained in the same manner as in the Section 4.4.1. In Figure 4.6 we give some examples of enhanced subvolumes. These figures illustrate the effect of the flow applied on both polyps and non-polyp structures and give SNO scores for polyps in these subvolumes. Observing these images we see that the effect of the VMCSF is more apparent on the spherically less symmetric polyps, which was our intention in creating the flow. By spherical symmetry we mean the sphericity of the patch that forms the polyp apex. SNO scores confirm this observation as well. For the two spherically less symmetric polyps in Figure 4.6, which are 10.4 and 8.2 *mm*, SNO scores increase from 2.37 to 2.52 and from 3.56 to 4.06 respectively. However for the spherically symmetric one, which is also 8.2 *mm* the score does not change much, it increases from 2.97 only to 3.07.

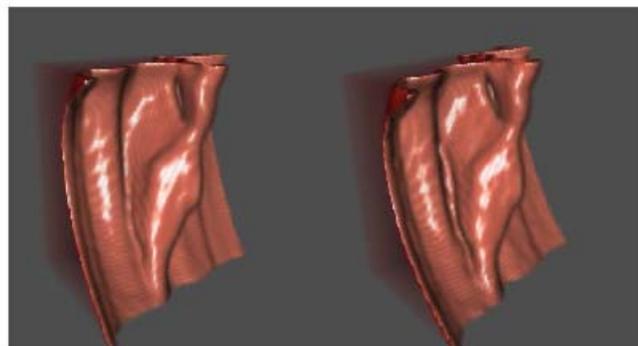
In Figure 4.7 we compare the SNO scores for all polyps detected by the CAD algorithm in the original and the enhanced dataset. This graph is plotted in the same manner as the graph in Figure 4.2, hence meaning of marks and the role of the line are the same in both figures. We observe that most marks are above the $y = x$ line, but except a few they are close to the line. In other words although there exists an increase in SNO scores, the amount is not very big. There are 5 small polyps those are below the $y = x$ line indicating that enhancing algorithm using VMCSF have decreased SNO scores for these polyps. There are 5 small polyps that were detected by the SNO in the enhanced colon but not in the original colon, and there is one small polyp for which the case is the reverse.



(a)



(b)



(c)

Figure 4.5. Problematic structures JCF is having trouble with: (a) PELS with JCF decreases the SNO score for this polyp (b) PELS with JCF introduces this false positive and (c) PELS with JCF increases the score for this false positive

