

Scale-based Decomposable Shape Representations for Probabilistic Medical Imaging Segmentation

A Thesis Proposal
Presented to
The Academic Faculty

by

Delphine Nain

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
January 2005

Scale-based Decomposable Shape Representations for Probabilistic Medical Imaging Segmentation

Approved by:

Professor Aaron Bobick, Adviser

Professor Steven Haker

Professor Allen Tannenbaum, Adviser

Professor Eric. W. L. Grimson

Professor Greg Turk

Date Approved _____

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
SUMMARY	1
0.1 Roadmap	4
I INTRODUCTION	5
1.1 Challenges addressed by this Thesis	8
1.2 Contribution of this Thesis	10
II BACKGROUND: SHAPE REPRESENTATION AND DEFORMABLE MODELS FOR MEDICAL IMAGE SEGMENTATION	12
2.1 Overview	12
2.2 Object Representation	12
2.2.1 2D Parametric Object representation	12
2.2.2 3D Parametric Object representation	15
2.2.3 Adding Data-Driven Shape Priors to Parametric Models	22
2.2.4 Implicit Object Representation	25
2.2.5 Shape Priors for Implicit Representations	26
2.3 Parametric Deformable Models	27
2.3.1 Snakes	27
2.3.2 Snakes Implementation	30
2.3.3 Snakes Limitations	31
2.3.4 Parametric Deformable Surfaces	31
2.3.5 Discussion	32
2.4 Geometric Deformable Models	32
2.4.1 Geometric Active Contours	33
2.4.2 Conformal (Geodesic) Active Contours	35
2.4.3 Conformal (Geodesic) Active Contours with Inflationary Term	36
2.4.4 Region-Based Active Contours	36
2.4.5 Shape Priors for Geometric Active Contours	37

2.4.6	Summary	38
III	SOFT SHAPE PRIORS: VESSEL SEGMENTATION USING A SHAPE DRIVEN FLOW	40
3.1	Blood Vessel Segmentation	40
3.2	Related Work	41
3.3	Shape Driven Flow	43
3.3.1	Region Based Flow	43
3.3.2	Shape Filters	44
3.3.3	Curve Evolution using Local Filters	46
3.4	Implementation	47
3.4.1	Numerical Methods	47
3.4.2	Optimizations	48
3.5	Results	49
3.5.1	2D Images	49
3.5.2	3D Images	52
3.6	Conclusion	53
IV	MULTI-SCALE 3D SHAPE ANALYSIS USING SPHERICAL WAVELETS	54
4.1	Biological Driver: NAMIC grant	55
4.1.1	Description of Dataset	55
4.1.2	Prostate	56
4.1.3	Caudate	57
4.2	Shape Representation	57
4.2.1	Shape Registration	57
4.2.2	Statistical Modeling	60
4.2.3	Hierarchical Representation for 3D Shapes	64
4.3	Spherical wavelets	65
4.3.1	Grid Structure for Spherical Wavelet Construction	66
4.3.2	Scaling Function	67
4.3.3	Wavelet Function	68
4.4	Shape Representation using spherical wavelets	70
4.4.1	Spectrum Analysis	71

4.5	Scale-Space Spherical Wavelet Prior for Statistical Population Analysis	73
4.5.1	Adaptive Band Selection by scale-space decomposition	74
4.5.2	Decomposition Results	75
4.5.3	Band Visualization	76
4.5.4	Building the Prior	78
4.6	Experiments	78
4.6.1	Shapes Priors used in the experiment	78
4.6.2	Test Shape Reconstruction	80
4.6.3	Results	81
4.7	Conclusions	82
V	PROPOSED WORK	87
5.1	Segmentation of medical imagery using spherical wavelets	87
5.1.1	Pose parameters	87
5.1.2	Surface parameters using spherical wavelets	88
5.1.3	Proposed Surface Evolution Derivation: case with no prior	89
5.1.4	Proposed Experiments for surface evolution with no shape prior	90
5.1.5	Proposed Surface Evolution Derivation: case <i>with</i> prior	91
5.1.6	Proposed Experiments for surface evolution with shape prior	92
5.2	Shape classification using spherical wavelets	94
APPENDIX A — APPENDIX	96	
REFERENCES	101	

LIST OF TABLES

- | | | |
|---|---|----|
| 1 | Properties of parametric and geometric active contours. Adapted from [14] | 39 |
|---|---|----|

LIST OF FIGURES

1 Three standard orientations shown with MRI brain slices and a 3D transparent model of the human head	5
2 To visualize the three-dimensional information in an MRI volume, the physician can visualize an ordered sequence of slices. In this figure, sagittal slices are displayed	6
3 Segmented White Matter from a MRI Brain Scan. The top window shows the three-dimensional model superimposed on a coronal slice. The output of a segmentation is also labelmap where pixels that belong to the interior or the boundary of structure are labeled in pink in the bottom three windows (respectively axial, sagittal and coronal views)	6
4 Level Sets of an embedding function ϕ for a closed curve in \mathbb{R}^2	13
5 Visualization of the spherical harmonic functions on the sphere. A darker color shows where the function has support. The figure shows the real parts of the spherical harmonic function Y_l^m , with l growing from 0 (top) to 5 (bottom), and m ranging from 0 (left) to l in each row (from [19])	19
6 Visualization of the spherical harmonic correspondence. A first order ellipsoid and six left ventricles are displayed. The surface net shows the (θ_i, ϕ_i) parametrization (same parameters = same homologous points). The ridges on the first order ellipsoid are the equator and $0, \pi/2, \pi, 3\pi/2$ meridian lines in all objects. The equator and meridian lines are emphasized in different colors. The poles are at the crossing of the meridian lines (from [19])	19
7 Visualization of a lateral ventricle (side view) at different spherical harmonics degrees; $m=1,4,8,12$ top to bottom (from [19])	20
8 Outline of several brain structures in a single model labeled with landmarks (from [11])	23
9 In ASM, the learned shape variation can be shown by selecting one of the eigenvectors (modes of variation) and varying its magnitude from $-n\sqrt{\sigma}$ to $n\sqrt{\sigma}$ with $n = 2$ and adding the result to the mean shape to display the variation (from [11])	24
10 Level Sets of an embedding function ϕ for a closed curve in \mathbb{R}^2	25
11 An example of a snake active contour (a) Intensity CT image of the heart left ventricle (b) edge detected image (c) initial contour (d) deformable contour moving toward the left ventricle boundary, driven by an inflating pressure force (from [32])	28
12 Bone segmentation in CT image with splitting shrinking conformal active contour.	34

13	Four steps in the segmentation of two different corpora callosa. The last image in each case shows the final segmentation in red. The cyan contour is the standard evolution without the shape influence (from [28])	37
14	A Region Based Geometric Active Contour is prone to leak if the image term is not reliable	44
15	(a) $\epsilon_1(\mathbf{x}, r)$ is the intersection of the ball of radius r centered at \mathbf{x} and the region R inside the contour. (b) Points inside the widening region have a higher ϵ_1 measure but most points on the contour have the same measure. (c) Contour points close to the widening of the contour have a higher ϵ_2 measure	45
16	Dilation Flow with Shape Prior	50
17	Flow with Shape Prior on 2D Synthetic Images	50
18	Vessel Flow on Angiogram Images	51
19	Vessel Flow on Angiogram Images	51
20	Different Flows on the first CT Coronary Data	52
21	Base Flow with Shape Prior on the second CT coronary dataset	53
22	Example of 10 shapes from the Prostate dataset	56
23	Example of 10 shapes from the Caudate dataset	56
24	Haker et. al. registration algorithm: shapes are conformally mapped to the sphere using 6 landmark points to constrain the mapping. The sphere is then re-triangulated, and the coordinates of the original surface S are interpolated at the vertices of the new triangulation, producing a final shape with the new triangulation.	58
25	Shape 1 and 2 from Figure 24 realigned with the Procrustes technique . . .	58
26	Two prostate surfaces. The surfaces have been mapped to the sphere and re-triangulated in the same way to establish point correspondence. The small spheres represent landmark points which were matched during the registration process.	60
27	Shape Model	61
28	A contour partitioned into smaller segments	62
29	Scale Space decomposition. The wavelet transform is iterated for the low pass branches (LP). The outputs of the high-pass branches (marked as HP in the figure) are divided into a number of bands. The number of bands for each high-pass branch is determined by the level at which the high-pass filtering is performed. In this way, for a p level wavelet decomposition the frequency domain is divided into $B = 2^p$ bands. The space-frequency domain partition of the signal is shown on the right.	62

30	A wavelet transform is applied to the parametric functions representing a deformable contour. The resulting wavelet coefficients are then grouped into 64 bands, and the joint distribution of each band is estimated from the available training samples, via its mean and covariance matrix. This effectively transforms the covariance matrix of the full joint distribution into a matrix that is close to, but not necessarily exactly, a block diagonal matrix. The submatrix corresponding to band B reflects global shape characteristics, whereas the submatrix corresponding to B reflects local shape characteristics at a particular segment	63
31	Visualization of Scaling and Wavelet Basis Functions. The color corresponds to the value of the function at each point on the sphere. Notice that the support of each basis gets smaller as the resolution increases	66
32	Recursive Partitioning of an octohedron: successive levels of the triangulation are obtained by subdividing triangles into 4 children triangles	67
33	Refinement of the surface grid by subdivision of the edges	67
34	Multi-Scale decomposition of a function using the wavelet transform. The original signal $f(x)$ is decomposed into a series of high-pass and low-pass coefficients. The scaling coefficient represent the low pass information of the original signal. The intermediate wavelet coefficients represent band pass information, and the highest resolution wavelet coefficient γ_3 represents the high pass information.	69
35	Shape at various resolution levels, see text for details	70
36	Covariances Matrices, see text for detail	75
37	Band Decomposition: various bands $B_{j,i}$, where j is the resolution and i is the band number, shown in Anterior view (A) and Posterior view (P), see text for color	76
38	Band Decomposition, Caudate Nucleus dataset: various bands shown for scale 2 (shown in Anterior view and Posterior view), see text for color	77
39	[Prostate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the Normal Test Shapes	83
40	[Caudate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the Normal Test Shapes	83
41	[Prostate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the Noise Test Shapes	84
42	[Caudate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the Noise Test Shapes	84

43	Prostate dataset: Ground Truth (GT), Noisy Shape (Noise), and reconstruction with PCA and Wavelet shape prior with 5 training samples when both the GT is projected onto the prior and the Noise is projected onto the prior. Color is error from blue (lowest) to red.	85
44	Caudate dataset: Ground Truth (GT), Noisy Shape (Noise), and reconstruction with PCA and Wavelet shape prior with 25 training samples when both the GT is projected onto the prior and the Noise is projected onto the prior. Color is error from blue (lowest) to red.	86

SUMMARY

In this thesis, we propose and evaluate two novel scale-based decomposable representations of shape for the segmentation and shape analysis of anatomical structures in medical imaging. We propose two representations that are each adapted to a particular class of anatomical structures and allow for a richer description and a more granular control over the deformation of models based on these representations, when compared to previous techniques. In particular, the decomposition of these shape representations can be localized both in space and scale, enabling the construction of more descriptive, non-global, non-uniform shape priors to be included in the segmentation framework. For each representation, we derive a segmentation algorithm using the parameters of the shape representation to easily include and benefit from the prior in the optimization framework.

In this proposal, we first review the two main classes of shape representation, parametric and implicit models, and discuss the impact of existing representations on the construction of data-driven or knowledge-driven shape priors and segmentation algorithms. In particular, we note that most shape priors fall either under the category of global priors, constraining the full shape model to a predefined shape space, or very local priors, constraining the smoothness of the model on a very local level. This thesis addresses the gap in between the two categories by proposing and evaluating multi-scale priors adapted to particular classes of anatomical shapes.

We then present two novel shape representations that allow for finer granularity and richness for shape analysis from which more accurate shape priors and segmentation algorithms can be derived. We present our existing work:

- A novel knowledge-driven semi-local prior based on local shape filters that measure shape properties for implicit representations of shape. We use these filters for the segmentation of blood vessels, and introduce the notion of segmentation with a *soft*

shape prior, where the segmented model is not globally constrained to a predefined shape space, but is penalized locally if it deviates strongly from a tubular structure. We introduce the concept of a scale-space shape filter that measures the deviation from a tubular shape in a local neighborhood of points, given a particular scale of analysis. Using this filter, we derive a region-based active contour segmentation algorithm for tubular structures that penalizes leakages. We present results on synthetic and real 2D and 3D datasets.

- A novel multi-scale shape representation using spherical wavelets for parametric models from which a data-driven prior can be learned to segment shapes with fine variations, such as cortical structures. This novel representation is motivated by the observation that the classical active shape models prior (ASM) technique cannot accurately encode fine, localized shape variations when the training set is of moderate size, since the relatively few eigenvectors will represent the most global modes of variation in the shapes. To address this issue, we present a novel algorithm that learns shape variations from data at *multiple scales and locations* using the spherical wavelet representation and spectral graph partitioning. Our results show that for a given training set size, our algorithm significantly improves the approximation of shapes in a testing set over ASM.

Based on this work, we propose the following three remaining research topics:

1. Segmentation of medical imagery using the spherical wavelet shape prior. We propose to derive an active surface evolution using the spherical wavelet coefficients as parameters of the segmentation energy to be minimized. The advantage of such a segmentation framework is that the spherical wavelet shape prior can directly be used to constrain the parameters during the surface evolution. We propose a set of experiments to test our surface evolution, both on synthetic and real data.
2. Testing the robustness of the shape registration and its influence on the segmentation results. Prior to building the spherical wavelet shape prior, it is necessary

to align and remesh all structures in the training set. We propose to test the sensitivity of the segmentation results to noise in the registration and remeshing process for the two methods we currently use.

3. Shape classification using spherical wavelets and non-parametric permutation testing. Using spherical wavelets coefficients as features to describe a population of shapes, we propose to use the existing non-parametric permutation testing technique to test for shape differences in the caudate brain structure among two population of patients with and without schizotopal disorder.

0.1 Roadmap

In **chapter 1**, we present an overview of existing shape representation for anatomical structures in medical imaging and their impact on the construction of data-driven or knowledge-driven shape priors and deformable model segmentation algorithms. We then present the main contributions of this thesis, as well as a summary of the proposed work.

In **chapter 2** we present in greater detail existing work for the topics of shape representation, shape priors and deformable models for medical image segmentation. We present two main shape representations: parametric models and geometric models, for both curve in \mathbb{R}^2 and surfaces in \mathbb{R}^3 . For parametric models, we focus in particular on shape representation using basis functions. We then discuss the types of knowledge-driven and data-driven shape priors that can be learned from existing shape representations. Finally, we give an overview of the theory of segmentation using active contours and active surfaces, both for the parametric and geometric models.

In **chapter 3**, we present a segmentation method for blood vessels using an implicit deformable model with a knowledge-driven localized shape prior. We combine image statistics and scale-space shape information in a variational framework to derive a region-based active contour that segments tubular structures and penalizes leakages. We present results on synthetic and real 2D and 3D datasets.

In **chapter 4**, we present a novel algorithm that learns the shape variation at *multiple scales and locations* from a training set. Our technique uses spherical wavelets to generate a multi-scale description of surfaces and spectral graph partitioning to adaptively discover independent shape variations at multiple scales. We compare our technique to the standard ASM technique and present results on 3D brain and prostate datasets.

In **chapter 5** we detail our proposed work for the completion of this dissertation.

CHAPTER I

INTRODUCTION

Medical Imaging has rapidly become a useful tool for physicians to peek non invasively in the human body and diagnose, treat and track the progress of a disease. The goal of the medical imaging community is to assist physicians by extracting, with the assistance of computers, clinically useful information about anatomical structures digitally imaged through CT, MRI and other modalities in an efficient, repeatable and accurate manner [3, 32].

A medical scanner often outputs a three-dimensional volume that represents a portion of the human body. The volume can then be visualized as a sequence of 2D images, by “slicing” the volume along a chosen direction. Standard medical directions are Axial, Coronal and Sagittal, shown in Figure 1. Figure 2 shows a typical sequence through an MRI volume of a human head in a sagittal direction. The size of the volume is $256 \times 256 \times 124$ pixels. By scrolling through the slices, physicians can create a mental 3D reconstruction of the body part being imaged. Often, however, it would be useful to be able to visualize particular structures directly in three dimensions to be able to directly observe the shape of the structure, as shown in Figure 3, a process known as segmentation. The figure shows segmented white matter from a MRI Brain Scan. The top window shows a three-dimensional surface

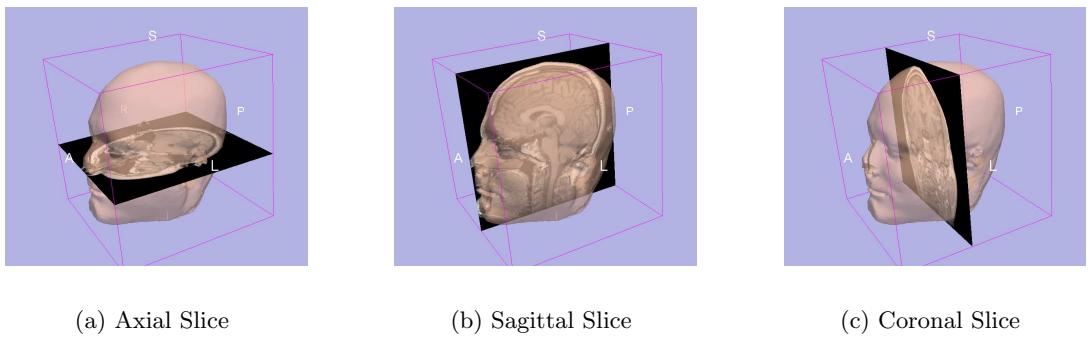


Figure 1: Three standard orientations shown with MRI brain slices and a 3D transparent model of the human head

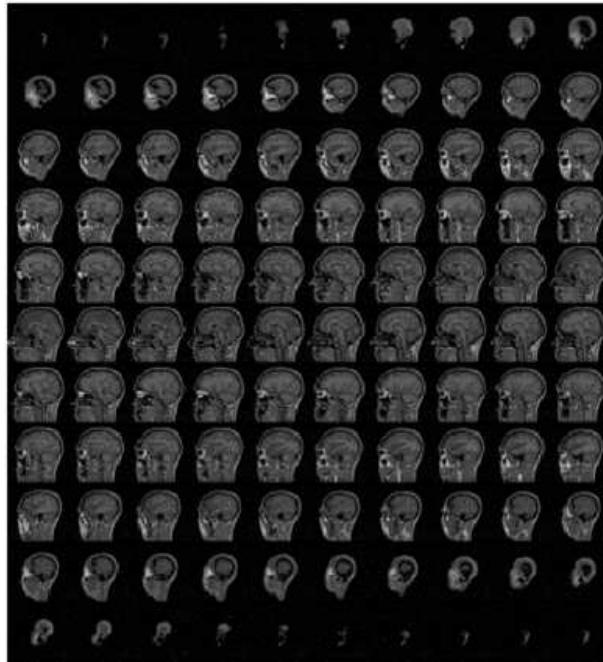


Figure 2: To visualize the three-dimensional information in an MRI volume, the physician can visualize an ordered sequence of slices. In this figure, sagittal slices are displayed

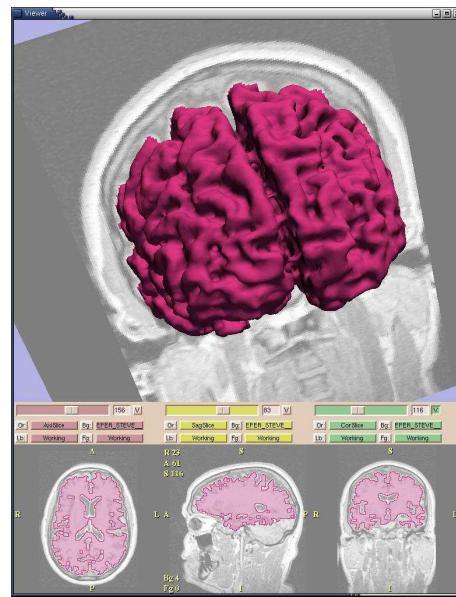


Figure 3: Segmented White Matter from a MRI Brain Scan. The top window shows the three-dimensional model superimposed on a coronal slice. The output of a segmentation is also labelmap where pixels that belong to the interior or the boundary of structure are labeled in pink in the bottom three windows (respectively axial, sagittal and coronal views)

model of the white matter superimposed on a coronal slice. The output of a segmentation is either a surface model that represents the boundary of the segmented object, or a region model that is the set of pixels that belong both to the surface and interior of the object. The region model results in a binary classification of the original volume called a *labelmap* where pixels are classified as either belonging or not belonging to the object. In the bottom three windows of Figure 3, the labelmap is shown for three slices.

Image segmentation is one of the fundamental tasks in medical image analysis. Formally, the task of segmentation is to detect and construct a compact geometric representation of anatomic structures embedded in a 2D or 3D medical image. Segmentation is often a precursor to other medical imaging tasks. Segmented structures can be visualized, registered to other structures for comparison, measured, or put in relation with other nearby segmented structures for diagnosis [48], study of anatomical structure [20] and computer-aided surgery [4].

Traditional segmentation techniques used in computer vision such as edge detection and thresholding techniques often fail to locate the object boundary or generate invalid boundaries that must be removed in a post-processing step, due to noise and sampling artifacts in medical images, or the fact that many structures do not have a distinctive intensity range or distinctive connected edges [32].

To address these difficulties, deformable models have been extensively studied and widely used in medical image segmentation, with good results. By representing the shape of the segmented structure boundaries with a model and deforming the model to fit the data, deformable models offer robustness to both image noise and boundary gaps [55]. There are two types of deformable models, based on the object representation used for the model: parametric deformable models and geometric deformable models. **Parametric** models are compact representations that are computationally efficient. However changes in model topology during the deformation, such as splitting and merging, need to be detected and require special handling. **Geometric** models have a higher computational cost, but topological changes during the evolution do not need to be detected and are automatically handled.

In medical images, the general topology, shape, location and orientation of anatomical

structures to be segmented are either known, or can be learned prior to the segmentation task. For this reason, object segmentation with deformable models and statistical shape modeling are often combined to obtain a more robust and accurate segmentation [11, 28, 50, 55, 32]. On the one hand, object segmentation using noisy data can be used to evolve the parameters of the shape model close to a solution. On the other hand, statistical shape information acquired through estimated parameters in a statistical shape model can guide the model fitting procedure.

There exists two types of shape priors: knowledge-driven and data-driven. **Knowledge-driven** shape priors are derived from human knowledge of what a shape should “look like” and the translation of this knowledge to a constraint on the parameters of the model. One of the most commonly used knowledge-driven prior is the local smoothness constraint applied to curves or surfaces. This smoothness prior can be applied by minimizing the curvature, or second derivative, of the curve by expressing curvature in terms of the model parameters. Unless the parameters of the shape representation have a particular meaning for the shape of the object (such as the radius parameter for a sphere), it is often hard to know a-priori how to constrain the parameters of the shape representation for a particular class of objects. Therefore in practice, the smoothness prior is the only common knowledge-driven prior used both for parametric and geometric deformable models.

A **data-driven** shape prior can be used to represent a class (or population) of shapes by learning the distribution of the shape parameters from a training set. The prior represents the generic shapes of the object belonging to the class, as well as the possible shape variations within the class, while excluding variations not previously seen in the training set. The most common approach is to assume a multivariate Gaussian distribution of the model parameters and learn the mean and major modes of variation of the training set by using principal component analysis. This prior is fully global by constraining all shape parameters.

1.1 *Challenges addressed by this Thesis*

This thesis focuses on the segmentation of two classes of important anatomical shapes: blood vessel structures and cortical structures, such as the caudate nucleus. Blood vessel

segmentation and visualization of blood vessels is important for clinical tasks such as diagnosis of vascular diseases, surgery planning and blood flow simulation. Segmentation of cortical structures is important for the study and diagnosis of neurological disorders as well as surgery planning.

The particular challenges we face with such structures are:

- Vessel structures do not have a fixed topology. Vessels have many branches, and the number and location of the branches are patient specific. Therefore a geometric active contour for segmentation is desirable since the topology is not known in advance. However, using image intensity alone to deform the contour often results in leakages at areas where the image information is ambiguous, motivating the need to use a shape prior. The issue is that so far only strictly local or fully global constraints have been developed for geometric active contours. A global data-driven shape prior is not desirable since there does not exist a known a-priori shape. A very local shape prior that imposes smoothness of the contour is not constraining enough to prevent leaks.
- Cortical structures have fixed topology, therefore a parametrized active contour is desirable due to its efficiency. Again, image information alone is not reliable for brain segmentation motivating the need to incorporate a data-driven shape prior in the segmentation. However shapes contain high frequency information that varies from patient to patient therefore the shape prior needs to encode high frequency variations at specific location. Existing data-driven shape priors often favor the encoding of global (low frequency) information, particularly if it has been trained on a limited number of samples. This means that high frequency information on the surface of the shapes (such as small bumps or sharp edges) are often smoothed in the prior representation.

Most shape priors used for deformable models fall either under the category of global priors, constraining the full shape model to a predefined shape space, or very local priors, constraining the smoothness of the model on a very local level. This thesis addresses the gap in between the two categories by proposing and evaluating multi-scale priors that can

be localized both in space and scale.

The notion of multi-scale has been developed by the computer vision community to decompose image data in order to control the scale of observation, using for example quadtrees or image pyramids [49]. Similarly an object boundary can be encoded as a 2D or 3D signal that can be decomposed and analyzed at different scales. The notion of scale is coupled with the location and spatial support of a filter used to analyze the signal. At the lowest scale of analysis, the filter has global support and is applied to the whole object boundary. As the scale is increased, the spatial support of the filter is decreased so that the filter is only applied to a segment of the object boundary. This decomposition reveals different information content of the shape signal depending on the scale, information not readily available from looking at the original signal. Filters at a low scale usually analyze the global, low-frequency content of the signal (global features) while filters at the highest scale analyze the local, high frequency content (local features). This decomposition can be important since shape priors should encode features at various scales.

In this thesis, we propose to use operators and basis functions that can decompose shape information both in space and scale, enabling the construction of more descriptive, non-global, non-uniform shape priors to be included in the segmentation framework.

1.2 *Contribution of this Thesis*

Our existing contributions are:

- A novel knowledge-driven prior based on local shape filters for implicit representations of shape. We use these filters for the segmentation of blood vessels, and introduce the notion of segmentation with a *soft* shape prior, where the segmented model is not globally constrained to a predefined shape space, but is penalized locally if it deviates strongly from a tubular structure. We introduce the concept of a scale-specific shape filter that measures the deviation from a tubular shape in a local neighborhood of points, given a particular scale of analysis. Using this filter, we derive a region-based active contour segmentation algorithm for tubular structures that penalizes leakages. We present results on synthetic and real 2D and 3D datasets.

- A novel multi-scale shape representation using spherical wavelets for parametric models from which a data-driven prior can be learned to segment shapes with fine variations, such as cortical structures. This novel representation is motivated by the observation that the classical active shape models prior (ASM) technique cannot accurately encode fine, localized shape variations when the training set is of moderate size, since the relatively few eigenvectors will represent the most global modes of variation in the shapes. To address this issue, we present a novel algorithm that learns shape variations from data at *multiple scales and locations* using the spherical wavelet representation and spectral graph partitioning. Our results show that for a given training set size, our algorithm significantly improves the approximation of shapes in a testing set over ASM.

Based on this work, we propose the following three remaining research topics:

1. Segmentation of medical imagery using the spherical wavelet shape prior. We propose to derive an active surface evolution using the spherical wavelet coefficients as parameters of the segmentation energy to be minimized. The advantage of such a segmentation framework is that the spherical wavelet shape prior can directly be used to constrain the parameters during the surface evolution. We propose a set of experiments to test our surface evolution, both on synthetic and real data.
2. Testing the robustness of the shape registration and its influence on the segmentation results. Prior to building the spherical wavelet shape prior, it is necessary to align and remesg all structures in the training set. We propose to test the sensitivity of the segmentation results to noise in the registration and remeshing process for the two methods we currently use.
3. Shape classification using spherical wavelets and non-parametric permutation testing. Using spherical wavelets coefficients as features to describe a population of shapes, we propose to use the existing non-parametric permutation testing technique to test for shape differences in the caudate brain structure among two population of patients with and without schizotopal disorder.

CHAPTER II

BACKGROUND: SHAPE REPRESENTATION AND DEFORMABLE MODELS FOR MEDICAL IMAGE SEGMENTATION

2.1 *Overview*

This chapter gives an overview of existing techniques that are relevant to our work and is intended as a reference.

We focus in particular on reviewing existing shape representation and their impact on the construction of data-driven or knowledge-driven shape priors in Section 2.2. The description of shape priors is of particular importance as a background to our work.

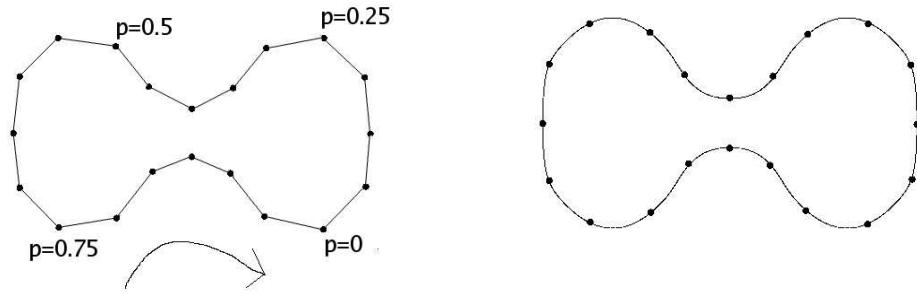
We then present deformable model segmentation algorithms. We present two types of deformable models based on the chosen shape representation: parametric models in Section 2.3 and geometric models in Section 2.4. Geometric Models will be used in Chapter 3 and parametric models will be used in Chapter 4 and the Proposed Work in Chapter 5.

2.2 *Object Representation*

In a model-driven segmentation framework, it is necessary to have a mathematical description of the boundary of an object. Two main approaches exist: to represent the boundary explicitly in a parametric form, or implicitly as the level set of a higher-dimensional function. We describe both approaches in this section. We note that this thesis only focuses on closed curves for 2D boundaries and simply-connected surfaces (surfaces with no holes) for 3D boundaries. See [36] for a treatment of open curves/open surfaces.

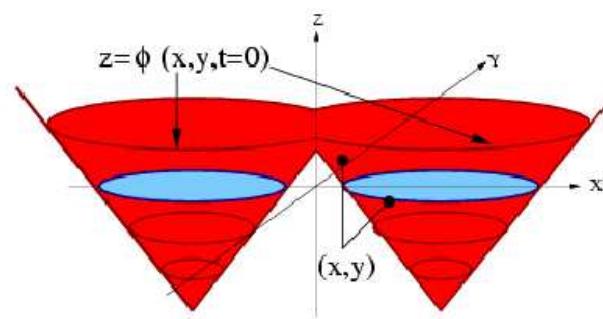
2.2.1 2D Parametric Object representation

Parametric models represent curves explicitly with a set of ordered control points located on the boundary of the curve. The points are identified with a parameter p varying from 0 to 1.

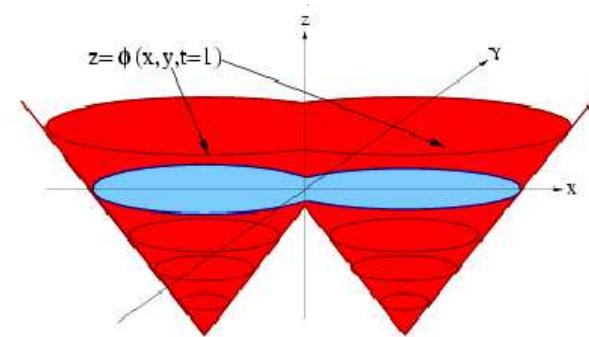


(a) A parametrized polygonal boundary

(b) The snake model is a parametrized closed boundary: it is a spline parametrized by a set of node points in \mathbb{R}^2



(c)



(d)

Figure 4: Level Sets of an embedding function ϕ for a closed curve in \mathbb{R}^2

Mathematically, the curve is represented as a collection of control points $\vec{\mathcal{C}}(p) : [0, 1] \rightarrow \mathbb{R}^2$ such that $\vec{\mathcal{C}}(p) = (x(p), y(p))$. If one assumes that it takes constant time to travel from one control point to another, a special parametrization is arc-length parametrization (denoted s in this thesis), where the distance traveled between points is equal (hence the curve is traversed at constant speed). In this case, the parameter s varies from 0 to L , where L is the total length of the curve. However with parametrized models, it is often the case that the distance between each control point is not equal, and we denote the arbitrary parametrization with p that varies between 0 and 1 (see Figure 4(a)).

To represent a closed curve with control points, the simplest technique is to use a polygonal approximation, where control points are vertices connected by straight lines (segments), as shown in Figure 4(a).

A more sophisticated and visually pleasing representation is to use a piecewise polynomial interpolation between control points (see Figure 4(b)). B-splines representation is an example of using a linear combination of local, continuous bases functions to represent a piecewise polynomial curve. For a n^{th} order B-spline, the bases functions are n^{th} order polynomials (with the first $n - 1$ derivatives continuous). Let $\mathbf{x}_i = (x(i), y(i))$, $i = 1, \dots, n$ be control points of a curve $\vec{\mathcal{C}}(p)$, and p be a linearly increasing parameter that has integer value at control points, that is $\mathbf{x}_i = \vec{\mathcal{C}}(i)$. The curve $\vec{\mathcal{C}}(p)$ is given by:

$$\vec{\mathcal{C}}(p) = \sum_{i=0}^{n+1} \mathbf{v}_i B_i(p) \quad (1)$$

where \mathbf{v}_i are coefficients and B_i are bases functions whose shape is given by the spline order. Third order splines are most common because it is the lowest order with a continuous second derivative, used to calculate curvature. The bases functions are non-negative and have local support. Each basis function $B_i(p)$ is non-zero only for $p \in (i - 2, i + 2)$. This means that if a control point changes its position, a resulting change of the curve will only occur in a small neighborhood around the control point. The position of points of the curve in between control points can be calculated directly from equation 1, using any positive real number for p .

Using B-splines, the curve can be *decomposed* spatially in terms of local polynomial

bases functions. The scale of the decomposition however (the size of the support of the bases function) is predetermined by the spacing of the control points.

To represent a curve at multiple scale, global Fourier bases functions can be used. Using arc-length parametrization, a continuous curve $\vec{\mathcal{C}}(s) : [0, L] \rightarrow \mathbb{R}^2$ can be represented as:

$$\vec{\mathcal{C}}(s) = \sum_n T_n e^{i(2\pi/L)ns} \quad (2)$$

The coefficients T_n of the series are called the Fourier descriptors and they uniquely represent the curve. They are given by:

$$T_n = \frac{1}{L} \int_0^L \vec{\mathcal{C}}(s) e^{-i(2\pi/L)ns} ds \quad (3)$$

In practice $\vec{\mathcal{C}}(s)$ is only evaluated at discrete control points, so the T_n are calculated from the discrete Fourier transform. The main advantage of using Fourier descriptors is that n represents the frequency of the Fourier basis, the higher the n , the higher the frequency. Therefore, the curve can be represented at a particular scale by filtering out (zeroing) Fourier descriptors that fall outside of the frequency content of that scale in equation 2. A spectrum analysis of the Fourier descriptors can also reveal which scale contains significant content, allowing for a compact representation of the curve by using only the Fourier descriptors with significant power.

2.2.2 3D Parametric Object representation

Just like for the 2D case, the continuous three-dimensional surface $\vec{\mathcal{S}} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ is represented as discrete points and linear combination of basis functions evaluated at those points.

The simplest representation for the shape is a triangular mesh (piecewise linear surface) and the use of finite difference method or finite element methods to discretize derivatives. In the finite difference case, one might choose the space of Dirac delta functions as bases functions. This means that the shape can be represented as a finite linear combination of K Dirac functions (where K is the number of vertices) defined at each vertex of the triangulation. In the finite element case, one uses the space of piecewise linear functions.

This means that the shape can be represented as a finite linear combination of K functions $v_k : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$, $k \in [0, K]$ (where K is the number of vertices). Each basis function v_k is defined to be 1 at vertex x_k and linearly decreases to 0 at every other vertex.

One disadvantage of such techniques is the local support of the bases functions (a function has support where it evaluates to nonzero). In the finite difference case for example, the movement of one vertex is completely independent of the movement of other vertices due to the fact that the Dirac delta function only has support at that vertex. As the surface evolves, vertices of triangles can move independently of each other and triangles can cross or overlap. As a result, frequent re-parametrization are needed to ensure that the surface does not develop holes, handles or foldings. In [10, 31], the surface is represented in the form of weighted sums of local polynomial basis functions. The use of higher order polynomials ensures a larger support for each basis function. However frequent re-parametrization remain an issue with these techniques.

For this reason, smooth, complete bases functions with global support are used in practice. The global support ensures a smoothness constraint between vertices of the triangulation and therefore require less frequent re-parametrization in practice [32]. The requirement of completeness means that every function $f \in L^2(\mathbb{R}^3)$ may be expanded in the basis as a linear combination of coefficients and basis functions [54]. We present such shape representation that are relevant to our work.

2.2.2.1 Fourier Surfaces

In the work of Staib et al. [45], a Fourier parametrization decomposes the surface into a weighted sum of sinusoidal basis functions. The surfaces are represented explicitly by three functions of two surface parameters:

$$\vec{\mathcal{S}}(u, v) = (x(u, v), y(u, v), z(u, v)) \quad (4)$$

The basis they chose is:

$$\phi = \left(\begin{array}{c} 1, \cos 2\pi mu, \sin 2\pi mu, \cos 2\pi lv, \dots \\ \cos 2\pi mu \cos 2\pi ls, \sin 2\pi mu \cos 2\pi lv, \\ \cos 2\pi mu \sin 2\pi lv, \sin 2\pi mu \sin 2\pi lv, \dots \\ (m = 1, 2, \dots; l = 1, 2, \dots) \end{array} \right)$$

The function is then represented by:

$$\vec{\mathcal{S}}(u, v) = \sum_{m=0}^{2K} \sum_{l=0}^{2K} \lambda_{m,l} [a_{m,l} \cos 2\pi mu \cos 2\pi lv + b_{m,l} \sin 2\pi mu \cos 2\pi lv + c_{m,l} \cos 2\pi mu \sin 2\pi lv + d_{m,l} \sin 2\pi mu \sin 2\pi lv]$$

where

$$\lambda_{m,l} = \begin{cases} 1 & \text{for } m = 0, l = 0 \\ 2 & \text{for } m > 0, l = 0 \text{ or } m = 0, l > 0 \\ 3 & \text{for } m > 0, l > 0 \end{cases}$$

The parameters of the model are $p = [a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y, a_z, b_z, c_z, d_z]$, where a_x represents all the a_m, l for the function $x(u, v)$. The main issue with this technique, as with any 3D parametrized active surface, is the choice of surface parametrization (u, v) . Spheres or cylinders can be easily parametrized, but more complex surfaces are not. The torus is the easiest surface to represent because surface parameters are forced to be periodic. Other types of surfaces can be described using subsets of the Fourier bases which flatten out or constrain the torus in different ways. Since parameters representing open surfaces result in discontinuities at the boundaries, the discontinuities are avoided by having the two surface parameters start at one side of the surface, trace along the surface to the other end, and then retrace the surface in the opposite direction to create a closed path and a periodic parametrization. Closed surfaces are the most difficult to represent because they are the most dissimilar to the torus. They are represented as open surfaces (tubes) whose ends close up to a point at both ends.