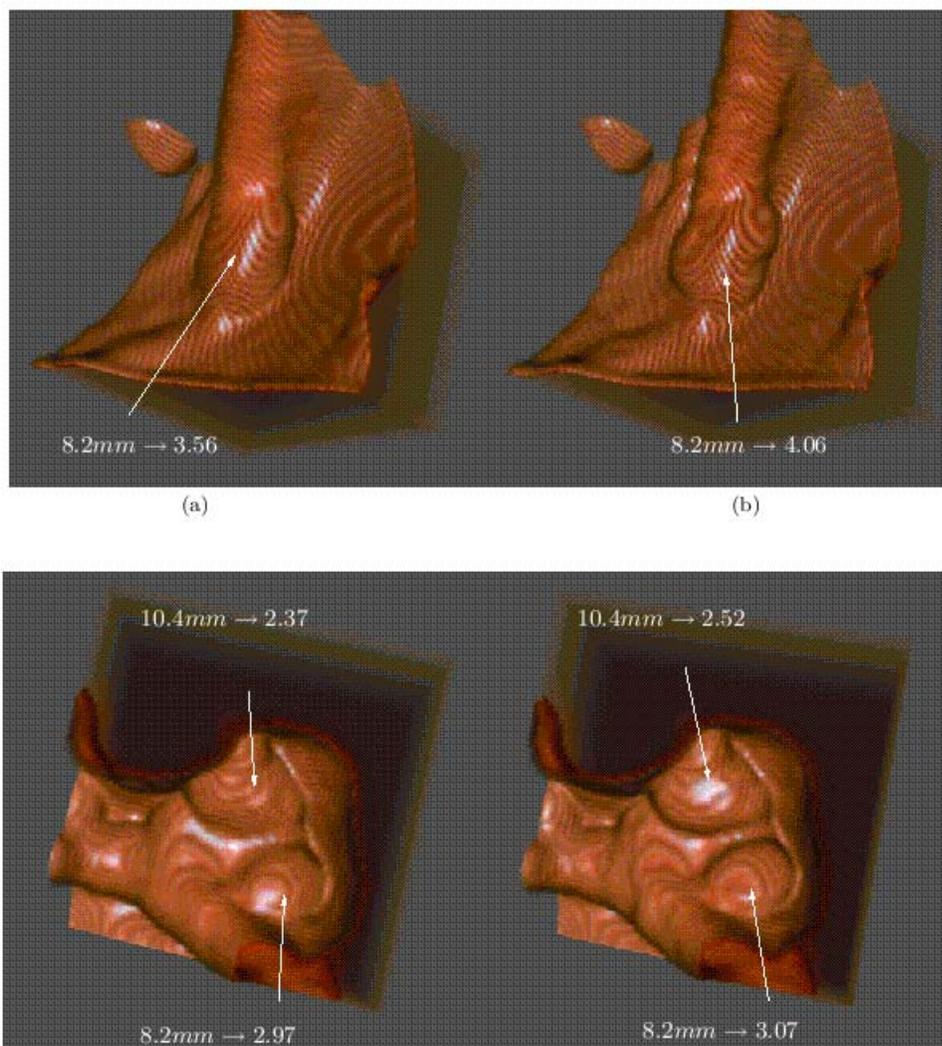


Figure 4.6. VMCSF results: (a) Original subvolume (b) Enhanced subvolume

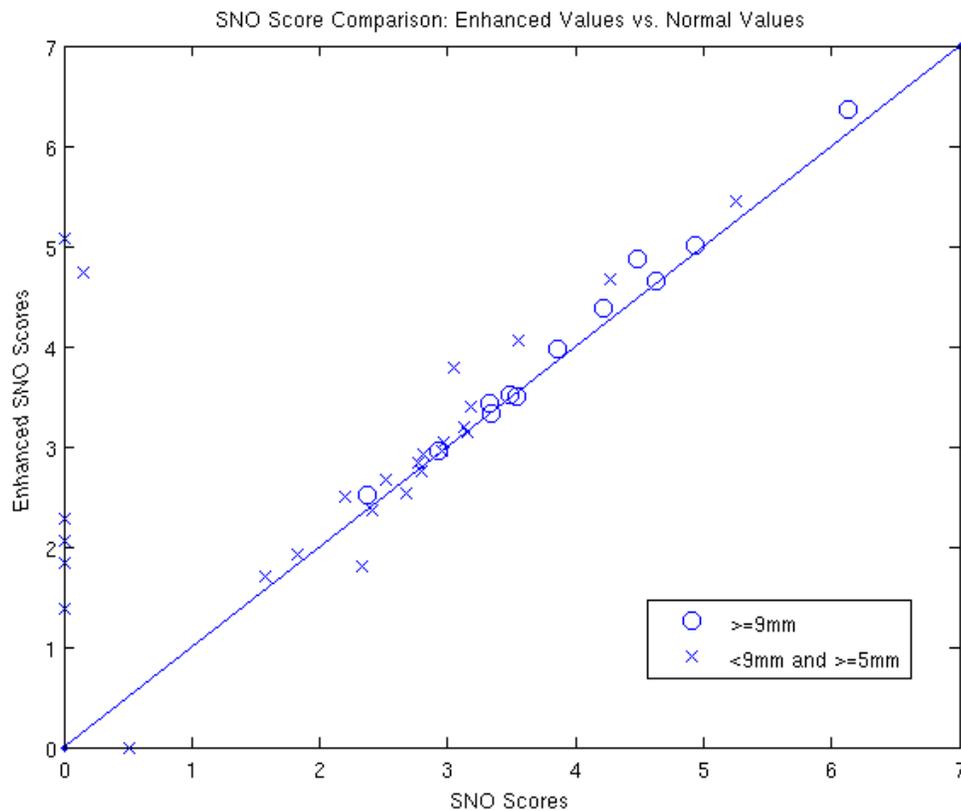
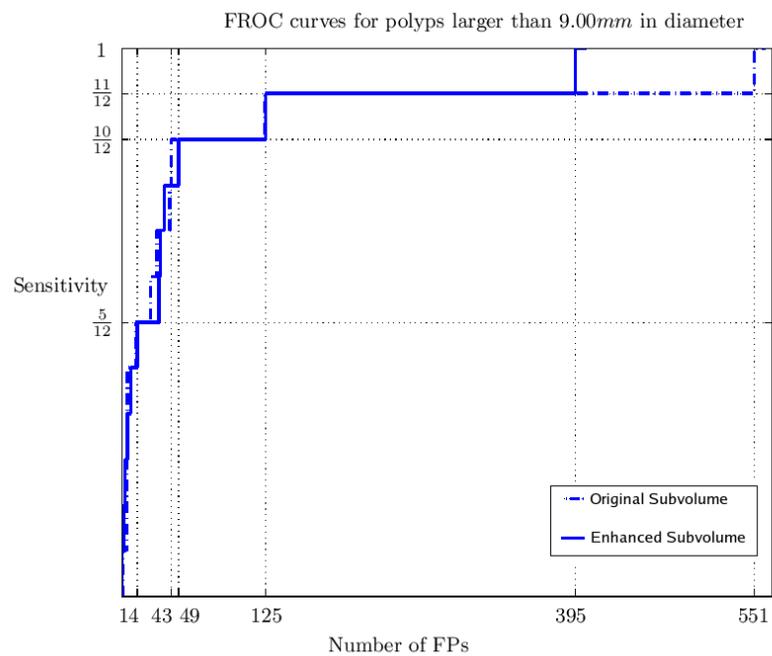


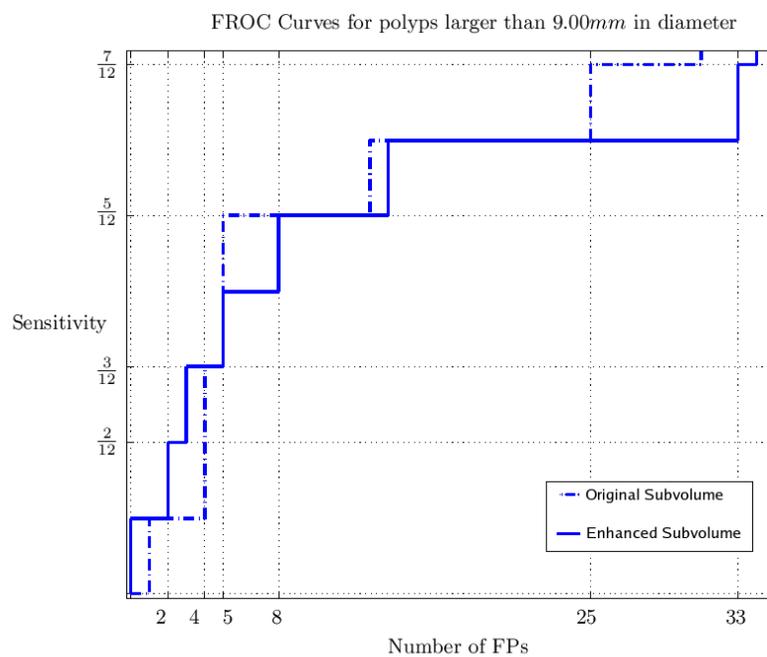
Figure 4.7. Graph of SNO scores of enhanced polyps vs. original polyps

As in Section 4.3.1 we performed FROC analysis on results of the SNO algorithm applied to the colon enhanced using PELS with VMCSF. The analysis was done twice in the manner explained in Section 4.3.1. In Figure 4.8 we provide FROC curves for "clinically significant" polyps and in Figure 4.9 we give FROC curves for all polyps larger than 5 mm. The method used to draw these FROC curves are similar to those given in Section 4.3.1, hence the meaning of solid and dashed lines are the same.