Despite this complexity, one nice feature of the Fourier representation is the inherent multi-resolution properties. The coefficients associated with the basis functions with higher frequency represent higher spatial variations on the surface. These coefficients can thus be truncated and the series will still represent relatively smooth objects accurately, using only

a few coefficients associated with the lowest frequencies. Another advantage of the Fourier representation is the geometric interpretation of the coefficients. Low index coefficients describe global shape deformations and higher indexed coefficients represent more local deformations.

More recent work on using Fourier bases has avoided the parametrization problem by first mapping the surface to the sphere, and decomposing the shape signal using bases functions defined on the sphere.

2.2.2.2 Spherical Harmonics

In the work of Brechbuhler et al. [5], a continuous, one-to-one mapping from the surface of an original object to the sphere is defined. The mapping is a nonlinear optimization constrained by two requirements: minimization of distortions and preservation of area. The basic idea is to start with an initial parametrization of the shape. Then the initial parametrization is optimized so that every surface patch gets assigned an area in parameter space that is proportional to its area in object space, while the distortion is minimized. This global parametrization allows for the systematical scanning of the object surface by the variation of two parameters θ and ϕ , overcoming previous limitations of expressing object surfaces in polar-coordinates, which restrict such descriptions to star-shaped objects. The object surface can then be expanded into a complete set of spherical harmonics basis functions.

The basis functions are defined as:

$$\begin{aligned} Y_l^m(\theta, \phi) &= \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos \theta) e^{im\phi} \\ Y_l^{-m}(\theta, \phi) &= (-1)^m Y_l^{m*}(\theta, \phi) \end{aligned} \quad (5)$$

where Y_l^{m*} denotes the complex conjugate of Y_l^m and P_l^m describes the associated Legendre polynomials

$$P_l^m(w) = \frac{(-1)^m}{2^l l!} (1-w^2)^{\frac{m}{2}} \frac{d^{m+l}}{dw^{m+l}} (w^2 - 1)^l \quad (6)$$

Figure 5 shows a visualization of the spherical harmonic functions on the sphere. A darker color shows where the function has support. The figure shows the real parts of the

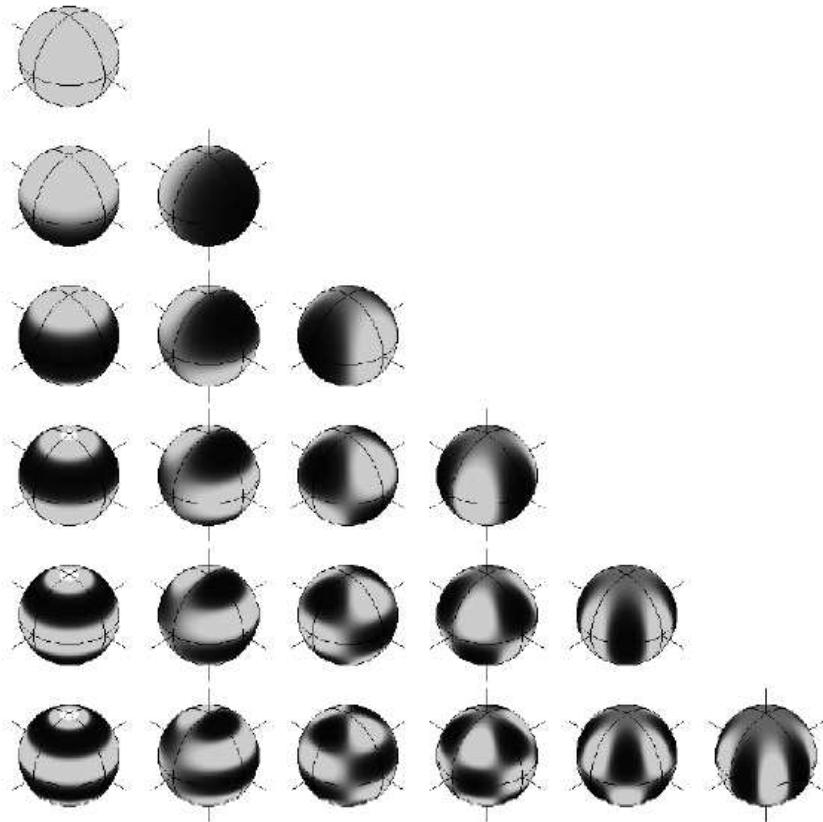


Figure 5: Visualization of the spherical harmonic functions on the sphere. A darker color shows where the function has support. The figure shows the real parts of the spherical harmonic function Y_l^m , with l growing from 0 (top) to 5 (bottom), and m ranging from 0 (left) to l in each row (from [19])

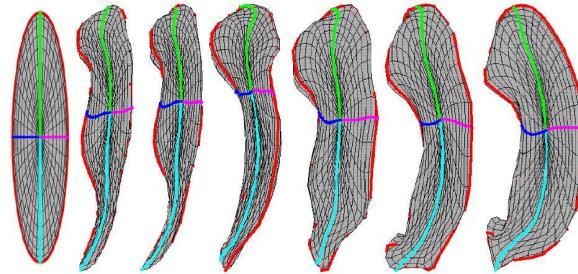


Figure 6: Visualization of the spherical harmonic correspondence. A first order ellipsoid and six left ventricles are displayed. The surface net shows the (θ_i, ϕ_i) parametrization (same parameters = same homologous points). The ridges on the first order ellipsoid are the equator and $0, \pi/2, \pi, 3\pi/2$ meridian lines in all objects. The equator and meridian lines are emphasized in different colors. The poles are at the crossing of the meridian lines (from [19])

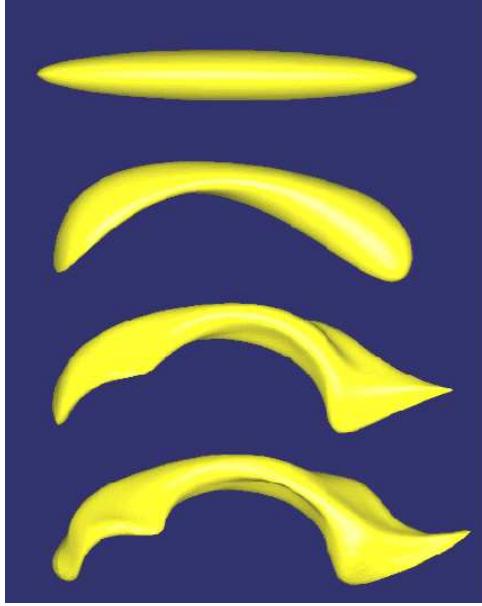


Figure 7: Visualization of a lateral ventricle (side view) at different spherical harmonics degrees; $m=1,4,8,12$ top to bottom (from [19])

spherical harmonic function Y_l^m , with l growing from 0 (top) to 5 (bottom), and m ranging from 0 (left) to l in each row.

To express the surface $\vec{\mathcal{S}}(\theta, \phi) = (x(\theta, \phi), y(\theta, \phi), z(\theta, \phi))$ using spherical harmonics:

$$\vec{\mathcal{S}}(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_l^m Y_l^m(\theta, \phi) \quad (7)$$

where the SPHARM coefficients c_l^m are three-dimensional vectors due to the three coordinate functions x,y and z. The coefficients c_l^m are obtained by solving a least squares problem, see [5] for the derivation.

Just like for the Fourier surfaces, the coefficients of the spherical harmonic functions of different degrees provide a measure of the spatial frequency constituents that comprise the structure. Partial sums of the series in Equation 7 can be used to represent selected frequencies of the object. As higher frequency components are included, more detailed features of the object appear. This process is shown in Figure 7.

The parametrization of surfaces is invariant to object scaling since the whole surface is mapped to the unit sphere. It is also invariant to translation if the object is centered at

the origin. However the coefficients obtained still depend on the orientation of the object in space. This means for example that if two objects with identical shape differ by a rotation, different points of the two objects will be mapped to the north pole. To get rid of these dependencies, it is necessary to rotate the object to a standard position, so that in our previous example, the two same points on the object map to the north pole. This can be done by representing the object using only the first order ellipsoid (using the first three coefficients) and applying a rotation in parameter space so that for all shapes the poles of the first order ellipsoid correspond to the poles of the sphere, and one end of the short axis of the ellipsoid correspond to the point where the Greenwich meridian meets the equator (there is a mirroring ambiguity depending on which end of the axis is mapped to the Greenwich-equator point. The author propose to use information from higher degree coefficient to disambiguate that case). Figure 6 shows this process.

Spherical harmonics have been used in medical imaging for shape classification [19]. For shape classification, a distance between two objects is calculated directly from their spherical harmonics coefficients via a difference calculation. The authors conducted a statistical shape analysis of the lateral ventricles, a fluid filled structure in the center of the human brain, that aimed to distinguish monozygotic twins from dizygotic twins and from unrelated pairs. They took pairwise differences in volume and SPHARM coefficient for each pair of twins. Significant differences between MZ and DZ pairs could not be found by volume measurements but only by SPHARM measurements. The authors identified that one weakness of shape analysis by SPHARM is the non-intuitive and non-localized nature of the set of coefficients. If a particular coefficient is found to be statistically significant in shape difference findings, the coefficient itself does not reveal the localization of the effect, only its “frequency”.

In chapter 4, we propose to use a set of basis functions, spherical wavelets that address this shortcoming.

2.2.3 Adding Data-Driven Shape Priors to Parametric Models

Parametric models can be restricted to a particular shape space by constraining the values of the coefficients of the bases functions used to represent the shapes. The main challenge is to learn the distribution of the coefficients from example shapes, called a training set. These shapes are said to belong to the same “population”, meaning that they represent the same type of object (for example brain ventricles). The most common assumption is to assume a joint Gaussian distribution for the shape coefficients, as in the active shape model.

2.2.3.1 Active Shape Models

In active shape models (ASM) [11], the authors represent shapes as linear combinations of landmark points. The landmark points must be placed in the same way on each object boundary in a training set. Figure 8 shows an example of landmarks placed on several brain structures in a single model.

Each shape can be described by three coordinate functions, $x, y, z \in R$ such that the position of the n^{th} landmark is $(x(n), y(n), z(n))^T, n \in [1, N]$.

The k^{th} shape in a population of K shapes is a column vector of size $3N \times 1$:

$$S_k = [x_k(1), \dots, x_k(N), y_k(1), \dots, y_k(N), z_k(1), \dots, z_k(N)]^T, k \in [1, K] \quad (8)$$

Since all landmarks are registered, we can interpret each entry in S_k as a random variable so that each shape is a realization from a multivariate probability distribution, $P(S)$.

A mean shape can then be calculated as $\bar{S} = \frac{1}{K}(\sum_{k=1}^K S_k)$.

The main assumption of ASM is that the set of K shapes in the training set gives a cloud of K points in $3N$ dimensional space that is approximately *ellipsoidal* and has a multivariate normal distribution in \mathbb{R}^{3N} :

$$P(\mathbf{S} = \mathbf{s}_i) = \frac{1}{z} \cdot \exp\{-(\mathbf{s}_i - \bar{\mathbf{S}})^T \mathbf{C}^{-1} (\mathbf{s}_i - \bar{\mathbf{S}})\} \quad (9)$$

where s_i is a shape realization, z is a normalization constant, \bar{S} is the mean shape and C is the covariance matrix that encodes the axes of the ellipsoid cloud of data.

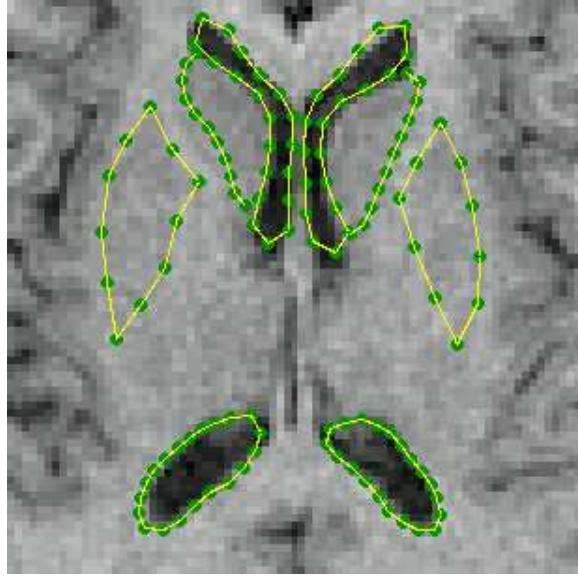


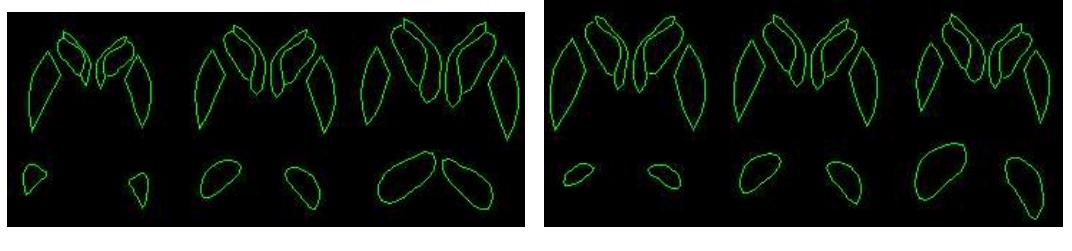
Figure 8: Outline of several brain structures in a single model labeled with landmarks (from [11])

To find the major axes, the covariance matrix C is estimated by building a covariance matrix SS^T from the data and diagonalizing it via singular value decomposition, such that $C \simeq SS^T = U\Sigma U^T$. The eigenvectors U are the major axis of the ellipsoidal cloud and form an orthonormal L^2 basis for the shapes. Hence each major axis represents a *mode of variation* of the data. The corresponding eigenvalue σ is the magnitude of the variation.

With this interpretation, it is common practice in the literature to show shape variation by selecting one of the eigenvectors and varying its magnitude from $-n\sqrt{\sigma}$ to $n\sqrt{\sigma}$ where $n = 2$ for example and adding the result to the mean shape to display the variation, as shown in Figure 9.

2.2.3.2 Using the Covariance Matrix for Shape Prior

To use the preceding L^2 basis as a shape prior, one could directly apply Equation (9) to a new shape s_n to test whether it has a high probability of belonging to the shape class. Alternatively, to restrict a new shape to belong to the population, we can project the shape unto the most significant eigenvectors U_t (the ones who cumulative eigenvalues account for a high percentage of the total eigenvalues) and translate the coordinates of the new shape



(a) Varying the most significant mode of variation (± 2 std)

(b) Varying the second most significant mode of variation (± 2 std)

Figure 9: In ASM, the learned shape variation can be shown by selecting one of the eigenvectors (modes of variation) and varying its magnitude from $-n\sqrt{\sigma}$ to $n\sqrt{\sigma}$ with $n = 2$ and adding the result to the mean shape to display the variation (from [11])

to the closest point that lies inside or at a reasonable distance of the training data point cloud. A shape s_n is corrected to s'_n by applying equations 57- 59:

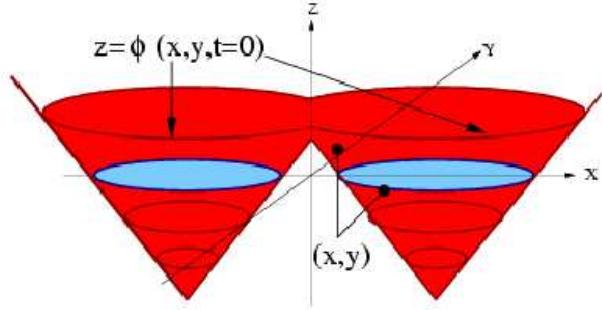
$$b_n = U_t^T(s_n - \bar{S}) \quad (10)$$

$$b'_n(m) = \begin{cases} b_n(m) & \text{if } -3\sqrt{(\sigma_m)} \leq b_n(m) \leq 3\sqrt{(\sigma_m)} \\ -3\sqrt{(\sigma_m)} & \text{if } b_n < -3\sqrt{(\sigma_m)} \\ 3\sqrt{(\sigma_m)} & \text{if } b_n > 3\sqrt{(\sigma_m)} \end{cases} \quad (11)$$

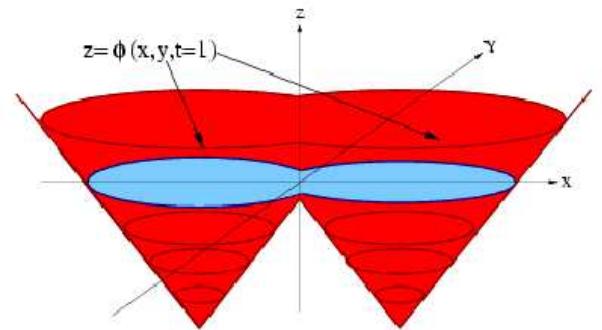
$$s'_n = \bar{S} + U_t b'_n \quad (12)$$

2.2.3.3 Limitation of PCA in Representing Finer Shape Details

One problem with this technique is that PCA favors the discovery of *global* variations over *local* variations. For a training set size of K shapes with N vertices, where $N \gg K$ and the size of S is $3N \times K$, the rank of the covariance matrix SS^T will be at most K . It can be shown that the eigenvectors of the covariance matrix describe the longest axes of the ellipsoid, and thus the most significant modes of variation in the vertices [11]. If there exist M shape “processes”, such that $M \gg K$, then only the first K processes that account for the most variation in the data (the most global processes) will be represented. This problem is aggravated if the training sample is very small since the number of independent processes discovered to build a shape prior will be at most K , the size of the training sample, as will be shown in Chapter 4.



(a)



(b)

Figure 10: Level Sets of an embedding function ϕ for a closed curve in \mathbb{R}^2

2.2.4 Implicit Object Representation

With this representation, curves are represented implicitly as the zero level set of a higher-dimensional scalar function in \mathbb{R}^2 and surfaces are the zero level set of a scalar function in \mathbb{R}^3 . The scalar function used is usually a signed distance map. In 2D, a curve $\vec{\mathcal{C}}(p) : [0, 1] \rightarrow \mathbb{R}^2$ to be represented is embedded as the zero level set of a surface, ϕ , in \mathbb{R}^3 whose height is sampled at regular intervals on the x, y grid as shown in Figure 10(a). Each height value $z = \phi(x, y)$ encodes the distance to the nearest point on the curve, with negative values inside the curve. Similarly, a surface $\vec{\mathcal{S}}(u, v) : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ to be represented is embedded as the zero level set of a hyper-surface, ϕ , in \mathbb{R}^4 . To visualize the curve, only the set of grid points $\phi(x, y) = 0$ are shown. For a surface, only the isosurface $\phi(x, y, z) = 0$ is

shown.

2.2.5 Shape Priors for Implicit Representations

Shape priors in the geometric active contours were introduced by Leventon et al [28]. They compute a statistical shape model over a training set of curves.

To build the shape model, they choose a representation of curves, and then define a probability density function over the parameters of the representation, in similar spirit to active shape models [11]. The major difference however is in the curve representation. Each curve in the training dataset is embedded as the zero level set of a higher dimensional surface, u , whose height is sampled at regular intervals (say N^d samples, where d is the number of dimensions). The embedding function chosen is the commonly used signed distance function, where each sample encodes the distance to the nearest point on the curve, with negative values inside the curve. Each such surface (distance map) can be considered a point in a high dimensional space ($u \in \mathbb{R}^{N^d}$). The training set, T , then consists of a set of n surfaces $T = u_1, u_2, \dots, u_n$.