The solid line in Figure 4.8 is very close to the dashed line. The number of FPs detected for different sensitivity levels are almost the same for both the original and the enhanced colon. The only big difference between curves is the number of FPs at 12/12 sensitivity level, for which 551 FPs were found in the original colon and 395 FPs were found in the enhanced colon. Other than that, difference between FP values are small and they are in favour of the original colon. At the 11/12 sensitivity level 124 FPs were found in the original colon and 125 FPs were detected in the enhanced colon; and at the 7/12 sensitivity level 25 FPs were found in the original colon while 33 FPs



(a)



(b)

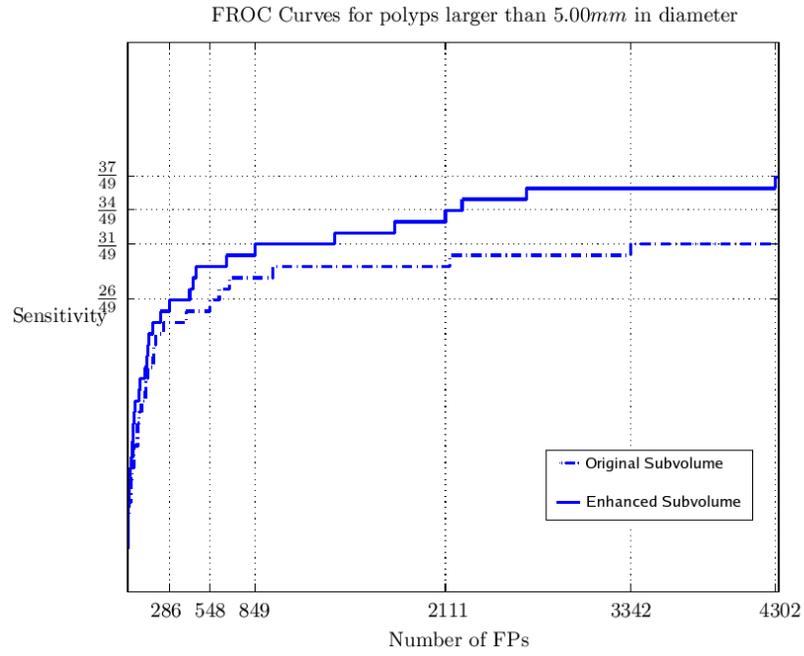
Figure 4.8. FROC curves for polyps larger than 9.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin

were found in the enhanced one. The difference between FROC curves is much more apparent in the case of all polyps larger than 5.00 mm . Once more the SNO algorithm could not reach 49/49 sensitivity level neither on the original colon or on the enhanced colon. The number of FPs detected at the maximum sensitivity level (37/49) on the enhanced colon is 4302 while the number of FPs detected on the original colon is 14792 at its maximum sensitivity level (33/49). Figure 4.9(a) does not show the whole FROC curve of the original colon for visualization purposes.

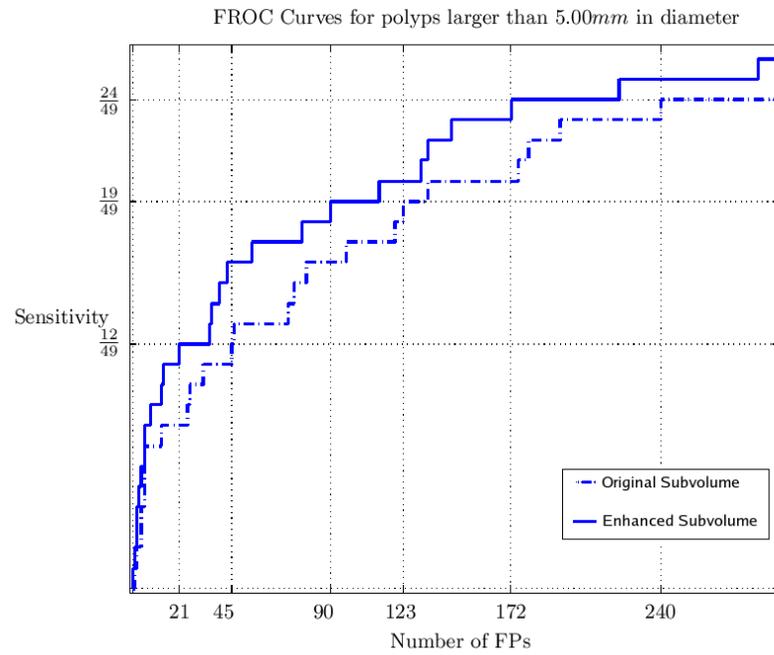
4.4.2. Discussions

The PELS algorithm when used with the Volume Maximizing Constant Surface Flow (VMCSF) increases SNO scores for all polyps larger than 9 mm and for almost all polyps larger than 5 mm . However the increase in scores of polyps larger than 9 mm is subtle as can be seen in Figure 4.7. The FROC curves seen in Figure 4.8 show that the VMCSF algorithm is not very successful in decreasing the number of false positives. Even though the number FPs drops to 395 from 551 at the 100% sensitivity, this improvement is not observed for other sensitivity levels. The FROC curves for polyps larger than 9 mm are very close to each other. On the other hand for the case of small polyps ($5 - 9\text{ mm}$) results are better. The increase in SNO scores of these polyps is more apparent and the FROC curves given in 4.9 confirm this. The FROC curve for the enhanced colon is above the FROC curve for the original colon for most of the time, showing an increase in the performance of the SNO algorithm in detecting small polyps.

There are two major drawbacks of the PELS algorithm when using VMCSF. The first one is that the SNO scores for "clinically significant" polyps do not increase much. However this is not the primary problem with this flow. The primary problem is the introduction of new false positives and the increase in SNO scores for existing false positives. Figure 4.10(a) shows one false positive that was introduced by the enhancing algorithm with a score of 3.81. We see that although the original colon wall is a cylinder, the flow is not successful in keeping this surface unchanged. The problem is in the scheme we are using to set the H_0 parameter. Ideally, this parameter



(a)



(b)

Figure 4.9. FROC curves for polyps larger than 5.0 mm: (a) The whole FROC curve and (b) Same curve but zoomed around the origin

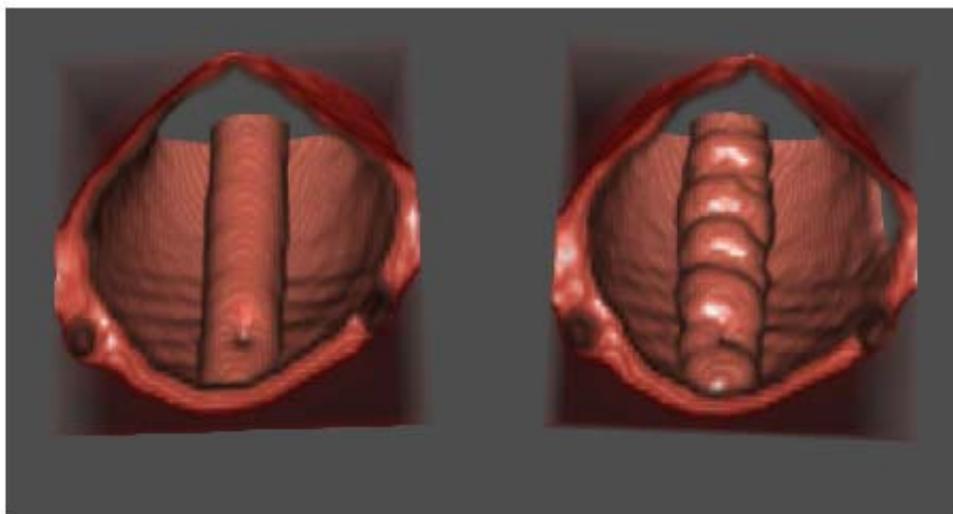
should be equivalent to the mean curvature values on a cylinder. However the convex regions in this surface patch have noisy mean curvature distributions. Moreover, we are choosing the largest 60 mean curvature values to estimate the H_0 parameter in every convex region. As a result connected convex regions on this patch grow and create false positives. The other problematic structure seen in Figure 4.10(b) is a false positive, which exists on the original colon wall but enhanced by the PELS algorithm. This thickened fold was also a problem for the JCF flow. This time however the SNO score of this structure went from 2.9 to 3.45 creating a much bigger problem. The fact that the thickened fold resembles a polyp with very little spherical symmetry makes it a perfect candidate for the VMCSF to enhance.

We see that the VMCSF algorithm is not very successful in increasing the SNO scores for large polyps. In addition to this, when the curvature distributions of connected convex regions are very noisy then the H_0 estimate becomes unrealistic and the flow may inflate or deflate regions which should not be altered at all. For these cases estimation scheme we are using becomes ineffective.

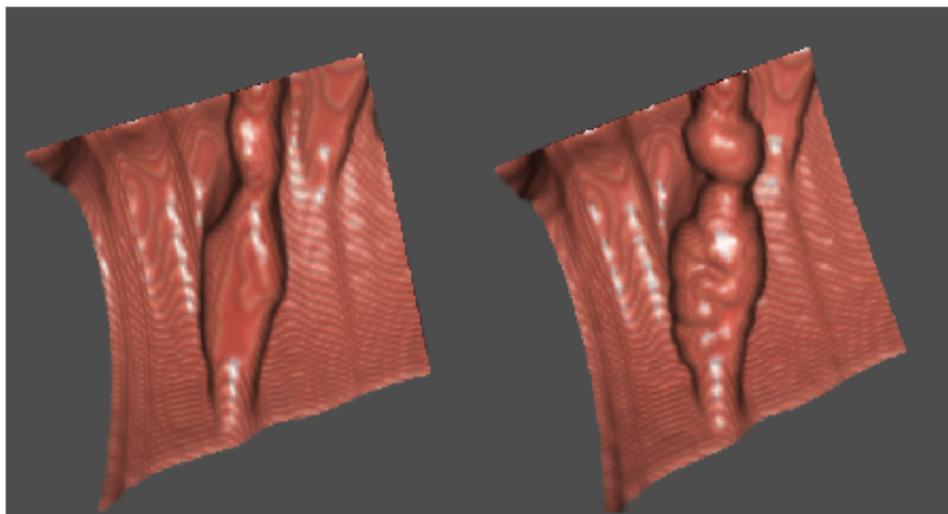
4.5. Comparison of JCF and VMCSF

We gave the evaluation and the results of the PELS algorithm using Joint Curvature Flow (JCF) in Section 4.3.1 and using Volume Maximizing Constant Surface Flow (VMCSF) in Section 4.4.1. In each section we compared the colon wall enhanced by the proposed flow with the original colon wall in terms of the performance of SNO algorithm. In this section we provide a similar comparison between the two proposed flows.

Comparing the performances of proposed flows on "clinically significant" polyps, we see that JCF does a better job in increasing SNO scores. For these polyps the intersection of surface normals is increased more by sinking polyp necks than increasing the spherical symmetry of polyp apices. Figure 4.11 shows the comparison of SNO scores for polyps obtained after enhancing with JCF and VMCSF. We observe that most polyps larger than 9 mm are above the $y = x$ line. On the other hand, most



(a)



(b)

Figure 4.10. Problematic structures VMCSF is having trouble with: (a) PELS with VMCSF introduces this false positive and (b) PELS with VMCSF increases the SNO score for this false positive

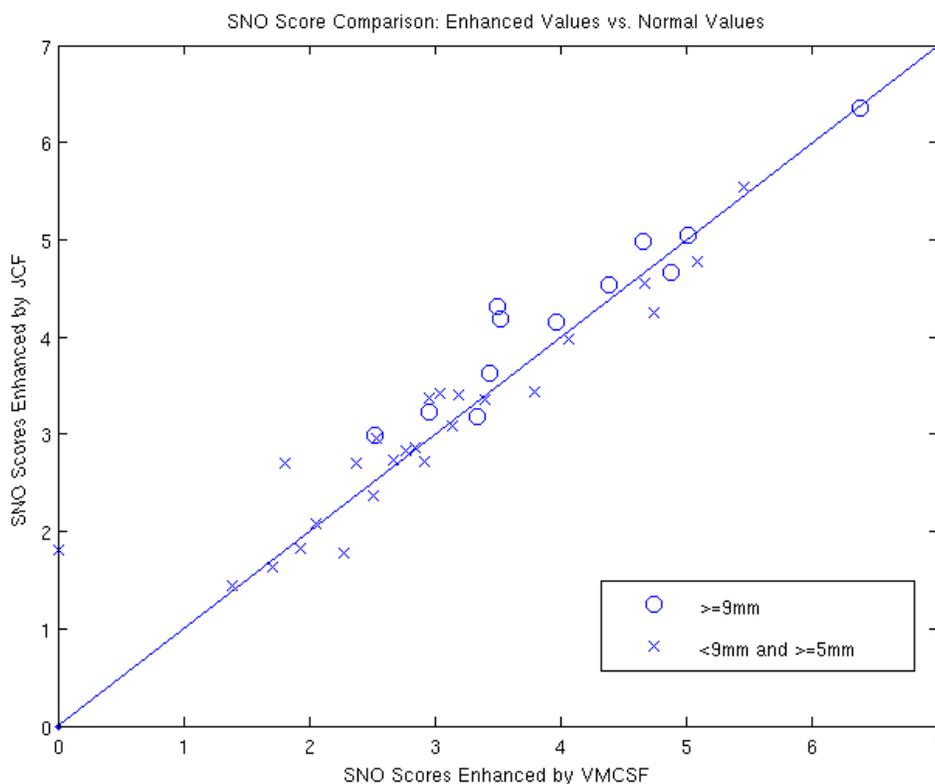


Figure 4.11. Graph of SNO scores of polyps enhanced with JCF vs. polyps enhanced with VMCSF