The cloud of points corresponding to the training set is approximated to have a Gaussian distribution, where most of the dimensions of the Gaussian collapse, leaving the principal modes of shape variation. The mean surface, μ , is computed by taking the mean of the signed distance functions,

$$\mu = \frac{1}{n} \sum u_i$$

. The variance in shape is computed using Principal Component Analysis (PCA). The mean shape, μ , is subtracted from each u_i to create an mean-offset map, \tilde{u}_i . Each such map, \tilde{u}_i , is placed as a column vector in an $N^d \times n$ -dimensional matrix M . Using Singular Value Decomposition (SVD), the covariance matrix

$$\frac{1}{n} M M^T$$

is decomposed as:

$$U \Sigma U^T = \frac{1}{n} M M^T \quad (13)$$

where \mathbf{U} is a matrix whose column vectors represent the set of orthogonal modes of shape variation and Σ is a diagonal matrix of corresponding singular values. An estimate of a novel shape, \mathbf{u} , of the same class of object can be represented by k principal components in a k -dimensional vector of coefficients, α .

$$\alpha = \mathbf{U}_k^T(\mathbf{u} - \mu) \quad (14)$$

where \mathbf{U}_k is a matrix consisting of the first k columns of \mathbf{U} that is used to project a surface into the eigen-space. Given the coefficients α , an estimate of the shape \mathbf{u} , namely \mathbf{u}' , is reconstructed from \mathbf{U}_k and μ .

$$\mathbf{u}' = \mathbf{U}_k\alpha + \mu \quad (15)$$

The authors note that in general \mathbf{u}' will not be a true distance function, since convex linear combinations of distance maps do not produce distance maps. However, in practice the surfaces generally still have advantageous properties of smoothness, local dependence, and zero level sets consistent with the combination of original curves.

2.3 Parametric Deformable Models

The classical parametric model is the snake formulation. We also present interesting extensions of parametric models relevant to our work.

2.3.1 Snakes

Mathematically, a snake is a parametrized contour embedded in the image plane $I(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$. The contour is represented by $\vec{\mathcal{C}}(p) = (x(p), y(p))^T$ where $p \in [0, 1]$ is an arbitrary parametrization of the curve. The shape of the contour is defined by the energy functional:

$$E(\vec{\mathcal{C}}) = \mathcal{S}(\vec{\mathcal{C}}) + \mathcal{P}(\vec{\mathcal{C}}) \quad (16)$$

The final shape of the contour will correspond to the minimum of this energy functional. The first term,

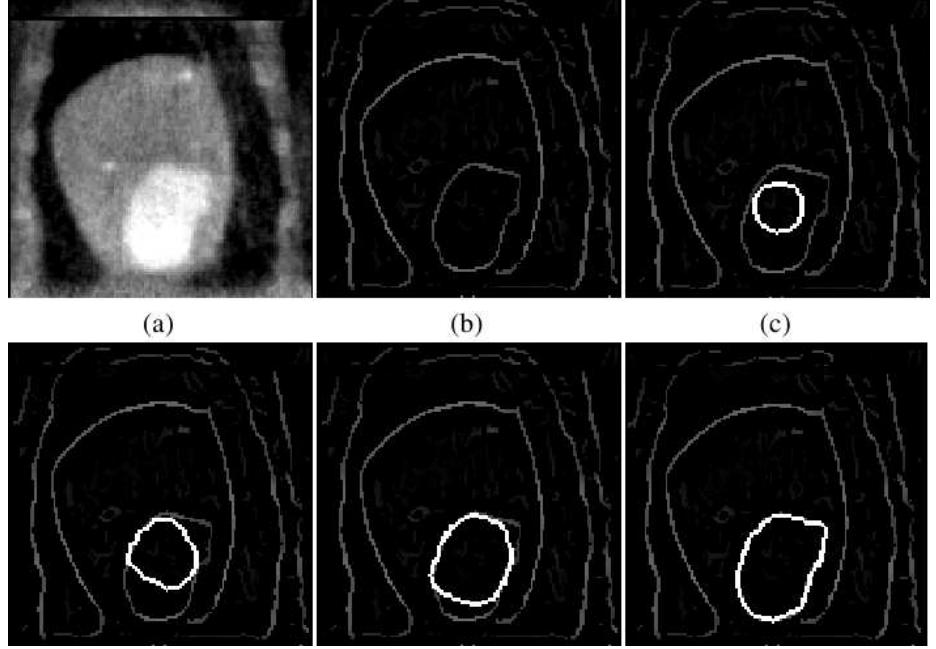


Figure 11: An example of a snake active contour (a) Intensity CT image of the heart left ventricle (b) edge detected image (c) initial contour (d) deformable contour moving toward the left ventricle boundary, driven by an inflating pressure force (from [32])

$$\mathcal{S}(\vec{\mathcal{C}}) = \int_0^1 \alpha(p)|\vec{\mathcal{C}}_p|^2 + \beta(p)|\vec{\mathcal{C}}_{pp}|^2 \, dp \quad (17)$$

is the internal deformation energy. $\alpha(p)$ controls the “tension” of the contour while $\beta(p)$ controls its “rigidity”. Since $\vec{\mathcal{C}}_p(p)$ is equivalent to the local length of the contour at p, a high value of α places a penalty on extraneous ripples of the curve. Similarly a high value of β places a penalty on extraneous bending. The second term,

$$\mathcal{P}(\vec{\mathcal{C}}) = \int_0^1 \Psi(\vec{\mathcal{C}}(p)) \, dp \quad (18)$$

is the potential energy that couples the snake to the image. $P(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a scalar function designed to have local minima to coincide with strong edges in the image that correspond to structure boundaries. A classical choice is:

$$\Psi(x, y) = \frac{1}{1 + \|(\nabla(G_\sigma(x, y) * I(x, y))\|^2} \quad (19)$$

where $G_\sigma(x, y)$ is the two-dimensional Gaussian function with standard deviation σ , ∇ is the gradient operator, and $*$ is the 2D image convolution operator. Minimizing $\mathcal{P}(\vec{\mathcal{C}})$ will therefore attract $\vec{\mathcal{C}}$ toward the edges.

Finding a curve $\vec{\mathcal{C}}$ that minimizes the overall energy function E is a variational problem [12]. To minimize Equation 16, $\vec{\mathcal{C}}$ must satisfy the following Euler-Lagrange equation[12]:

$$\frac{\partial E}{\partial \vec{\mathcal{C}}} = -\frac{\partial}{\partial p}(\alpha \frac{\partial \vec{\mathcal{C}}}{\partial p}) + \frac{\partial^2}{\partial p^2}(\beta \frac{\partial^2 \vec{\mathcal{C}}}{\partial p^2}) + \nabla P(\vec{\mathcal{C}}) = 0 \quad (20)$$

To solve Equation 20 given an initial curve $\vec{\mathcal{C}}_0$, the curve is made dynamic by augmenting it with a time parameter, $\vec{\mathcal{C}}(p, t)$, with $\vec{\mathcal{C}}(p, 0) = \vec{\mathcal{C}}_0(p)$.

By the chain rule,

$$\left\langle \frac{\partial E}{\partial \vec{\mathcal{C}}}, \frac{\partial \vec{\mathcal{C}}}{\partial t} \right\rangle = \frac{\partial E}{\partial t} \quad (21)$$

where $\langle \cdot, \cdot \rangle$ represents an inner product between two functions (infinite dimensional vectors). Therefore the energy is minimized in time when $\frac{\partial E}{\partial \vec{\mathcal{C}}} = -\frac{\partial \vec{\mathcal{C}}}{\partial t}$ (the inner product between two vectors $\langle u, v \rangle$ is minimum when $u = -v$).

The curve is then deformed at each time step according to the following gradient descent equation:

$$\gamma \frac{\partial \vec{\mathcal{C}}}{\partial t} = \frac{\partial}{\partial p}(\alpha \frac{\partial \vec{\mathcal{C}}}{\partial p}) - \frac{\partial^2}{\partial p^2}(\beta \frac{\partial^2 \vec{\mathcal{C}}}{\partial p^2}) - \nabla \Psi(\vec{\mathcal{C}}) \quad (22)$$

The coefficient γ is introduced to make units consistent. Thus the minimization is solved by placing an initial contour on the image domain and allowing it to deform at each time step according to equation 22. This can be done with Forward-Euler Equation [42]:

$$\vec{\mathcal{C}}(p, t+1) = \vec{\mathcal{C}}(p, t) + \Delta t * \frac{\partial \vec{\mathcal{C}}(p, t)}{\partial t} \quad (23)$$

where Δt is chosen based on stability conditions (the CFL condition [42]).

One limitation of the classical snake model is that it needs to be initialized very close to the object boundary in order to be within the attraction range of the potential force [32]. To address this limitation, Cohen [9] proposes to increase the attraction range by using a

pressure force in addition to the potential energy defined in Equation 18. The pressure force can either inflate or deflate the contour, removing the requirement that the model is initialized near the desired object boundary. The pressure force is defined as:

$$P(\vec{\mathcal{C}}) = w_p \mathcal{N}(\vec{\mathcal{C}}) \quad (24)$$

where $\mathcal{N}(\vec{\mathcal{C}})$ is the inward unit normal and w_p is a constant weighting parameter. $P(\vec{\mathcal{C}})$ is added to as an additional term in Equation 22 and has the effect of moving the control points of the model either along their inward normal (if w_p is positive) or along their outward normal (if w_p is negative). An example of using a deformable contour using an inflationary ($w_p < 0$) pressure force is shown in Figure 11. See [32, 55] for a detailed survey of snakes and their use in medical image analysis.

2.3.2 Snakes Implementation

Various numerical implementations of snakes exist in the literature [32, 55]. In order to numerically compute the snake evolution, it is necessary to discretize the energy $E(\vec{\mathcal{C}})$. The approach is to represent the continuous contour $\vec{\mathcal{C}}$ in terms of discrete points and linear combinations of basis functions expressed at those sample points. Finite differences [24], finite elements [37] and geometric splines [16] are local basis functions. Fourier Bases have also been used as global representation methods [44]. The trade-off is that local representations have more degrees of freedom to represent complex shapes with high local variations, but require special care to guarantee the smoothness of the curve. Global representations have built-in smoothness constraints, but fewer degrees of freedom.

The points are parametrized by the parameter p and the spacing between the points evolves dynamically as the curve evolves. The spacing between the points can be represented mathematically as the length of the tangent at each sample point, $\|\frac{\partial \vec{\mathcal{C}}}{\partial p}\|$. If the spacing between two points becomes too small or too large (relative to the rest of the points), a re-parametrization is necessary to keep the numerical implementation well behaved.

2.3.3 Snakes Limitations

Parametric deformable models such as snakes have been applied successfully in a wide range of applications [55]. However two limitations are often cited in the literature:

- In situations where the initial contour and the final solution differ greatly, it is necessary to re-parametrize the curve to keep the sample points equally spaced to more faithfully recover the object boundary and keep sample points from crossing and create “loops” in the contour. Methods for re-parametrization in 2D are straightforward, but in 3D re-parametrization is more complex and requires computationally expensive methods.
- The contour has a fixed topology and cannot automatically merge or split. If a topology change is needed, a new parametrization needs to be constructed which requires sophisticated schemes for detection and re-parametrization [33].

On the other hand, if the desired topology of the contour is known a-priori and a shape prior is applied to the contour to regularize the parametrization, parametrized models are still a method of choice due to their fast running time. In addition, parametrized models, especially when expressed with few parameters that can have a shape interpretation, can compactly incorporate shape priors. We will discuss the pros and cons of parametric models versus geometric models further at the end of this chapter.

2.3.4 Parametric Deformable Surfaces

Since most medical structures are three-dimensional, the use of a true 3D deformable surface to segment a structure in a volume ensures a globally smooth and coherent segmentation. The theory presented in section 2.3.1 can easily be extended to surfaces by using a parametrized surface embedded in the volume $I(x, y, z) : \mathbb{R}^3 \rightarrow \mathbb{R}$. The surface is represented by $\vec{\mathcal{S}}(u, v) = (x(u, v), y(u, v), z(u, v))^T$ where $u \in [0, 1], v \in [0, 1]$ parametrize the surface.

As an example of surface evolution, Staib et al. [45] use the Fourier surface representation presented in 2.2.2.1 for segmentation of 3D medical images. In order to fit one of these

models to the image data, a measure of fit is optimized by varying the model parameters. They assume that the surface is distinguishable by a measure of boundary strength $b(\vec{\mathcal{S}})$ computed from the image (using a term similar to Ψ defined in Equation 19). A measure of fit for curves can then be written as:

$$M(b, \mathbf{p}) = \int \int |b(x(\mathbf{p}, u, v), y(\mathbf{p}, u, v), z(\mathbf{p}, u, v))| dA \quad (25)$$

where \mathbf{p} is a vector consisting of the Fourier coefficients. Equation 25 can be evaluated by numerical integration by taking the derivative of the function $M(b, p)$ with respect to each parameter in \mathbf{p} . They show results for smooth spherical shapes, such as the heart's endocardium.

2.3.5 Discussion

Parametric Deformable Models can be a very fast and powerful technique, so long as topological changes are not expected to happen during the segmentation. If topological changes happen, more complex bookkeeping is needed. In addition to their fast running time, another main advantage of parametric models is the fact that shape priors can easily be included in the formulation by learning a prior distribution of the model parameters and limiting the evolution within a particular shape space by keeping the model parameters within the observed distribution. One main issue is the problem of aligning training samples in order to learn a prior distribution. In 3D, techniques that use spherical harmonics to align shapes address this issue.

2.4 Geometric Deformable Models

Geometric deformable models, proposed independently by Caselles et al. [6] and Malladi et al. [38], provide an elegant solution to address the topological change limitation of parametric deformable models. Curves and surfaces are evolved using only geometric measures, resulting in an evolution that is independent of the parameterization. As in parametric deformable models, the evolution is coupled with the image data to recover object boundaries. Since the evolution is independent of the parameterization, the evolving curves and surfaces

can be represented implicitly as a level set of a higher-dimensional function [34, 42]. As a result, topology changes can be handled automatically.

The purpose of curve evolution theory is to study the deformation of curves using only geometric measures such as the unit normal and curvature as opposed to the quantities that depend on parameters such as the derivatives of an arbitrary parameterized curve. Let us consider a moving curve $\vec{\mathcal{C}}(p, t) = (x(p, t), y(p, t))$ where p is an arbitrary parametrization and t is an artificial time parameter. If we denote the inward normal at point p at time t as $\mathcal{N}(p, t)$ and the curvature at point p at time t as $\kappa(p, t)$, then the evolution of the curve along its normal direction can be characterized by the following partial differential equation:

$$\frac{\partial \vec{\mathcal{C}}(p, t)}{\partial t} = V(\kappa(p, t))\mathcal{N}(p, t) \quad (26)$$

where $V(\kappa)$ is called speed function, since it determines the speed of the curve evolution. Only the normal direction is needed since the tangent deformation affects only the curves parameterization, not its shape and geometry [34].

The most extensively studied curve deformations in curve evolution theory are curvature deformation and constant deformation. Curvature deformation are *edge-based* active contours, meaning that the evolution of the contour only depends on quantities defined *on the curve boundary*. Constant deformations are region-based active contours, meaning that the evolution of the contour depends on quantities defined both *on the curve boundary* and *inside the curve*.

2.4.1 Geometric Active Contours

Geometric models have been first developed by Caselles [7] and Malladi [38]. As in the Snake framework, the active contour $\vec{\mathcal{C}}$ is deformed by a velocity composed of two terms. One is responsible for attracting the curve to some feature of the image as defined by some potential energy Ψ . The other term ensures that $\vec{\mathcal{C}}$ remains regular:

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = \Psi(\vec{\mathcal{C}})(c - \kappa)\mathcal{N} \quad (27)$$

where, as previously Ψ is small at edges and large elsewhere, c is a positive “inflation” constant, κ is the curvature and \mathcal{N} is the outward unit normal. This deformation is geometric,

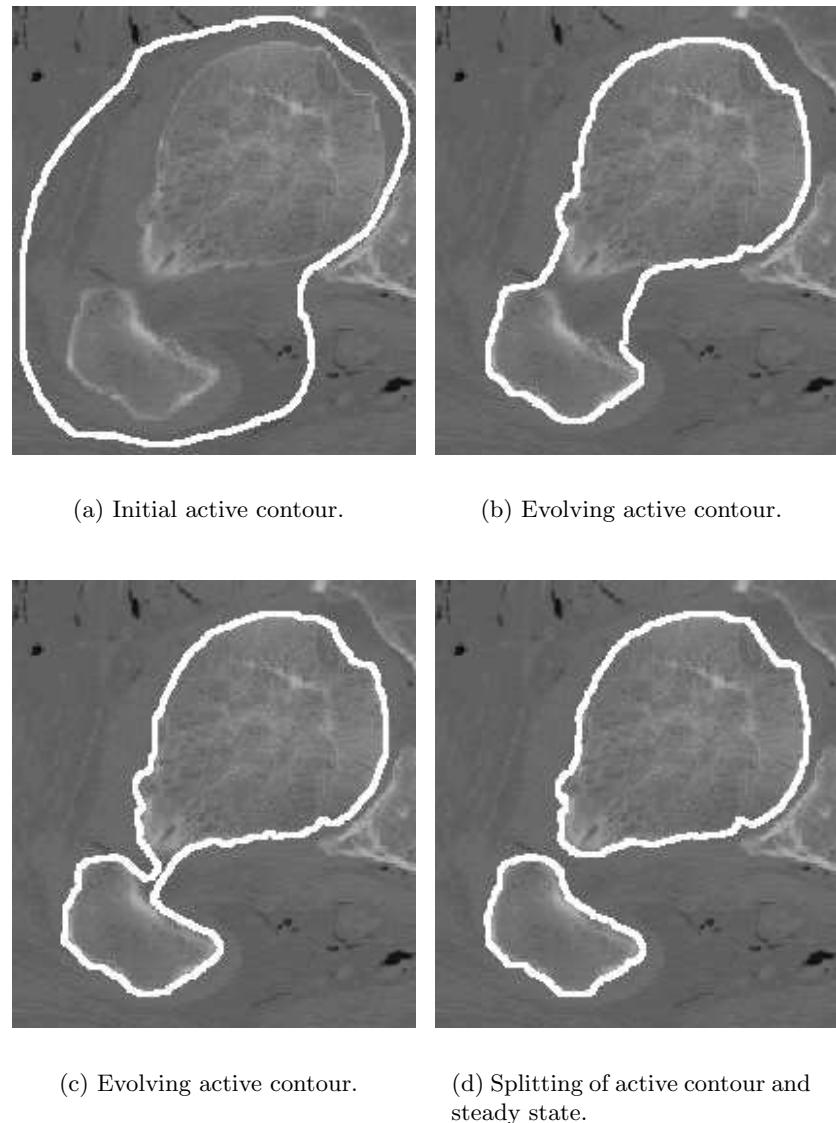


Figure 12: Bone segmentation in CT image with splitting shrinking conformal active contour.

i.e., it only depends on geometric quantities, such as the curvature and normal of the curve and does not depend on a particular parameterization of $\vec{\mathcal{C}}$.

This scheme works well for objects with good contrast. However, when the object boundary is indistinct or has gaps, the geometric deformable contour may leak out because the multiplicative term only slows down the curve near the boundary rather than completely stopping the curve. Once the curve passes the boundary, it will not be pulled back to recover the correct boundary.

2.4.2 Conformal (Geodesic) Active Contours

To remedy the latter problem, Caselles et al. [7] and Kichenassamy et al. [25] used an energy minimization formulation to design the speed function. It is based on finding a geodesic in a Riemannian space where the metric is derived from the image content, i.e., a curve of minimal weighted length. The metric used is Ψ , previously defined in Equation 19. This will result in finding a curve that has minimum energy when it is located on an edge (when Ψ is the smallest).

$$E(\vec{\mathcal{C}}) = \int_0^L \Psi(\vec{\mathcal{C}}) \, ds \quad (28)$$

This leads to the gradient flow:

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = \{(\nabla \Psi \cdot \mathcal{N}) - \Psi \kappa\} \mathcal{N} \quad (29)$$

Comparing (30) to the evolution of the geometric active contour (27) we see that the problematic constant c was replaced by $\nabla \Psi \cdot \mathcal{N}$. This term is large when the gradient of the edge detector Ψ coincides with the normal \mathcal{N} of the evolving curve and leads the curve into the boundary and eventually forces it to stay there [7]. An example of the geodesic active contour is shown in Figure ???. Note that in the case where Ψ would be uniform, Eq. (30) reduces to the geometric heat equation [21]. Figure 12 shows a bone segmentation in CT image with splitting shrinking conformal active contour.

2.4.3 Conformal (Geodesic) Active Contours with Inflationary Term

The geodesic active contour of Equation 30 is strictly deflationary, meaning that in the presence of no edges, the contour will just shrink to minimize its length (due to the curvature term). Other contours have been devised to allow for inflation as well, using an inflationary term c [55].

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = \{(\nabla \Psi \cdot \mathcal{N}) - (\kappa + c)\Psi\}\mathcal{N} \quad (30)$$

2.4.4 Region-Based Active Contours

One issue with edge-based active contours is that they are not robust to noise in the image and the gradient terms can stop the curve evolution at spurious edges. Recently there has been a considerable amount of work on image segmentation using region-based curve evolution techniques. In those techniques, the force that influences the evolution of the curve depends on a region statistics, inspired by the region competition work of Zhu and Yuille [58] and more recently the work of Chan and Vese [8] and Yezzi [57]. For example in [57], the authors developed a segmentation technique that evolves a curve in order to maximizes the difference between the mean of the pixels that lie inside the curve and the mean of the pixels that lie outside the curve.

The basic underlying mathematical idea is to write down an energy that minimizes a particular function f inside a region R , enclosed inside the curve $\vec{\mathcal{C}}$. The goal is then to find a gradient flow of $\vec{\mathcal{C}}$ that minimizes the following energy:

$$E(\vec{\mathcal{C}}) = \int_R f(\mathbf{x}) \, d\mathbf{x} \quad (31)$$

The gradient flow for this energy is derived in the appendix, and we give its final form:

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = -f \vec{\mathcal{N}} \quad (32)$$

This type of flow can be very interesting for medical imaging applications where strong edges are not necessarily present, but image statistics can be used reliably to segment

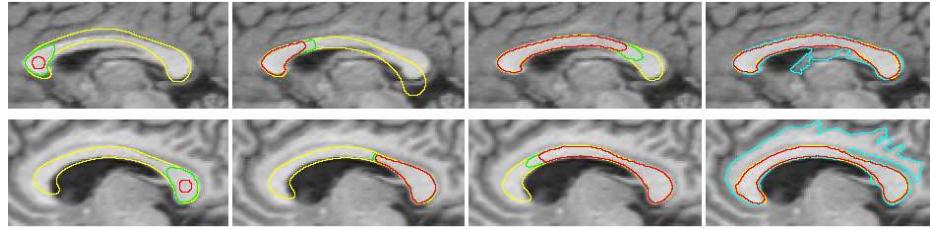


Figure 13: Four steps in the segmentation of two different corpora callosa. The last image in each case shows the final segmentation in red. The cyan contour is the standard evolution without the shape influence (from [28])

particular structures. For example, if an organ has a known mean and variance for the pixel values that lie inside and on its boundary, then this type of energy can be used:

$$E(\vec{\mathcal{C}}) = \int_R \phi(\mathbf{x}) d\mathbf{x} \quad (33)$$

where ϕ is negative if it lies within the range and positive otherwise (where the magnitude depends on how close it is in the range). This type of curve evolution will be used in Chapter 3.

2.4.5 Shape Priors for Geometric Active Contours

Shape priors in the geometric active contours were introduced by Leventon et al [28]. Their approach to object segmentation extends geodesic active contours by incorporating shape information into the evolution process. They first compute a statistical shape model as described in section 2.2.5 over a training set of curves. To segment a structure from an image, they then evolve an active contour both locally, based on image gradients and curvature, and globally to a maximum a posteriori estimate of shape and pose.

Overall the authors use the learned α shape parameters, as well as rigid pose parameters p (rotation, translation) to parametrize the model. To segment the curve, their technique uses a two step approach: they evolve an initial contour according to the geodesic curve evolution defined in Equation ???. Then, they seek to estimate the shape parameters, α , and the rigid pose parameters, p , of the final curve using a maximum a posteriori approach:

$$\langle \alpha_{MAP}, p_{MAP} \rangle = \operatorname{argmax}_{\alpha, p} P(\alpha, p | \vec{\mathcal{C}}(t), \nabla I) \quad (34)$$

In this equation, $\vec{\mathcal{C}}(t)$ is the evolving curve (or surface) at some point in time t . The term ∇I is the gradient of the image containing the object to be segmented. In calculating Equation 34, the authors use a shape prior $P(\alpha)$ to be the Gaussian model over the shape parameters, α , with shape variance Σ_k :

$$P(\alpha) = \frac{1}{\sqrt{2\pi^k}|\Sigma_k|} \exp\left(-\frac{1}{2}\alpha^T \Sigma_k^{-1} \alpha\right) \quad (35)$$

Once the optimal shape and pose parameters are found, the optimal curve $\vec{\mathcal{C}}^*$ is calculated from α_{MAP} and p_{MAP} and the prior is balanced with the image prior according to the following curve evolution equation:

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = \lambda_1 \{(\nabla \Psi \cdot \mathcal{N}) - \Psi \kappa\} \mathcal{N} + \lambda_2 \{\vec{\mathcal{C}}^* - \vec{\mathcal{C}}\} \quad (36)$$

This technique gives convincing results on imagery where the classic geodesic active contour leaks, as shown in Figure 13.

2.4.6 Summary

The parametric representations is Lagrangian in nature, meaning that coordinates representing the shape (either the coordinate of the control points, or the coefficients of the basis functions) move as the shape deforms. In contrast, implicit representations are Eulerian representation where the underlying coordinate system remains fixed as the shapes change (it is the value of each grid point that changes). One advantage of using an implicit representation is that topological changes in the shape during the evolution are automatically handled, as shown in Figure 10. Figure 10(a) shows the shape at time $t = 0$ that consists of two disjoint circles. As the value of the function ϕ changes over time, the shape becomes a single blob at time $t = 1$. No special handling is needed during merging or splitting of the shape. In contrast, changes in model topology for parametric models during the deformation, such as splitting and merging, need to be detected and require special handling. For example when two disjoint (individually parametrized) boundaries merge, the single object needs to be re-parametrized to reflect the new topology. One drawback of the implicit representation however is the higher dimensionality of the model as well as the

loss of meaningful shape representation since the shape of the object is only represented as a collection of unordered pixels. Further analysis is required to link pixels in a particular order to represent the notion of a parametrized curve.

Table 1 summarizes this chapter by comparing parametric and geometric active contours.

	Parametric Active Contours	Geometric Active Contours
Efficiency	***	*
Ease of Implementation (2D)	***	**
Ease of Implementation (3D)	**	**
Topology Change	No	Yes
Inclusion of Smoothness Shape Prior	Yes	Yes
Inclusion of Global Shape Prior	Yes	Yes

Table 1: Properties of parametric and geometric active contours. Adapted from [14]

We note therefore that the need for topology change vs. efficiency are big factors in deciding whether one should choose parametric or geometric active contours for shape segmentation.

Shape priors, both for parametric or geometric models, generally fall into two categories:

- Very local Priors that impose a smoothness constraint (usually based on second derivatives)
- Fully Global Priors that constrain the *whole* shape of the model to a predefined shape space, learned from a training sample.

These priors are good to segment structures that have a consistent shape and topology across the population, for example the brain’s ventricles that are consistently “butterfly-shaped” objects. However some structures, such as vessels, do not always have a consistent shape across patients. Each patient can have a different number of branches, located at different areas down from the root of the vessel. In another situation, when the training set is very small, the global prior can be very constraining since it only allows new shapes to resemble shapes seen in the training set. While it might be true that the shapes “overall” look the same (at a low resolution), it is usually the case that higher frequency information differs among shapes. We address these issues in our work.

CHAPTER III

SOFT SHAPE PRIORS: VESSEL SEGMENTATION USING A SHAPE DRIVEN FLOW

In this chapter, we present a segmentation method for vessels using an implicit deformable model with a knowledge-driven scale-specific shape prior. Blood vessels are challenging structures to segment due to their branching and thinning geometry as well as the decrease in image contrast from the root of the vessel to its thin branches. Using image intensity alone to deform an active contour for the task of segmentation often results in leakages at areas where the image information is ambiguous. To address this problem, we combine image statistics and a semi-local shape prior in a variational framework to derive a region-based active contour that segments tubular structures and penalizes leakages. We present results on synthetic and real 2D and 3D datasets.

3.1 *Blood Vessel Segmentation*

Blood vessel segmentation and visualization of blood vessels is important for clinical tasks such as diagnosis of vascular diseases, surgery planning and blood flow simulation. A number of methods have been developed for vessel segmentation, however most of those techniques do not use a shape prior, or use a strong global shape prior.

Using image intensity alone for the task of segmentation often results in leakages perpendicular to the vessel walls at areas where the image information is ambiguous. Leakages cause the segmented model to expand into areas that are not part of the vessel structure, and results in incorrect segmentation.

To remedy this problem, strong knowledge-driven global shape priors can be used, such as tubular shape templates to constrain the segmentation to a particular shape space. However since diseased vessels can have abnormal shapes, a strict shape template may result in incorrect segmentation that misses important anatomical information.

Data-driven shape templates, where a shape prior is learned from a training set, is not typically not used for vessel structures given that the topology of vessels (the number and location of branches) is highly variable based on the patient.

In this work, we introduce the notion of segmentation with a *soft* knowledge-driven shape prior, where the segmented model is not constrained to a predefined shape space with a predefined shape template, but is penalized if it deviates strongly from a tubular structure, since those deviations have a high probability of being leaks. Our method uses a shape prior localized in space in addition to image statistics to deform an active contour for the task of blood vessel segmentation.

3.2 Related Work

Many geometric methods exist for vessel segmentation that range from using no shape priors to strong shape priors. Tubular structures can be identified by the response of derivative and Gaussian filters convolved with the image. Sato et al. use the second derivatives of a set of multi-scale Gaussian filters to detect curvilinear structures and penalize high intensity (bumps) on the vessel walls [39]. The filter response can be used to visualize the vessels through Maximum Intensity Projections (MIP) or isosurface extraction. Since these methods rely on the intensity of the image, a noisy intensity map may result in incorrect filter response and additional shape information might be needed for a correct segmentation. Skeletons [17, 15] can be used as a basis of graph analysis of vessels, and further processing is needed to extract the 3D shape of the vessel. Krissian et al. use multi-scale filtering, based on a set of Gaussian kernels and their derivatives to extract a skeleton of vasculature [26]. The local maxima of the filter response is used to find centerpoints and radius information in order to fit a cylindrical model to the data. The restriction of the shapes can be a limitation since diseased vessels can have cross-section that deviate from an elliptical shape.

As presented in chapter 2, deformable models are a powerful technique for flexible automatic 3D segmentation. Deformable models are based on an initial contour or surface deformed by a combination of internal (shape) and external (image and user defined) forces to segment the object of interest. In particular, the addition of a shape prior as an internal

force can greatly increase the robustness of deformable models when applied to the task of vessel segmentation. Snakes are parametrized models and shape templates can easily be incorporated in this framework [18, 33]. However those methods have a limitation since the surface cannot handle topological changes as easily as the level set methods and re-parameterization is often necessary and complex in 3D.

Wang et. al. [52] did an analysis of different active contour models and reported that geometric active contours are the best choice for vessel segmentation due to their topological adaptability. As presented in section 2.4, geometric active represent a surface implicitly by the zero level set of a scalar-valued function. The evolution is carried out within this framework without the need for parameterization or explicit handling of topological changes [34, 42].

However, since there is no explicit parameterization, incorporating a knowledge-driven shape prior in the geometric active contour framework is more difficult and ad-hoc. For example, to address the issue of leaks that form at the root of vessels during a fast marching segmentation, Deschamps et al. [47] freeze a percentage of the points that are closer to the starting seed (since it is assumed that they have segmented the structure of interest) while allowing the fast evolving points to evolve normally. However this technique does not prevent leaks that form far away from the root of the vessels, close to the fast evolving part of the front. Strong knowledge-driven shape priors in combination with level set techniques for vessel segmentation was used by Lorigo et al. [29]. They evolve 1D curves in a 3D volume and then estimate the radius of the vessels locally using the inverse of principal curvature. However, this technique makes a strong assumption about the shape of the vessels since they are modeled as tubes with varying width.

To segment vessels with a knowledge-driven shape prior, it would be useful to have a shape prior that constrains the shape of the model locally to have a tubular shape, without over-constraining the shape globally. We call this concept a *soft* shape prior. The scale of the soft prior should be adapted to the vessel radius, so that deviations from a tubular shape on a small scale is acceptable, to account for fine details in vessel structure, but that on the scale of the vessel radius, the structure keeps a tubular structure and does not exhibit

leaks. To our knowledge, no existing geometric active contour uses such a prior for vessel segmentation.

3.3 Shape Driven Flow

We first present a classical region-based geometric active contour that is prone to leakages problems. We then present our method that adds a soft shape prior for a better behaved flow.

3.3.1 Region Based Flow

The classical region-based geometric active contour flow deforms the curve of interest according to a smoothing term and an image term ϕ . In this work, we use image statistics to determine the optimal image term ϕ . We learn the mean μ and standard deviation σ of a sample of pixels located inside the vessel structure to be segmented (the pixels are chosen interactively by the user). The ϕ term evaluated at pixel \mathbf{x} is then an adaptive threshold:

$$\phi(\mathbf{x}) = 1 - \left| \frac{I(\mathbf{x}) - \mu}{\sigma} \right|$$

where $I(\mathbf{x})$ is the image intensity at point \mathbf{x} . The term $\phi(\mathbf{x})$ is positive if $I(\mathbf{x})$ falls within one standard deviation of the learned mean μ and negative otherwise. More sophisticated schemes can be used to determine ϕ , such as using a Bayesian classifier [56].

We define the following energy in the region R inside the curve $\vec{\mathcal{C}}$ parameterized by arc-length s :

$$E(\vec{\mathcal{C}}) = - \int_R \phi \, d\mathbf{x} + \int_{\vec{\mathcal{C}}} ds \quad (37)$$

The first term is a region based active contour (see Section 2.4.4) that moves the curve towards the region of interest (the energy is minimum when the region enclosed inside the curve is as large as possible, with ϕ of positive value). The second term is an edge-based active contour that regularizes the curve as shown in Section ???. As shown in section 2.4.4 and the Appendix, we minimize Equation 37 by computing its first variation and solving the obtained Euler-Lagrange equation by means of gradient descent. We find the following curve evolution equation:



Figure 14: A Region Based Geometric Active Contour is prone to leak if the image term is not reliable

$$\frac{\partial \vec{C}(\mathbf{x})}{dt} = (-\phi(\mathbf{x}) + \kappa(\mathbf{x})) \vec{N} \quad (38)$$

where ϕ is a speed determined by the underlying image, $\kappa(\mathbf{x})$ is the curvature of the curve at point \mathbf{x} and \vec{N} is the unit inward normal to the curve. We evolve this active contour using Level Set Techniques [34, 42].

Image statistics alone might lead to unsatisfactory segmentations. Figure 14 shows an example of very noisy image data where areas of pixels close to the vessel have very similar image statistics. This results in a leak when segmented with the type of flow in Equation 38. More sophisticated algorithms can be devised based on image statistics or prior knowledge such as multi-scale filter responses tuned to detect vessels [57, 18, 51], but these algorithms will be very specific to the type of data and image acquisition. This leakage problem can be addressed in a more general way by adding a soft shape constraint to the flow so that the algorithm penalizes obvious leaks.

3.3.2 Shape Filters

We would like to locally determine the shape of a contour, and particularly areas where it is widening and potentially leaking. The information from derivatives of the curve, such as curvature, is too local since widening of the contour cannot be discriminated from small noise and bumps. We propose to use a local filter at a scale larger than the derivative scale.

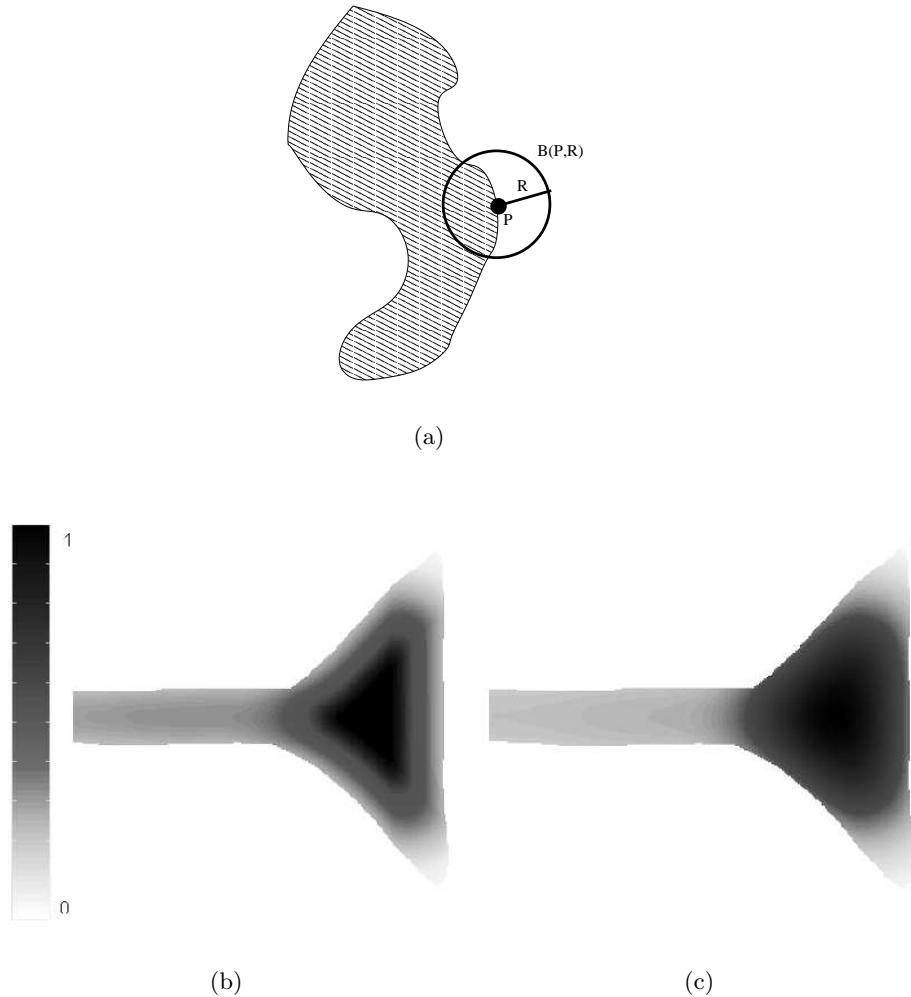


Figure 15: (a) $\epsilon_1(\mathbf{x}, r)$ is the intersection of the ball of radius r centered at \mathbf{x} and the region R inside the contour. (b) Points inside the widening region have a higher ϵ_1 measure but most points on the contour have the same measure. (c) Contour points close to the widening of the contour have a higher ϵ_2 measure

We define a local neighborhood $B(\mathbf{x}, r)$ in the shape of a ball (disk in R^2 , solid sphere in R^3) centered at the point \mathbf{x} and of radius r , see Figure 15(a). For every point \mathbf{x} inside and on the contour (region R), we define a filtering operation that calculates a measure ϵ_1 in the neighborhood $B(\mathbf{x}, r)$. The measure ϵ_1 is the percentage of points that fall both within the ball centered at \mathbf{x} and the region R inside the contour¹:

¹If $R \rightarrow 0$, then $\epsilon_1(\mathbf{x})$ is just the curvature at \mathbf{x} .

$$\epsilon_1(\mathbf{x}, r) = \int_R \mathcal{X}(\mathbf{x}, r, \mathbf{y}) d\mathbf{y} \quad \text{where } \mathcal{X}(\mathbf{x}, r, \mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \in B(\mathbf{x}, r) \\ 0 & \text{if } \mathbf{y} \notin B(\mathbf{x}, r) \end{cases} \quad (39)$$

The parameter r defines the scale of analysis. It must be chosen by the user and be an upper bound to the expected radius of the vessels. In our simulations, the user picked the width of the largest vessel with the mouse to define r .