small polyps ($5 - 9\text{ mm}$) are below the $y = x$ line showing that VMCSF is better in increasing SNO scores of these polyps. For small polyps sinking the neck region reduces the volume of the polyp, while increasing the spherical symmetry of the apex part enlarges the polyp. The enlargement of the volume increases the number of intersecting surface normals more.

The FROC analysis on these flows have shown that for polyps larger than 9 mm , JCF reduces the number of false positives for the same sensitivity level, while VMCSF does not change the number of false positives so much. This observation is consistent with the observations on SNO scores for these polyps. In Figure 4.12 we plot the FROC curves of results of both flows for "clinically significant" polyps. We see that the number of false positives detected on the colon enhanced by JCF is almost all the time smaller at the same sensitivity level. There is only one point for which VMCSF yields a smaller number of FPs.

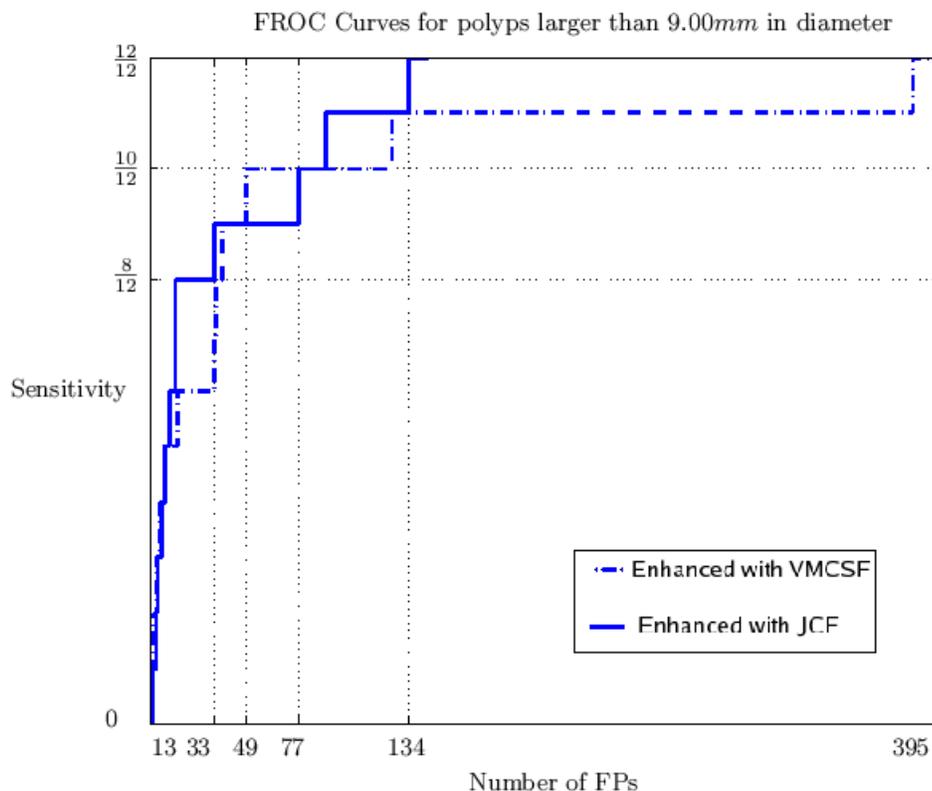


Figure 4.12. FROC curves for polyps larger than 9.00 mm

When we include the small polyps in the FROC analysis we notice that the difference seen in Figure 4.13 is reduced. Although JCF performs better for large polyps, VMCSF performs better for small polyps and the solid and the dashed lines meet each other at some point. There are some sensitivity levels for which VMCSF yields less FPs and there are some JCF yields less FPs. Comparing these two flows we can say that the JCF increases the performance of SNO algorithm more than the VMCSF does. Although for small polyps they are comparable, for "clinically significant" polyps JCF is far better. As a result we can say that PELS algorithm that uses the JCF is more preferable than the PELS algorithm using VMCSF.