The filter response ϵ_1 for a synthetic shape that represents a potential leak is shown in Figure 15(b). Given a radius that is the width of the tube, the points inside the widening region will have a higher measure than the points inside the tube. We will formalize this observation in the next Section by defining an energy minimization that uses the measure ϵ_1 to penalize regions inside the contour that deviate from a tubular shape.

3.3.3 Curve Evolution using Local Filters

We design a flow that uses the shape filter to penalize leaks by using the following energy:

$$E(\vec{\mathcal{C}}) = - \int_R \phi \, d\mathbf{x} + \int_{\vec{\mathcal{C}}} ds + \alpha \int_R \epsilon_1^p(\mathbf{x}) \, d\mathbf{x} \quad (40)$$

The first and second term are the same as for the region flow previously introduced. The third term is a constraint on shape. When we minimize $E(\vec{\mathcal{C}})$, the third term will force deviations from tubular shapes to be penalized. In order to obtain a curve evolution formula, we need take the first variation of Equation 40.

The first variation for such an equation with nested integrals was first derived in [2]. We give the full derivation in Appendix A.2 and state here its result (from Equation 95):

$$\frac{\partial C}{\partial t} = \left(-\phi(\vec{\mathcal{C}}) + \kappa(\vec{\mathcal{C}}) + \alpha \epsilon_2(\vec{\mathcal{C}}, p, r) \right) \mathcal{N} \quad (41)$$

where

$$\epsilon_2(\vec{\mathcal{C}}, p, r) = \epsilon_1^p(\vec{\mathcal{C}}, r) + p \int_R \epsilon_1^{p-1}(\vec{\mathcal{C}}) \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) \, d\mathbf{x} \quad (42)$$

We note that ϵ_2 that comes out of the energy minimization is an interesting term. The measure ϵ_2 is again the output of a local ball filter. For a point, the response is its ϵ_1 measure plus the sum of the ϵ_1 measure of its neighboring points that are inside the contour. For a

radius r similar to the vessel width, most points on the contour have the same ϵ_1 measure since locally the same percentage of neighbors fall within the filter radius. This can be seen in Figure 15(b), on the left. To see if the contour point lies near a leak region, it is necessary to look at the ϵ_1 measure of its neighbors *inside* the contour since their measure is high for points inside widening regions. This is what ϵ_2 measures, as shown in Figure 15(c), on the right. We observe that contour points close to the widening of the contour have a higher measure than contour points on the tube ².

Since ϵ_2 is always positive, the third part of Equation 41 is an erosion term (flow along the inward normal) that is proportional to the ϵ_2 measure. At a point with a high measure, the contour shape deviates from a tubular shape and therefore the flow is penalized. The parameter α is chosen according to the amount of penalty desired.

3.4 Implementation

We use the level set technique to implement the proposed flow.

3.4.1 Numerical Methods

We use the level set technique to evolve the curves described in Equations 38 and 41. The level set implementation of Equation 38 looks like:

$$\psi_t = -\frac{dC}{dt} \cdot \nabla \psi = \left(\phi + \kappa + \alpha(\epsilon_2) \right) \mathcal{N} \cdot \nabla \psi \quad (43)$$

If we define C as the set where $\psi = 0$ and let $\psi < 0$ inside C and $\psi > 0$ outside C , then $\vec{\mathcal{N}} = \nabla \psi / \|\nabla \psi\|$ and $\kappa = \nabla \cdot (\nabla \psi / \|\nabla \psi\|)$ allowing us to write:

$$\psi_t = \left(\phi + \nabla \cdot \frac{\nabla \psi}{\|\nabla \psi\|} + \alpha(\epsilon_1^p + (p-1)\epsilon_2^{p-1}) \right) \|\nabla \psi\| \quad (44)$$

We then use a Forward-Euler Equation to update the value of the level set function ψ at every time step:

$$\psi^{t+1} = \psi^t + \Delta t * \psi_t \quad (45)$$

²In our implementation, we scale both measure to lie between 0 and 1 so that all three terms in Equation 42 have similar scaling.

We choose a Δt that satisfies the CFL condition (ψ^{t+1} cannot be greater than $\psi^t + 1$).

3.4.1.1 Extensions

The image term ϕ should only depend upon the values of the image along the curve (the zero level set). For other level sets, we have to use the value of the image from nearby points on the zero level set. We therefore use the *extension* technique described in [42]. Similarly the value of ϵ_1 and ϵ_2 is extended from the zero level set to all other level sets.

3.4.1.2 Curvature

The second term is a curvature term that depend on the second and first derivatives of the curve C. We calculate those derivatives using central differences (for more details on numerical schemes for curvature, see [42]).

3.4.1.3 Upwinding and Entropy Scheme

When the sum of the first, third and fourth term is negative, the flow is a dilation since we are going in the negative direction of the inward normal. Conversely, when the sum is positive, the flow is an erosion. We therefore need to use a different upwinding scheme depending on the sign of the sum. We also use an entropy scheme for shock points. These schemes are detailed in [42].

3.4.2 Optimizations

The filter operations are time consuming. At every time step, we need to calculate ϵ_1 for every point inside the curve and ϵ_2 for every point on the zero level set.

We devised a scheme to only recompute those values for the set of grid points that changes from being positive to negative (points that were outside the contour and are now inside the contour).

We define a *valid neighborhood* of a point P as its neighbor points that falls both inside the local filter centered at P and inside the curve. We store for every point inside the curve their valid neighborhood size.

At every time step, we do the following:

- When the curve evolves, we detect all the grid points that were positive and became negative and put them in the set S^+ . We then update the valid neighborhood of points $\in S^+$ by adding +1 to their valid neighborhood count. Finally, we calculate the measure for points $\in S^+$.
- We then detect all the points that were removed from inside the contour (points that were negative and became positive) and put them in the set S^- . We then update the valid neighborhood of points $\in S^-$ by adding -1 to their valid neighborhood count.

3.5 Results

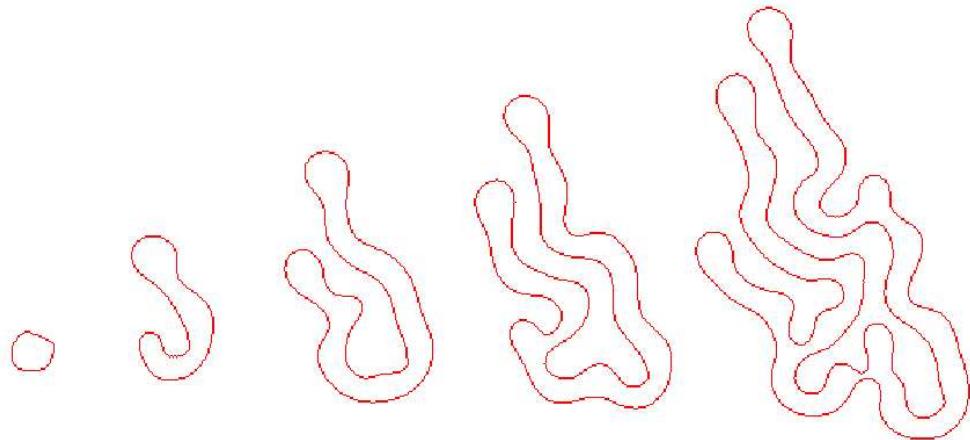
It is interesting to evolve a curve using Equation 41 without an image force, to see the effect that our flow has on the evolution of a small near-circle. We chose $\phi = 1$ (a constant image term) so that without the second and third term of the equation, the curve evolution would be a simple dilation of the circle. We then ran the full curve evolution, with 2 different radii ($r = 20, r = 40, \alpha = 0.75$). As seen in Figures 16(a) and 16(b), the curve evolves keeping a “tubular” shape at all times. The width of the tube depends on the radius of the local filter.

We now test our flow on both 2D and 3D datasets for synthetic and real data. For all flows presented, the user specified the expected biggest radius by clicking two points on the image.

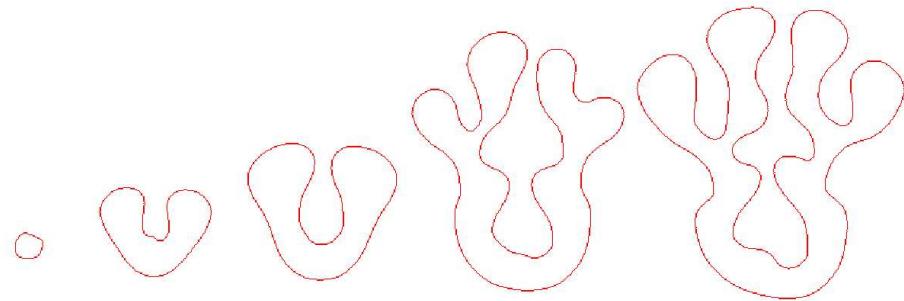
3.5.1 2D Images

In 2D, we first used our flow evolution on a synthetic model of a branching vessel. The radius was chosen to be the largest width of the vessel. We observed that the value of the chosen α influences the penalty on deviations from tubular structures. In Figure 17(a), for $\alpha > 0.65$, we observe an erosion around the branching area since points in that region have a higher measure. Figure 17(b), for $\alpha \leq 0.65$, the penalty is softened and the vessel is correctly segmented. This value is used for subsequent segmentations and erosion in branching areas was not observed.

We used our flow on a noisy projection of an angiogram and compared it to the base



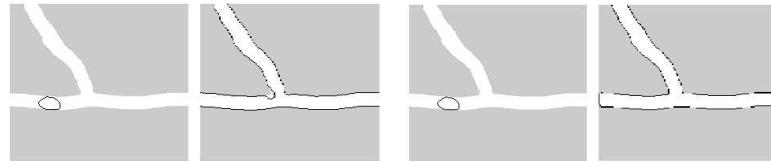
(a) Evolution in time, $p=2$, $r=20$, $\alpha=0.75$



(b) Evolution in time, $p=2$, $r=40$, $\alpha=0.75$

Figure 16: Dilation Flow with Shape Prior

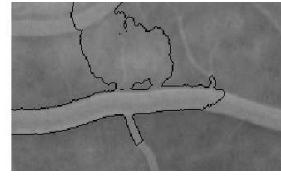
flow without a shape constraint. We show details of the segmentation where leaks were detected. When the neck of the leak is thin, the leak disconnects from the main vessel as shown in Figure 18. This is because points on the contour near the leak region have a high



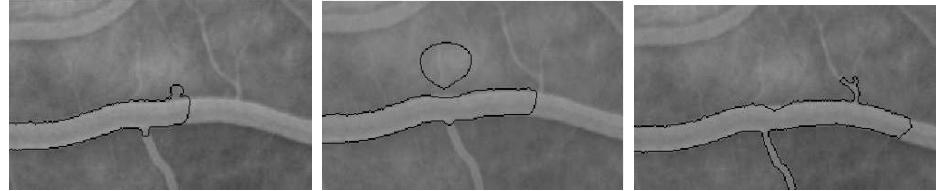
(a) A high penalty $\alpha=0.75$ causes erosion around the branch.

(b) $\alpha=0.65$. Erosion around the branch is not observed.

Figure 17: Flow with Shape Prior on 2D Synthetic Images



(a) Base Flow, no Shape Prior, 100 iterations



(b) Flow with Shape Prior, at $t=50$ (left), 100 (middle) and 200 (right) iterations.

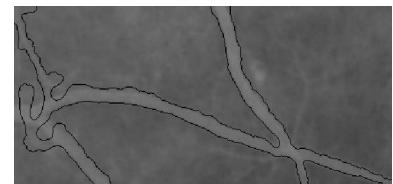
Figure 18: Vessel Flow on Angiogram Images

ϵ_2 measure (as can be seen in Figure 15(c)) that causes the contour to erode and eventually pinch off. Once the leak pinches off, the user can then click on the disconnected contour to eliminate it, or the algorithm can detect the change of topology [23] and automatically remove the leakage. The contour points at the neck of the leak are frozen so that the leak does not re-appear while the rest of the contour evolves.

In Figure 19 we observe that the flow with a shape prior is able to prevent many leaks and produce a flow that is much better behaved than the same flow without a shape prior. We notice that without a shape prior, the flow becomes “chaotic” and leaky regions merge with vessel regions (see Figure 19(a)). The repair of such leaks with user interaction would



(a) Base Flow, no Shape Prior



(b) Flow with Shape Prior

Figure 19: Vessel Flow on Angiogram Images

almost amount to a manual segmentation. Figure 19(b), shows that the flow is much better behaved and produces a better segmentation for most of the image. In the left part of the image, we notice an interesting behavior of the shape-constrained flow. This part of the image is very noisy and image statistics are almost identical inside and outside the vessel, so the base flow without a shape prior completely leaks out of the vessel area. The flow with a shape prior also leaks into the background since the image statistics no longer discriminates between foreground and background, but the flow maintains a “vessel-like” shape and the leak is mostly contained. This is important since we do not want the leak to expand to areas of the image that are correctly segmented.

3.5.2 3D Images

In this section we demonstrate our flow on two 3D CT datasets of a coronary artery. In all Figures, the color on the surface represents the measure ϵ_2 . We see that this measure is closely related to the thickness of a vessel.

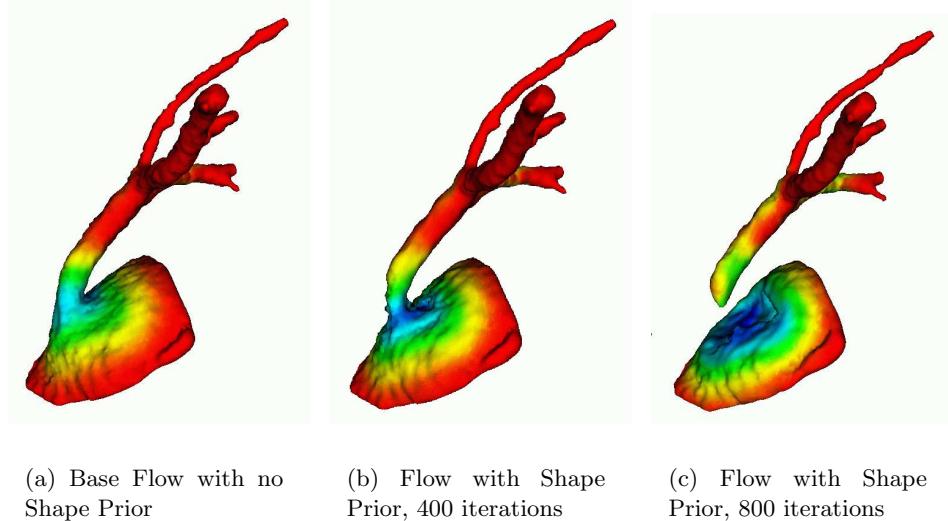


Figure 20: Different Flows on the first CT Coronary Data

For the first CT coronary dataset, if we use a flow based on image statistics alone, the artery “leaks” into the heart cage (Figure 20(a)). As we can see, the connecting area between the coronary and the heart has the highest measure. However, when we use the flow with the shape constraint, the leak “pinches” off from the main vessel artery (Figure 20(b))

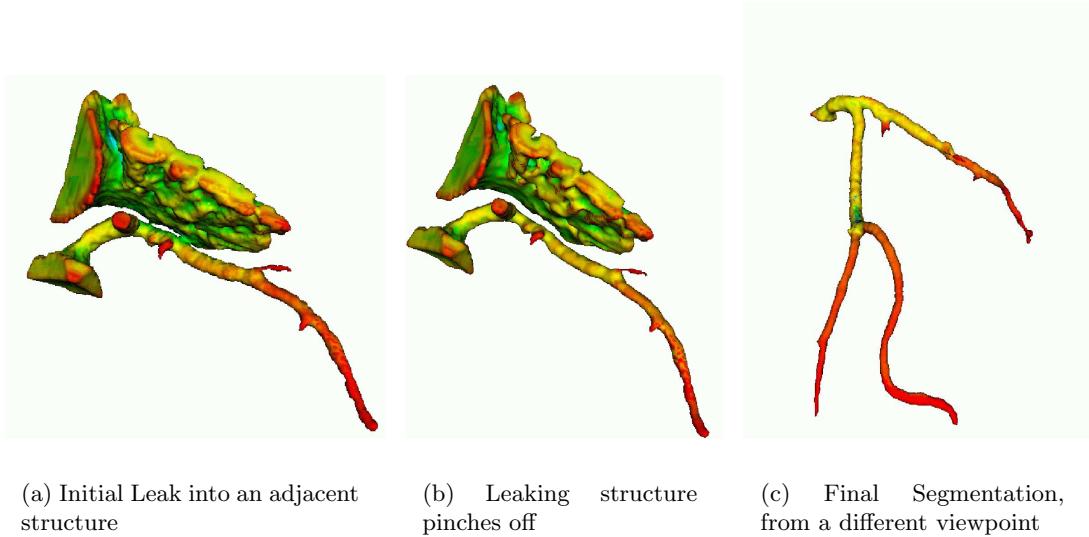


Figure 21: Base Flow with Shape Prior on the second CT coronary dataset

and 20(c)). The second CT coronary dataset leaks into an adjacent structure (Figure 21, left). Again, running the vessel flow separates the coronary from the leak. The user can then click on the isolated leak to remove it. The full segmentation is shown on the right.

3.6 Conclusion

We have presented a soft shape prior that can be combined with any other image force to deform an active contour and penalize leaks. We found that the shape prior successfully penalizes leak regions and either disconnects the leak from the vessel, or contains the leak. We find these results encouraging since in the presence of noise, the shape driven flow is better behaved than the flow based on image statistics alone. This flow can be combined with minimal user interaction to repair leak effects that were not prevented by the algorithm.

CHAPTER IV

MULTI-SCALE 3D SHAPE ANALYSIS USING SPHERICAL WAVELETS

Many structures in the body have fixed topology and similar shapes across patients, yet with some patient specific variations. In particular, various structures contain high frequency information on the surface (small patches of the surface with high curvature). For example, the surface of cortical structures found in the brain such as the caudate nucleus contains sharp features (see Figure 23). During the segmentation of such structures, this high frequency information needs to be kept, and therefore encoded in the shape prior. As seen in Section 2.2.3, shape variation in a population that is localized in space and scale cannot be well described by standard PCA analysis, used in active shape models (ASM). In addition, PCA suffers from poor approximation properties when the training set is small. To address these issues, we present a novel multi-scale 3D shape representation using spherical wavelets and a shape prior derived from this representation. We construct the shape prior by learning shape variation at *multiple scales and locations* from a training set. Our technique uses spectral graph partitioning to adaptively represent shape variations at multiple scales and spatial locations. This formulation achieves two objectives. First, our results show that our algorithm significantly improves the approximation of shapes in a testing set over PCA that tends to oversmooth the data, even in the presence of noise. Second, our algorithm can accurately reconstruct finer details even with a very small training set.

We propose to use this multi-scale representation for active contour segmentation by using as parameters the spherical wavelet coefficients. This will allow us to naturally combine the multi-scale shape prior presented in this chapter with the image prior by using the same parameters for the surface evolution flow. The multi-scale shape representation will also allow us to evolve the surface in a coarse-to-fine manner, hopefully providing robustness to noise.

4.1 Biological Driver: NAMIC grant

The National Alliance for Medical Imaging Computing (NAMIC) is a multi-institutional, interdisciplinary team of computer scientists, software engineers, and medical investigators who develop computational tools for the analysis and visualization of medical image data.