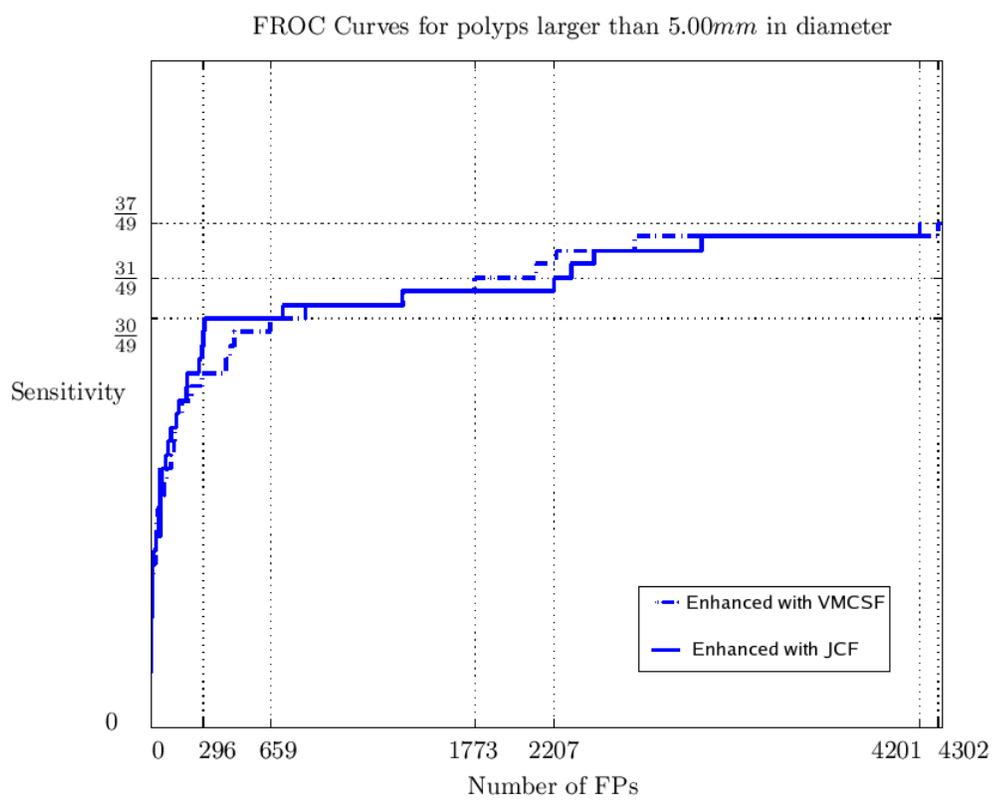


Figure 4.13. FROC curves for polyps larger than 5.00 mm

5. CONCLUSIONS

In this thesis, we have proposed a polyp enhancement scheme, based on the well known level set formulation. The aim of this algorithm is to evolve the whole colon wall using level set methods so to enhance geometric shapes colonic polyps and increase the performance of existing CTC-CAD algorithms. In order to build polyp enhancing level set methods (PELS) we proposed two different speed functions created by different approaches. We have used the geometric approach to create the Joint Curvature Flow (JCF). This flow was based on geometric observations on the real patient colon wall. Volume Maximizing Constant Surface Flow (VMCSF) was created using a variational approach, for which we started from the general question: How to increase the spherical symmetry of a surface? We have used the Surface Normal Overlap (SNO) algorithm in order to evaluate the PELS with these two flows. We have seen that both flows increased the performance of the SNO algorithm to some extent. JCF performed better in the case of "clinically significant" ($\geq 9\text{ mm}$) polyps. For smaller polyps VMCSF performs slightly better than the JCF. As a result because "clinically significant" polyps are more important JCF is preferred over VMCSF to be used with PELS.

In the evaluation of the PELS we have only used SNO as the CAD algorithm, which was designed to detect polyps of size larger than 9 mm in diameter. For a more accurate assessment of PELS we need to evaluate it with several other CAD algorithms designed to detect polyps of any size. The dataset used in evaluation consisted of 9 patients containing 12 polyps bigger than 9 mm , and 37 polyps of size between $5 - 9\text{ mm}$. Evaluating the PELS algorithm with a bigger dataset would provide us a better insight about its performance.

Along with the approaches mentioned above, as a future work we aim to combine geometric and the variational approach in one flow. JCF and VMCSF perform better at different sizes, combining these flows together we can make use of the advantages of both flows. The resulting flow would sink polyp necks while increasing spherical

symmetries of polyp apices.

One other topic for future research is to design a CAD algorithm that uses the PELS and the information coming from the level set evolution. The amount of deformation at different points of the colon wall may be used to detect colonic polyps. In addition to this an optimal CAD algorithm may be designed that makes direct use of the spherical symmetries of polyps in detecting them, which would be used with PELS for better CAD performances.

Current implementation of the algorithm is not optimized and because of large data size the speed of the algorithm is poor. It takes about 1 day to enhance a human colon and detect colonic polyps in it, which is too long. By further optimizations on the code and increased computational power the speed issues can be overcome and the processing time for a colon may be reduced to a few hours.

APPENDIX A: SIMPLIFYING EULER-LAGRANGE EQUATIONS

In Section 3.2.2.1 we have seen that the Euler-Lagrange equations for the constraint maximization problem given in Equation 3.11 is given as:

$$\begin{aligned}
 F_x - \frac{\partial}{\partial u} F_{x_u} - \frac{\partial}{\partial v} F_{x_v} &= 0 \\
 F_y - \frac{\partial}{\partial u} F_{y_u} - \frac{\partial}{\partial v} F_{y_v} &= 0 \\
 F_z - \frac{\partial}{\partial u} F_{z_u} - \frac{\partial}{\partial v} F_{z_v} &= 0
 \end{aligned} \tag{A.1}$$

where F is given as:

$$F = \nabla w \cdot (S_u \times S_v) - \lambda |S_u \times S_v| \tag{A.2}$$

for a Lagrange multiplier λ . Since w is twice continuously differentiable and S is differentiable, F is also twice continuously differentiable. Before we begin to simplify the Euler-Lagrange equations let us review some terms coming from differential geometry [38]. For a regular surface parameterized by $S(u, v)$, where S is differentiable (have continuous partial derivatives of all orders), the first fundamental form I and the second fundamental form II are defined as:

$$\begin{aligned}
 E_I &= S_u \cdot S_u \\
 F_I &= S_u \cdot S_v \\
 G_I &= S_v \cdot S_v \\
 e_{II} &= N \cdot S_{uu} \\
 f_{II} &= N \cdot S_{uv} = N \cdot S_{vu} \\
 g_{II} &= N \cdot S_{vv}
 \end{aligned}$$

where \cdot is dot product, subscripts I and II denote the first and second fundamental forms respectively, and other subscripts denotes partial derivatives. Using these terms let us define a_{11} and a_{22} as:

$$\begin{aligned} a_{11} &= \frac{f_{II}F_I - e_{II}E_I}{E_I G_I - F_I^2} \\ a_{22} &= \frac{f_{II}F_I - g_{II}E_I}{E_I G_I - F_I^2} \end{aligned}$$