One of the driving biological projects is the study of schizophrenia. Schizophrenia is a multi-faceted illness affecting 1% of the US population and consuming a significant portion of the healthcare budget (estimates of yearly costs are 60 billion dollars) [53]. Yet the science of schizophrenia is only now beginning to take concrete form, primarily because neuroimaging techniques are finally providing a sufficiently detailed picture of the structure of the living brain and tracking the way the brain functions in controlled experimental settings. One important aspect of schizophrenia study is to segment selected cortical structures, such as the ventricles or the caudate nucleus whose shapes are believed to be indicators of the disease. The segmentation of such structures must be highly accurate, and therefore driven by prior knowledge and include high frequency information in order to conduct further shape analysis. One application of the segmentation is the analysis of shape difference in hippocampus between normals and subjects with schizophrenia. Currently those segmentation are conducted by hand. An automatic tool would be a great advance to be able to reliably and reproducibly segment cortical structures for multiple patients, across multiple time points.

4.1.1 Description of Dataset

In this chapter, we will use two datasets to illustrate our technique . The first dataset of prostates is not a NAMIC brain dataset, but was chosen as a test dataset due to the existence of some interesting high frequency content on the surface (for example on the superior side of the surface). Additionally segmentation of the prostate is currently being used intra-operatively to guide a surgery where radioactive seed are inserted inside the prostate for cancer treatment. Segmentation is currently being done manually, so a fast and accurate automatic segmentation technique would be desirable, using for example our proposed technique. The second dataset, caudate nucleus shapes is from the NAMIC dataset.

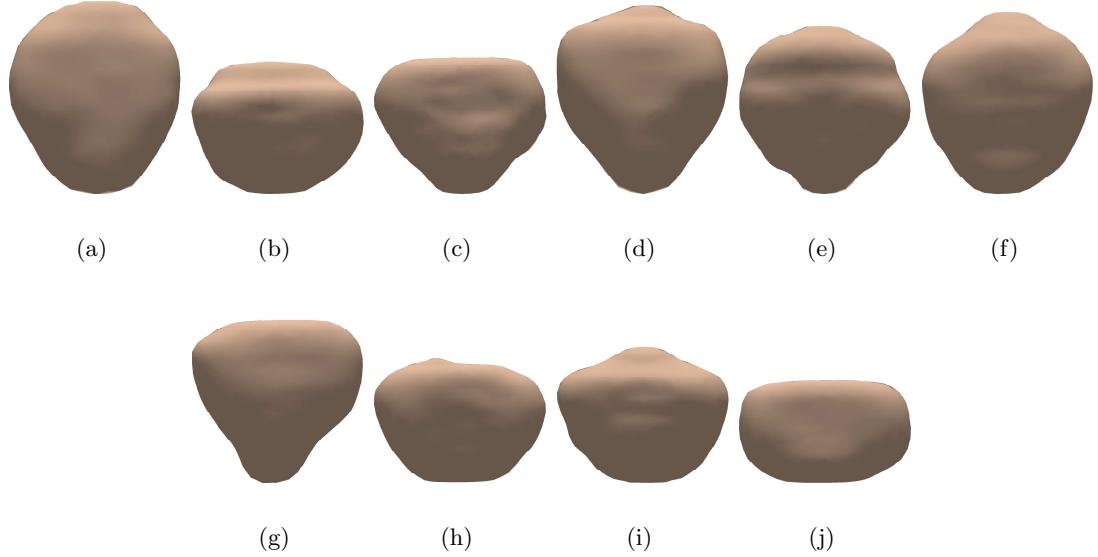


Figure 22: Example of 10 shapes from the Prostate dataset

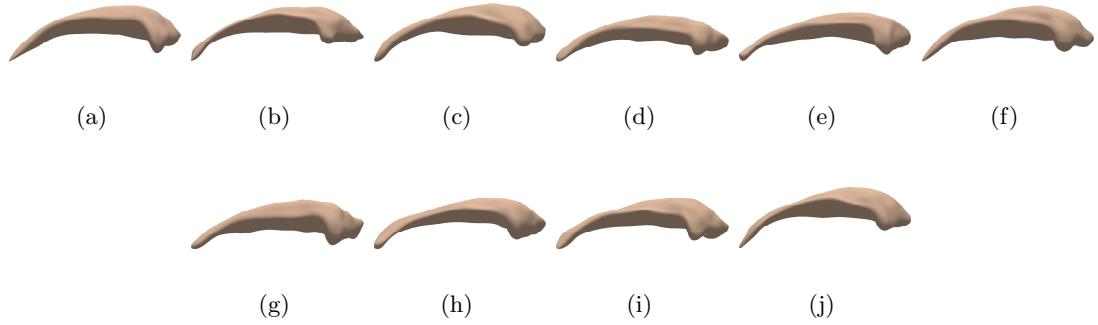


Figure 23: Example of 10 shapes from the Caudate dataset

4.1.2 Prostate

In this work, we use a dataset of 39 prostates (shown in Figure 22) obtained from pre-operative 1.5T MRI scans (axial, T2-weighted, 120mm field of view, matrix size 256×256 , 3.0mm thickness, 0.5mm gap, 1.5T, using both endorectal and pelvic coil arrays) using endorectal coil imaging provided to us by the Surgical Planning Lab of Brigham and Women's Hospital. In these images the prostate capsule is visible and was manually segmented by a radiologist. Each manual segmentation defined a 3D surface which was extracted as a triangulated surface using the Marching Cubes algorithm.

4.1.3 Caudate

From the NAMIC dataset, we used 29 caudates (shown in Figure 23) obtained from a 1.5 Tesla GE Echospeed system (General Electric Medical Systems, Milwaukee, WI) consisting of coronal series of contiguous spoiled gradient (SPGR) images (124 slices of 1.5 mm thickness, voxel dimensions $0.9375 \times 0.9375 \times 1.5$ mm). The labelmaps for the caudate were hand segmented. Each manual segmentation defined a 3D surface which was extracted as a triangulated surface using the Marching Cubes algorithm.

4.2 *Shape Representation*

In this chapter, we consider 3D shapes with a fixed topology and detailed surfaces. We chose to represent shapes using a parametrized surface as opposed to distance maps used in the level set framework. This representation can be more accurately and efficiently manipulated compared to a distance map representation of the shape that is higher-dimensional and increases in complexity if the volume is supersampled. Additionally, as long as topology changes are not expected during the segmentation evolution, a parametrized active contour can be used in this framework which is more efficient than a geometric active contour implemented via level sets.

4.2.1 Shape Registration

To build a shape prior of a population of shapes, we assume that we have ground truth labelmaps for a training set from that population, from which surfaces can be extracted using an isosurfacing algorithm. The next step is then to find an accurate point-to-point correspondence between surfaces so that further statistical analysis can be applied to the population. This can be achieved by finding a one-to-one mapping from each surface of the population to the sphere, constrained by the requirements that similar points on the shapes get mapped to the same point on the sphere. This can be achieved for example by using the technique of Brechbuhler et. al, as described in Section 2.2.2. In this work, we have used both the Brechbuhler technique, as well as another similar technique from Haker et. al that we describe next.

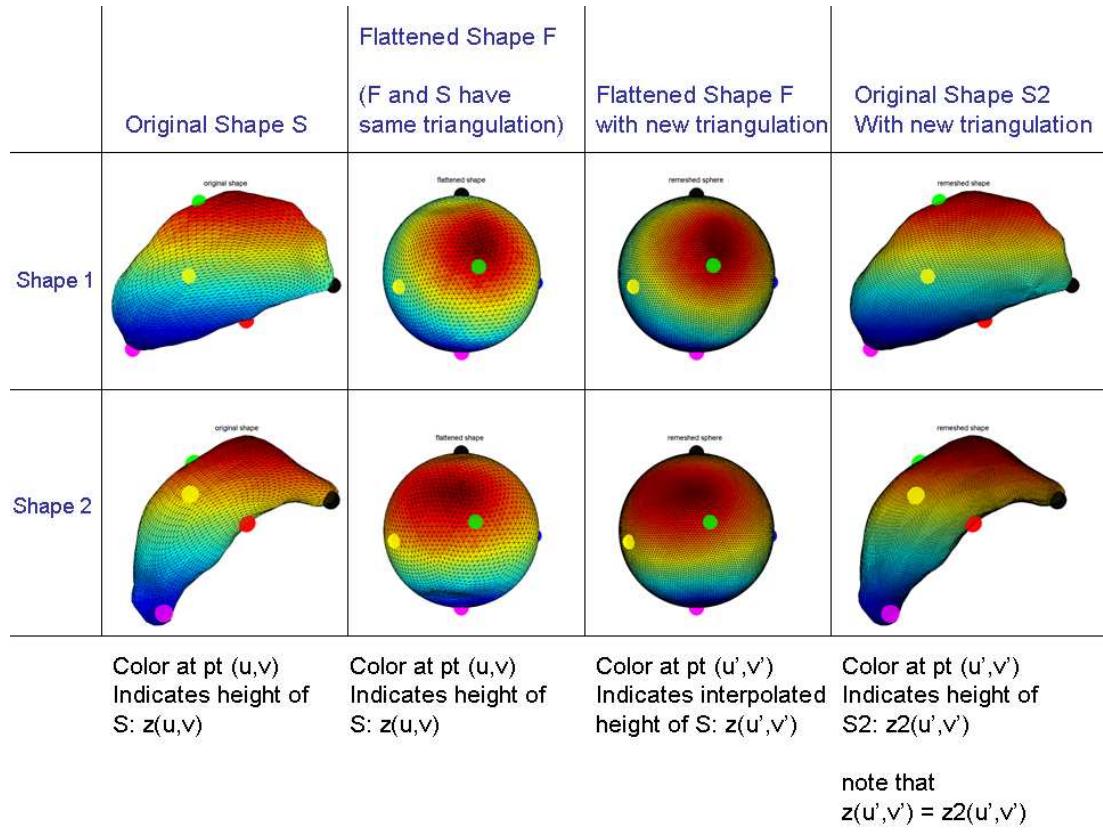


Figure 24: Haker et. al. registration algorithm: shapes are conformally mapped to the sphere using 6 landmark points to constrain the mapping. The sphere is then re-triangulated, and the coordinates of the original surface S are interpolated at the vertices of the new triangulation, producing a final shape with the new triangulation.

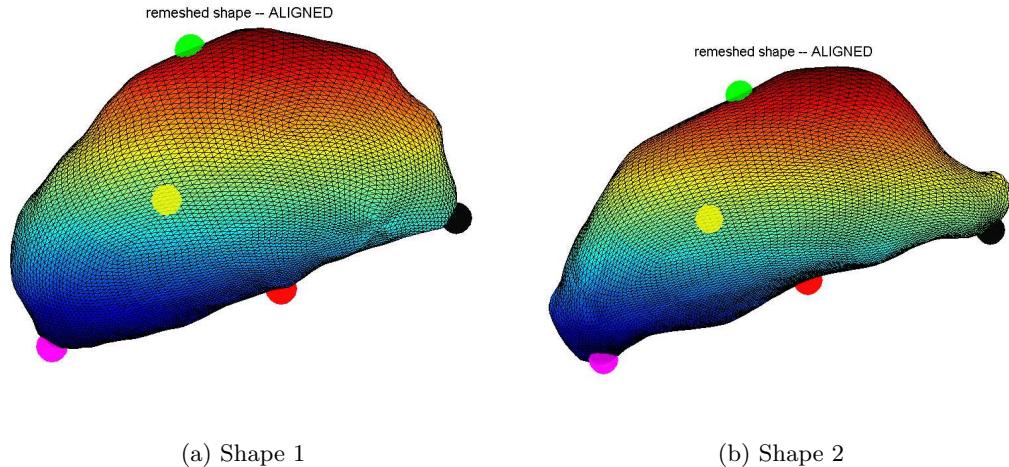


Figure 25: Shape 1 and 2 from Figure 24 realigned with the Procrustes technique

Initially, all shapes in a dataset have an arbitrary triangulation, resulting from the marching cubes. The first step is to map the shape to the unit sphere using a conformal (angle-preserving) mapping technique as described in [1, 22], $f : \Sigma \rightarrow S^2$, where Σ is the surface and S^2 is the sphere.

The technique uses conformal mapping while sending consistently chosen landmark points to pre-specified locations on the sphere using a thin-plate spline technique [?]. This ensures that the mapping of the original surfaces to the sphere is done in a consistent manner if the landmarks are consistently picked to be in the same location on all surfaces, thereby providing a point-by-point registration of all surfaces in the dataset.

As a result of this operation, $F = f(\Sigma)$ is the original shape “flattened” to the sphere (therefore Σ and F have the same triangulation, but different values for the coordinates of the vertices). This is shown in the second column of Figure 24. The color on the original surface Σ (first column) and the flattened surface F (second column) represents the value of the z coordinate of Σ .

The second step is to remesh F with a canonical and regular spherical mesh to produce a new shape F_2 . This mesh will be the same for all shapes. During the remeshing, the original coordinates of Σ defined at each vertex of F are interpolated, so they are now defined at the vertices of the re-triangulated shape F_2 . Finally, the remeshed shape Σ_2 is created using the new mesh and the interpolated coordinates.

The final step is to align the new shapes in \mathbb{R}^3 using the point-to-point correspondence provided by the remeshing. This can be done using Procrustes analysis [?] and is shown in Figure 25.

The key issue with this technique is to pick consistently 6 landmark points on each shape. For the prostate dataset, we used as landmark points the 6 points of intersection of the major axes of rotation of the prostate surface with the surface itself. The final result is shown in Figure 26 for two prostates.

For the caudate shapes, we have found the two farthest points on each shape (corresponding to the tip of the head and tail of the caudate), and then consistently picked 4 points on a plane that intersects the midpoint of the line connecting the two farthest points.

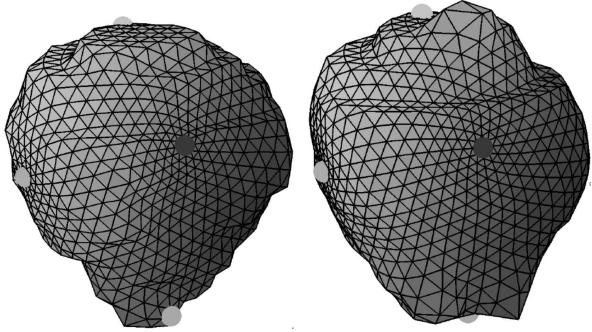


Figure 26: Two prostate surfaces. The surfaces have been mapped to the sphere and re-triangulated in the same way to establish point correspondence. The small spheres represent landmark points which were matched during the registration process.

We have also used the technique in [5] that does not require landmark points. Given the importance of the shape remeshing in the remaining statistical analysis, we plan to investigate the use of these different techniques and the effect of noise (so that shapes are slightly mis-registered), as will be discussed in chapter 5.

4.2.2 Statistical Modeling

Once registered, all shapes have N vertices and each shape can be described by its three coordinate functions, $x, y, z \in \mathbb{R}$ such that the k^{th} shape S_k is a column vector of size $3N$:

$$S_k = [x_k(1), \dots, x_k(N), y_k(1), \dots, y_k(N), z_k(1), \dots, z_k(N)]^T.$$

Since all vertices in the shape population are registered, we interpret each entry of S_k as a random variable and each shape as a realization from a multivariate probability distribution. A population of K shapes can be described by a mean shape $\bar{S} = \frac{1}{K}(\sum_{k=1}^K S_k)$ and a set of transformations $T = [T_1, \dots, T_M]$ that describe the variability observed in the population. Each transformation vector T_m is of size $3N$ where the i^{th} entry is a transformation applied to the i^{th} entry of the mean shape with a corresponding magnitude $\beta_m \in \mathbb{R}$, as shown in Figure 27.

Each transformation vector, or variation mode, can be characterized by :

1. *scale*: the transformation vectors T_k can be global and apply to all vertices (all entries of T_k are non-zero) or local (T_k is a sparse vector).

$$\text{Mean Shape: } \bar{S} = \frac{1}{K}(\sum_{k=1}^K S_k)$$

$$\text{kth Shape: } S_k = \bar{S} + \begin{bmatrix} T_1(1) & \dots & T_M(1) \\ \vdots & \dots & \vdots \\ T_1(i) & \dots & T_M(i) \\ \vdots & \dots & \vdots \\ T_1(N) & \dots & T_M(N) \end{bmatrix} \begin{bmatrix} \beta_k(1) \\ \vdots \\ \beta_k(i) \\ \vdots \\ \beta_k(M) \end{bmatrix}$$

mean Transformation vectors magnitude of transformation

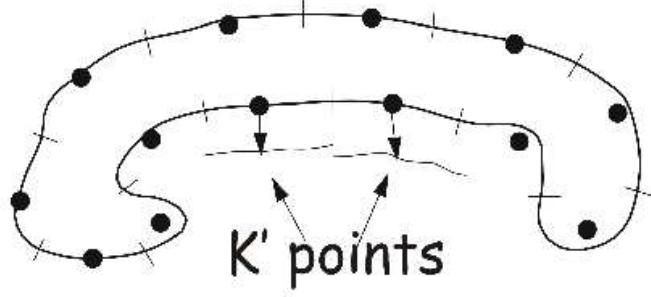
Figure 27: Shape Model

2. *spatial location*: the effect of the transformation vector T_k can be localized in space (non-zero entries of T_k are close spatially), or not.
3. *magnitude*: the value of each β_k

Characterization of local variations could be important for shape analysis since a disease, such as cancer, could affect only a portion of an organ's surface. Therefore descriptive shape priors should discern shape variations at different scales and spatial location.

4.2.2.1 Hierarchical Representation of Shape

As we have previously described in Section 2.2.3.3, PCA on a limited training set captures the most global modes of variation of the population. Recall that PCA favors the discovery of *global* variations over *local* variations. For a training set size of K shapes with N vertices, where $N \gg K$ and the size of S is $3N \times K$, the rank of the covariance matrix SS^T will be at most K . It can be shown that the eigenvectors of the covariance matrix describe the longest axes of the ellipsoid, and thus the most significant modes of variation in the vertices [11]. If there exist M shape "processes", such that $M \gg K$, then only the first K processes that account for the most variation in the data (the most global processes) will be represented. This problem is aggravated if the training sample is very small since the number of independent processes discovered to build a shape prior will be at most K , the size of the training sample



Schematic drawing for global/local spatial partition of the shape.

Figure 28: A contour partitioned into smaller segments

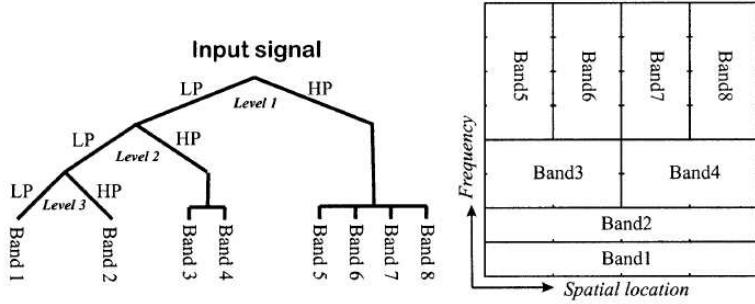


Figure 29: Scale Space decomposition. The wavelet transform is iterated for the low pass branches (LP). The outputs of the high-pass branches (marked as HP in the figure) are divided into a number of bands. The number of bands for each high-pass branch is determined by the level at which the high-pass filtering is performed. In this way, for a p level wavelet decomposition the frequency domain is divided into $B = 2^p$ bands. The space-frequency domain partition of the signal is shown on the right.

To address this limitation, the authors in [13] have proposed a hierarchical active shape model framework for contours in 2D medical imagery using 1D wavelets, with convincing results.