The derivations of all three equations seen in A.1 are similar, thus we demonstrate this derivation only for the first one. Taking the partial derivatives of F with respect to x , x_u and x_v we obtain:

$$\begin{aligned} \frac{\partial F}{\partial x} &= \left(\frac{\partial}{\partial x} \nabla w \right) \cdot (S_u \times S_v) \\ \frac{\partial F}{\partial x_u} &= -(w_y z_v - w_z y_v) - \lambda \vec{N} \cdot [(1, 0, 0) \times S_v] \\ \frac{\partial F}{\partial x_v} &= (w_y z_u - w_z y_u) - \lambda \vec{N} \cdot [S_u \times (1, 0, 0)] \end{aligned}$$

in which we have used $\vec{N} = \frac{S_u \times S_v}{|S_u \times S_v|}$. Taking further partial derivatives with respect to u and v we get the following:

$$\begin{aligned} \frac{\partial}{\partial u} \frac{\partial F}{\partial x_u} &= - \left(\frac{\partial}{\partial y} \nabla w \cdot S_u \right) z_v - w_y z_{vu} + \left(\frac{\partial}{\partial z} \nabla w \cdot S_u \right) y_v + w_z y_{vu} \\ &\quad - \lambda \{ \vec{N}_u \cdot [(1, 0, 0) \times S_v] + \vec{N} \cdot [(1, 0, 0) \times S_{vu}] \} \\ \frac{\partial}{\partial v} \frac{\partial F}{\partial x_v} &= - \left(\frac{\partial}{\partial z} \nabla w \cdot S_v \right) y_u - w_z y_{uv} + \left(\frac{\partial}{\partial y} \nabla w \cdot S_v \right) z_u + w_y z_{uv} \\ &\quad - \lambda \{ \vec{N}_v \cdot [S_u \times (1, 0, 0)] + \vec{N} \cdot [S_{uv} \times (1, 0, 0)] \} \end{aligned}$$

Combining the terms F_x , $\frac{\partial}{\partial u} F_{x_u}$ and $\frac{\partial}{\partial v} F_{x_v}$ we write the Euler-Lagrange equation as:

$$\begin{aligned} \frac{\partial F}{\partial x} - \frac{\partial}{\partial u} \frac{\partial F}{\partial x_u} - \frac{\partial}{\partial v} \frac{\partial F}{\partial x_v} &= \frac{\partial}{\partial x} \nabla w \cdot (S_u \times S_v) + \frac{\partial}{\partial y} \nabla w \cdot (S_u z_v - S_v z_u) \\ &\quad + \frac{\partial}{\partial z} \nabla w \cdot (S_v y_u - S_u y_v) - \lambda(a_{11} + a_{22})[y_v z_u - y_u z_v] = 0 \end{aligned}$$

where we have used the fact that partial derivatives of \vec{N} reside in the tangent plane of the surface parameterized by S , in other words $\vec{N}_u = a_{11}S_u + a_{21}S_v$ and $\vec{N}_v = a_{12}S_u + a_{22}S_v$, [38]. After cancellations and collecting the alike terms we get:

$$(w_{xx} + w_{yy} + w_{zz})[z_v y_u - z_u y_v] - \lambda(a_{11} + a_{22})[y_v z_u - y_u z_v] = 0$$

Noticing the laplacian operator and using the fact that $H = \frac{a_{11}+a_{22}}{2}$, where H is the mean curvature of the surface we get the final result for the first equation in A.1 to be:

$$(\Delta w + 2\lambda H)[z_v y_u - z_u y_v] = 0 \tag{A.3}$$

Similar manipulations on the second and the third equations of A.1 yield the following equations:

$$(\Delta w + 2\lambda H)[z_u x_v - z_v x_u] = 0 \tag{A.4}$$

$$(\Delta w + 2\lambda H)[x_u y_v - x_v y_u] = 0$$

Combining the A.3 and A.4, noticing the cross product and placing $\Delta w = -1$ we get the final result as given in Equation A.5.

$$-[1 - 2\lambda H](S_u \times S_v) = 0 \tag{A.5}$$

REFERENCES

1. Komaromy, M., *Colon Cancer: What is the Colon Cancer?*, http://www.genetichealth.com/CRC_Colon_Cancer_What_is_Colon_Cancer.shtml, 2000.
2. Coin, C. G., F. C. Wollet, J. T. Coin, M. Rowland, R. K. Deramos and R. Dandrea, "Computerized Radiology of the Colon: A Potential Screening Technique", *Comput. Radiology*, Vol. 7, No. 4, pp. 215-221, 1983.
3. Vining, D. J., "Virtual Colonoscopy", *Gastrointest. Endosc. Clin. N. Am.*, Vol. 7, No. 2, pp. 285-291, 1997.
4. Lee, T. Y., P. H. Lin, C. H. Lin, Y. N. Sun and X. Z. Lin, "Interactive 3-D Virtual Colonoscopy System", *IEEE Trans. Inf. Tech. Biomed.*, Vol. 3, No. 2, pp. 139-150, 1999.
5. Beaulieu, C. F., R. B. Jeffrey, C. Karadi, D. S. Paik and S. Napel, "Display Modes for CT Colonography, Part II. Blinded Comparison of Axial CT and Virtual Endoscopic and Panoramic Endoscopic Volume-Rendered Studies", *Radiology*, Vol. 212, No. 1, pp. 203-212, 1999.
6. Wang, G., E. G. McFarland, B. P. Brown and M. W. Vanier, "GI Tract Unraveling with Curved Cross Sections", *IEEE Trans. Med. Imaging*, Vol. 17, No. 2, pp. 318-322, 1998.
7. Haker, S., S. Angenent, A. Tannenbaum and R. Kikinis, "Nondistorting Flattening Maps and the 3D Visualization of Colon CT Images", *IEEE Trans. Med. Imaging*, Vol. 19, pp. 665-670, 2000.
8. Summers, R. M., "Challenges for Computer-Aided Diagnosis for CT Colonography", *Abdominal Imaging*, Vol. 27, pp. 268-274, 2002.

9. Summers, R. M., C. F. Beaulieu, L.M. Pusanik, J. D. Malley, R. B. Jeffrey, D. I. Glazer and S. Napel, "Automated Polyp Detector for CT Colonography: Feasibility Study", *Radiology*, Vol. 216, pp. 284-290, 2000.
10. Summers, R. M., C. D. Johnson, L. M. Pusanik, J. D. Malley, A. M. Youssef and J. E. Reed, "Automated Polyp Detection at CT Colonography: Feasibility Assessment in a Human Population", *Radiology*, Vol. 219, pp. 51-59, 2001.
11. Jerebko, A. K., S. B. Teerlink, M. Franaszek and R. M. Summers, "Polyp Segmentation Method for CT Colonography Computer Aided Detection", *Proceedings of SPIE*, Vol. 5031, 2003.
12. Jerebko, A. K., J. D. Malley, M. Franaszek and R. M. Summers, "Support Vector Machines Committee Classification Method for Computer-aided Polyp Detection in CT Colonography", *Computer-Assisted Radiology and Surgery*, Vol. 12, pp. 479-486, 2005.
13. Kiss, G., J. Van Cleynenbreugel, M. Thomeer, P. Suetens and G. Marchal, "Computer-aided Diagnosis in Virtual Colonography via Combination of Surface Normal and Sphere Fitting Methods", *European Radiology*, Vol. 12, pp. 77-81, 2002.
14. Yoshida, H. and J. Nappi, "Three-dimensional Computer-aided Diagnosis Scheme for Detection of Colonic Polyps", *IEEE Trans. on Med. Imag.*, Vol. 20, pp. 1261-1274, 2001.
15. Yoshida, H., Y. Masutani, P. MacEneaney, D. T. Rubin and A. H. Dachman, "Computerized Detection of Colonic Polyps at CT Colonography on the Basis of Volumetric Features: Pilot Study", *Radiology*, Vol. 222, pp. 327-336, 2002.
16. Paik, D. S., C. F. Beaulieu, G. D. Rubin, B. Acar, R. B. Jeffrey, J. Yee, J. Dey and S. Napel, "Surface Normal Overlap: A Computer-Aided Detection Algorithm With Application to Colonic Polyps and Lung Nodules in Helical CT", *IEEE Trans. on*

- Medical Imaging*, Vol. 23, No.6, pp. 661-675, 2004.
17. Gokturk, S. B., C. Tomasi, B. Acar, C. F. Beaulieu, D. S. Paik, R. B. Jeffrey, J. Yee and S. Napel, "A Statistical 3-D Pattern Processing Method for Computer-aided Detection of Polyps in CT Colonography", *IEEE Trans. on Med. Imag.*, Vol. 20, pp. 1251-1260, 2001.
 18. Acar, B., C. F. Beaulieu, S. B. Gokturk, C. Tomasi, D. S. Paik, R. B. Jeffrey, J. Yee and S. Napel, "Edge Displacement Field-based Classification for Improved Detection of Polyps in CT Colonography", *IEEE Trans. on Med. Imag.*, Vol. 21, pp. 1461-1467, 2002.
 19. Yao, J., M. Miller, M. Franaszek and R. M. Summers, "Colonic Polyp Segmentation in CT Colonography-Based on Fuzzy Clustering and Deformable Models", *IEEE Trans. on Med. Imag.*, Vol. 23, No. 11, 2004.
 20. Konukoğlu, E., B. Acar, D. S. Paik, C. F. Beaulieu and S. Napel, "Heat Diffusion Based Detection of Colonic Polyps in CT Colonography", *to appear in the Proceedings of EUSIPCO 2005*, Antalya, Turkey, 2005.
 21. Chakraborty, D. P., "Maximum Likelihood Analysis of Free-response Receiver Operating Characteristics (FROC) Data", *Medical Physics*, Vol. 16, No. 4, pp. 561-568, 1989.
 22. Kass, M., "Snakes: Active Contour Models", *Int. J. Comput. Vision*, Vol. 1, pp. 312-333, 1988.
 23. Cohen, L. D., "On Active Contour Models and Balloons", *CVGIP: Image Understanding*, Vol. 53, pp. 211-218, 1991.
 24. Xu, C. and J. L. Prince, "Snakes, Shapes and Gradient Vector Flow", *IEEE Trans. Imag. Proc.*, Vol. 7, No. 3, pp. 359-369, 1998.
 25. Malladi, R., J. A. Sethian and B. C. Vemuri, "Shape Modeling with Front Prop-

- agation: A Level Set Approach”, *IEEE Trans. Patt. Anal. Mach. Intel.*, Vol. 17, pp. 158-175, 1995.
26. Caselles, V., F. Catta, T. Coll and F. Dibos, ”A Geometric Model for Active Contours in Image Processing”, *Numerische Mathematik*, Vol. 66, pp. 1-31, 1993.
27. Caselles, V., R. Kimmel, G. Sapiro, ”Geodesic Active Contours”, *Int. J. Comput. Vision*, Vol. 22, pp. 61-79, 1997.
28. McInerney, T. and D. Terzopoulos, ”Deformable Models in Medical Image Analysis: A Survey”, *Medical Image Analysis*, Vol. 1, No. 2, pp. 91-108, 1996.
29. Han, X., C. Xu and J. L. Prince, ”A Topology Preserving Level Set Method for Geometric Deformable Models”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 6, 2003.
30. Sethian, J. A., *Level Set Methods and Fast Marching Methods*, Cambridge University Press, New York, 1999.
31. Osher, S. and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, Applied Mathematical Sciences, Vol. 153, New York, 2003.
32. Osher, S. and J. A. Sethian, ”Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations”, *J. Comput. Phys.*, Vol. 79, pp. 12-49, 1988.
33. LeVeque, R. J., *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, Berlin, 1992.
34. Weickert, J., ”A Review of Nonlinear Diffusion Filtering”, *Lecture Notes in Computer Science: Scale-Space Theory in Computer Vision*, Springer, Berlin, pp. 3-28, 1997.
35. Weicker, J., B. M. T. H. Romeny and M. A. Viergever, ”Efficient and Reliable

- Schemes for Nonlinear Diffusion Filtering”, *IEEE Trans. on Image Processing*, Vol. 7, No. 3, pp. 398-410, 1998.
36. Strauss, W. A., *Partial Differential Equations An Introduction*, Wiley & Sons, New York, 1992.
37. Giles, B., ”Simple Points, Topological Numbers and Geodesic Neighborhoods in Cubic Grids”, *Pattern Recognition Letters*, Vol. 15, pp. 1003-1011, 1994.
38. M. DoCarmo, *Differential Geometry of Curve and Surfaces*, Prentice-Hall ,1976.
39. Monga, O. and S. Benayoun, ”Using Partial Derivatives of 3D Images to Extract Surface Features”, *Comp. Vision and Image Underst.*, Vol. 61, No. 2, pp. 171-189, 1995.
40. Aubert, G., M. Barlaud, O. Faugeras and S. Jehan-Besson, ”Image Segmentation using Active Contours: Calculus of Variations or Shape Gradients?”, *SIAM J. Appl. Math.*, Vol. 63, No. 6, pp. 2128-2154, 2003.
41. Van Brunt, B., *The Calculus of Variations*, Springer-Verlag, New York, 2004.
42. Gelfand, I. M., S. V. Fomin, *Calculus of Variations*, Prentice-Hall, 1963.