The basic idea is to use a hierarchical representation of shape. Assume that we are only interested in representing local variations of a segment of a shape, comprising K' points (see Figure 28). If K' is small enough, then a small training set of N shapes will be adequate to represent the variability of that segment using PCA, since K' and N are

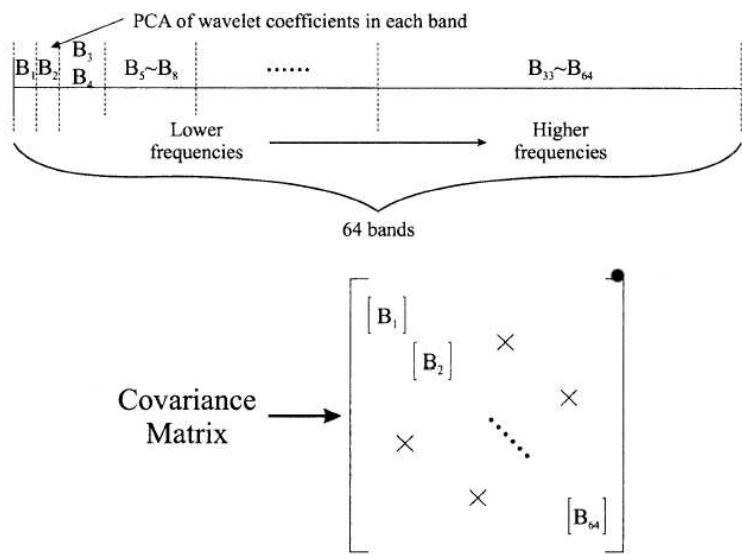


Figure 30: A wavelet transform is applied to the parametric functions representing a deformable contour. The resulting wavelet coefficients are then grouped into 64 bands, and the joint distribution of each band is estimated from the available training samples, via its mean and covariance matrix. This effectively transforms the covariance matrix of the full joint distribution into a matrix that is close to, but not necessarily exactly, a block diagonal matrix. The submatrix corresponding to band B reflects global shape characteristics, whereas the submatrix corresponding to B reflects local shape characteristics at a particular segment. B reflects local shape characteristics of a neighboring segment

close in magnitude. The eigenvectors for each segment will locally capture the fine details (high frequency) variations of that segment. This is achieved, however at the expense of disregarding spatial relationships between points of that segment and points of other segments of the shape. To capture such relationships, an ad-hoc technique would be to apply standard ASM to the centers of mass of each segment. The authors present a more principled framework using the wavelet decomposition of the contour (seen as a 3 separate 1D signal using the x,y and z coordinates of its points).

Instead of the heuristic partition of the spatial domain described above, the wavelet transform provides an elegant way to perform a scale-space decomposition [30]. When decomposing a signal, the wavelet transform produces a scale space decomposition of the signal. At each scale (also called level), a set of wavelet coefficients represent the frequency content of the signal for a particular frequency band. Using this decomposition, the authors propose to use a logarithm tree 2-band wavelet packet to divide the space-frequency domain, as shown in Figure 29 (See Appendix for a brief treatment of the wavelet transform).

The authors apply a wavelet transform to the parametric functions (x,y,z) representing a deformable contour. The resulting wavelet coefficients are then grouped into B bands (in the case of above figure, B=8), and the joint distribution of each band is estimated from the available training samples, via its mean and covariance matrix. This effectively transforms the covariance matrix of the full joint distribution into a matrix that is close to, but not necessarily exactly, a block diagonal matrix, as shown in Figure 30. The submatrix corresponding to Band-1 reflects global shape characteristics, whereas the submatrix corresponding to Band-8 reflects local shape characteristics at a particular segment. Band-7 reflects local shape characteristics of a neighboring segment.

Using this technique, the authors show that a segmentation using the wavelet shape prior is more accurate than a segmentation with traditional active shape models.

4.2.3 Hierarchical Representation for 3D Shapes

In this chapter, we propose to extend this framework in two novel ways. First we describe a multi-scale representation of surfaces in 3D medical imagery using conformal mapping and

spherical wavelets. Spherical wavelets have been used primarily by the computer graphics community to generate multiresolution description of 3D shapes but have not yet been widely used by the medical imaging community[41].

Further, we present a novel algorithm to discover optimal multi-scale bands from the data. Our technique is different from [13] where the authors cluster coefficients of spatially adjacent bases into bands in each frequency plane. In this work, we cluster highly correlated coefficients into a band, with the constraint that coefficients across bands have minimum cross-correlation. Such a decomposition and its visualization can in itself be interesting for shape analysis.

We first present the theory of Spherical Wavelets in Section 4.3. We then explain our shape representation using spherical wavelets in Section 4.4. In section 4.5, we explain our framework to conduct statistical population analysis using spherical wavelets. We derive and compare a number of shape priors based on spherical wavelets to pick the optimal scale-space band decomposition when building the prior, and present our results.

4.3 *Spherical wavelets*

Wavelet basis functions are localized in space and characteristic scales and therefore match a wide range of signal characteristics, from high frequency edges to slowly varying harmonics [?] (See Figure 31). Spherical wavelets are second-generation wavelets adapted to manifolds with non-regular grids [40].

Spherical scaling and wavelet functions are defined on the manifold with a multiresolution grid. One main difference with traditional wavelet construction, is that the functions on the finer grid are not scaled and shifted versions of the function on the coarser grid, although they are similar in shape.

Spherical wavelet functions can be defined on any surface with spherical topology (there exists a one-to-one mapping between the surface and the sphere), with a multiresolution mesh. A multiresolution mesh is created by recursively subdividing an initial mesh so that each triangle is split into four children triangles at each new subdivision. This process is shown on Figure 32. The starting shape can be for example an octohedron (a polyhedron

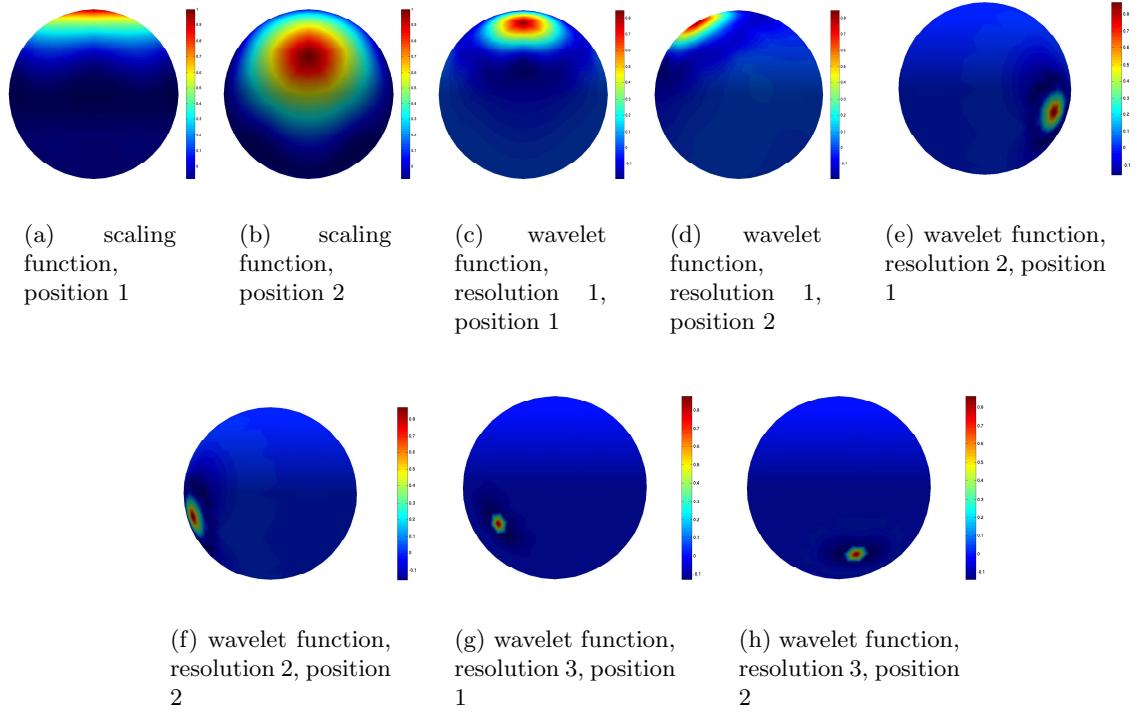


Figure 31: Visualization of Scaling and Wavelet Basis Functions. The color corresponds to the value of the function at each point on the sphere. Notice that the support of each basis gets smaller as the resolution increases

with 8 faces) or an icosahedron (20 faces). The recursive subdivision of an icosahedron is known to provide a good approximation to the sphere [40]. We note that during the registration step described in 4.2.1, the new mesh applied during the remeshing step can be chosen to be a multiresolution mesh similar to the one shown in the last picture of Figure 32.

We now sketch the construction of the spherical wavelet and scaling functions, the analysis step that transforms a signal on the sphere to the wavelet domain and the synthesis step that transforms a signal in the wavelet domain back to a signal defined on the sphere.

4.3.1 Grid Structure for Spherical Wavelet Construction

Spherical scaling functions and spherical wavelets functions are defined on a multiresolution mesh as shown in Figure 32. On this mesh, we denote the set of all vertices obtained after j subdivisions with an index set $K(j)$ (see Figure 33). The $j+1^{\text{th}}$ resolution mesh is obtained by introducing new nodes, identified by an index set $M(j)$ which subdivide existing edges

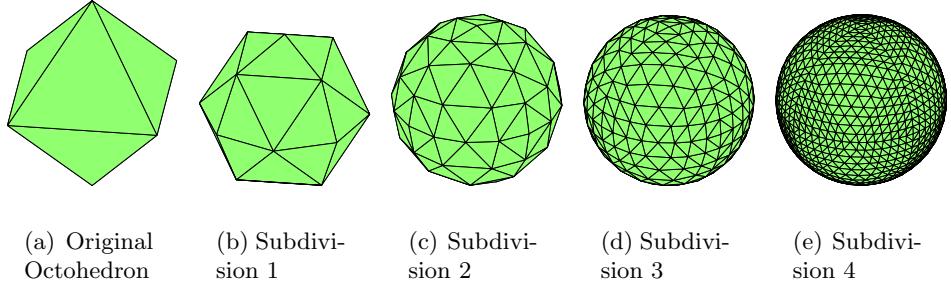


Figure 32: Recursive Partitioning of an octohedron: successive levels of the triangulation are obtained by subdividing triangles into 4 children triangles

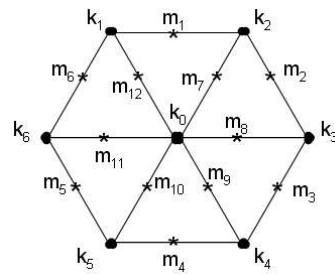


Figure 33: Refinement of the surface grid by subdivision of the edges

(typically at their midpoint, but this is not a requirement). The complete set of nodes in the $j + 1^{th}$ resolution mesh is given by $K(j + 1) = K(j) \cup M(j)$. Figure 33 represents a portion of a triangular surface mesh at resolution $j + 1$, here $K(j) = [k_0, \dots, k_6]$, $M(j) = [m_7, \dots, m_{12}]$ and $K(j + 1) = [k_0, \dots, k_6, m_7, \dots, m_{12}]$.

4.3.2 Scaling Function

At resolution level j , a scaling function is defined for every vertex $k \in K(j)$: $\Phi_{j,k}(\mathbf{x}) : \mathbb{S}^2 \rightarrow \mathbb{R}$. We use the notation $\mathbf{x} = (u, v)$ to denote a point (vertex) on the surface. One commonly used scaling function is the hat scaling $\Phi_{j,k}(\mathbf{x}_k)$ function that varies linearly from the value 1 at vertex \mathbf{x}_k to 0 at neighboring vertices \mathbf{x}_i ($i = 1, \dots, 6$) (See Figure 33). As in the one-dimensional case, the scaling function on the j^{th} resolution mesh can be expressed as

a combination of scaling functions on the $j + 1$ th resolution mesh:

$$\Phi_{j,k}(\mathbf{x}) = \Phi_{j+1,k}(\mathbf{x}) + \sum_{m \in n(j,k)} h_0^j[k, m] \Phi_{j+1,m}(\mathbf{x}) \quad (46)$$

where $n(j, k)$ is the set of neighboring subdivision points at the $j + 1$ level that share an edge with the vertex \mathbf{x}_k at level j (in the Figure 33, $n(j, k) = [m_7, \dots, m_{12}]$). The hat function fits this equation with:

$$h_0^j[k, m] = \begin{cases} \|\mathbf{x}_m - \mathbf{x}'_k\| / \|\mathbf{x}_k - \mathbf{x}'_k\| & m \in n(j, k), k' \in K(j), k' \neq k \\ \|\mathbf{x}_m - \mathbf{x}'_k\| / \|\mathbf{x}_k - \mathbf{x}'_k\| & m \in n(j, k), k' \in K(j), k' \neq k \end{cases} \quad (47)$$

Note that the filter h_0^j becomes much simpler if the edges are divided at their midpoints, since then all the non-zero coefficients in the filter are $\frac{1}{2}$. One can also construct higher-order scaling functions by expanding the neighborhood $n(j, k)$ beyond immediate neighbors, such as the butterfly scheme [40]).

4.3.3 Wavelet Function

At resolution level j , a wavelet function is defined for every vertex $m \in M(j)$: $\Psi_{j,m}(\mathbf{x}) : \mathbb{S}^2 \rightarrow \mathbb{R}$.

$$\Psi_{j,m}(\mathbf{x}) = \Phi_{j+1,m}(\mathbf{x}) - \sum_{k \in A(j,m)} h_1^j[k, m] \Phi_{j,k}(\mathbf{x}) \quad (48)$$

where $A(j, m)$ is a neighborhood of node m . The simplest choice for $A(j, m)$ is to use the two immediate neighbors, e.g. the parents of the subdivision point m .

Each $\Psi_{j,k}(\mathbf{x})$ is therefore a combination of scaling functions at the higher resolution $j + 1$. At level j , they can therefore represent finer details than scaling function $\Phi_{j,\cdot}$.

Wavelet functions capture finer features since they are composed of higher resolution $(j + 1)$ scaling functions. Figures 31 show scaling and wavelet functions for different values of j, k and m . Note that the support of the functions becomes smaller as the resolution increases.

Therefore, the wavelet function can be viewed as a high pass filter, which approximates the signal. The result of the wavelet function is the difference between value calculated

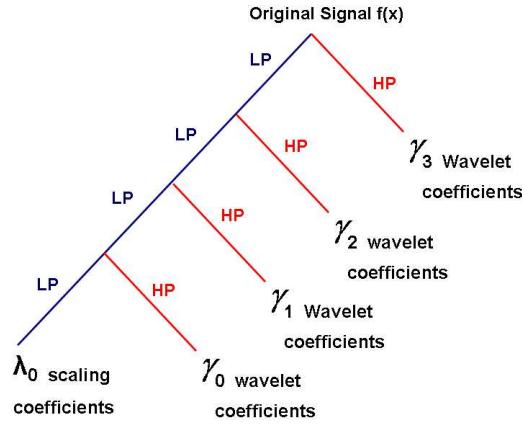


Figure 34: Multi-Scale decomposition of a function using the wavelet transform. The original signal $f(x)$ is decomposed into a series of high-pass and low-pass coefficients. The scaling coefficient represent the low pass information of the original signal. The intermediate wavelet coefficients represent band pass information, and the highest resolution wavelet coefficient γ_3 represents the high pass information.

by the wavelet function and the actual data. The scaling function calculates a smoothed version of the data, which becomes the input for the next iteration of the wavelet function.

4.3.3.1 Spherical Wavelet transform of functions:

The coarsest level scaling function and all wavelet scaling functions construct a basis for the function space L^2 (all functions of finite energy).

$$L^2 = \{\Phi_0, k | k \in K(0)\} \bigcup \{\Psi_j, m | j \geq 0, m \in M(j)\} \quad (49)$$

A function defined on the sphere $f(\mathbf{x}) : \mathbb{S}^2 \rightarrow \mathbb{R}$, can be represented in this basis by:

$$f(\mathbf{x}) = \sum_{k \in K(0)} \lambda_{0,k} \varphi_{0,k}(\mathbf{x}) + \sum_{0 \leq j} \sum_{m \in M(j)} \gamma_{j,m} \psi_{j,m}(\mathbf{x}). \quad (50)$$

$\lambda_{0,k}$ are called the scaling coefficients and $\gamma_{j,m}$ are the wavelet coefficients. This can also be written in matrix notation:

$$f(\mathbf{x}) = \Phi \Gamma \quad (51)$$

where if \mathbf{x} is of size $N \times 1$, Γ is the vector of coefficients of size $N \times 1$ and Φ is the matrix of basis functions of size $N \times N$, where each column of Φ is a basis function.

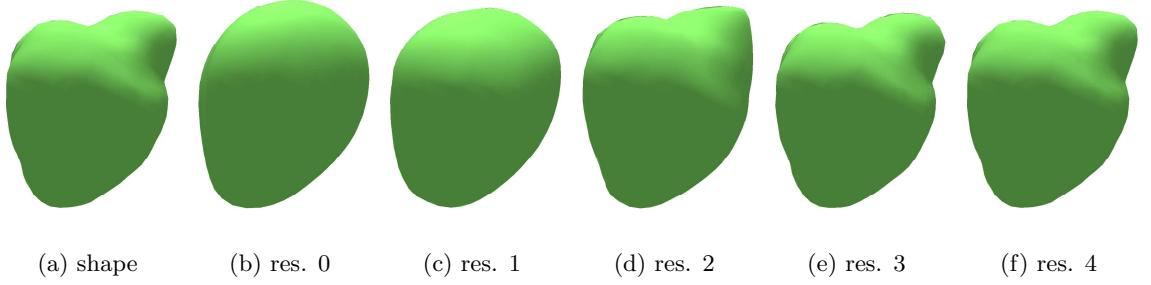


Figure 35: Shape at various resolution levels, see text for details

The scaling coefficient represent the low pass information of the original signal. The intermediate wavelet coefficients represent band pass information, and the highest resolution wavelet coefficient γ_3 represents the high pass information. This is shown graphically in Figure 34

When a function is transformed into the wavelet domain, its wavelet coefficients are calculated by inner product with dual functions (see [40] for more details.). We use the biorthogonal set $\{\psi_{j,m}(\mathbf{x}), \tilde{\psi}_{j,m}(\mathbf{x}), \varphi_{j,k}(\mathbf{x}), \tilde{\varphi}_{j,k}(\mathbf{x})\}$ [40] to find coefficients for the scaling functions $\lambda_{j,k}$ given by inner product $\langle f(\mathbf{x}), \tilde{\varphi}_{j,k}(\mathbf{x}) \rangle$ and wavelet coefficients $\gamma_{j,m}$ given by inner product $\langle f(\mathbf{x}), \tilde{\psi}_{j,m}(\mathbf{x}) \rangle$.

4.4 Shape Representation using spherical wavelets

To represent a population of shapes using the spherical wavelet transform, we first need each shape in the population to have a multiresolution mesh, as defined in Section 4.3. To achieve this, we remesh each shape with the required mesh using the registration technique described in Section 4.2.1. This provides a one-to-one correspondence between shapes, allowing us to calculate a mean shape. We then build our scaling/wavelet functions *on the mean shape*, using the technique described in the previous section. This creates a set of basis functions adapted to the topology of the mean shape, and therefore more specific to each shape in the population than if we had used bases functions built on a sphere.

We then represent each shape in the population by encoding the deviation from the mean using the spherical wavelet transform.

We first subtract the mean shape from all the shapes in the set. We then apply the wavelet transform independently to the residual x, y and z coordinates of the N vertices of the shape. The k^{th} shape in a population of K shapes can then be described by a vector of wavelet coefficients of size $3N$:

$$\Gamma_k = \{\lambda_{0,k}^x, \lambda_{0,k}^y, \lambda_{0,k}^z, \gamma_{j,m}^x, \gamma_{j,m}^y, \gamma_{j,m}^z | j = 0, \dots, 4; m \in M(j); k \in K(0)\} \quad (52)$$

Figure 35 shows a wavelet decomposition of a prostate. Figure 35(a) is the shape before decomposition. Each Figure 35(b)- 35(f) is the mean shape plus the cumulative signals up to that resolution. We observe more high frequency content as the resolution increases.

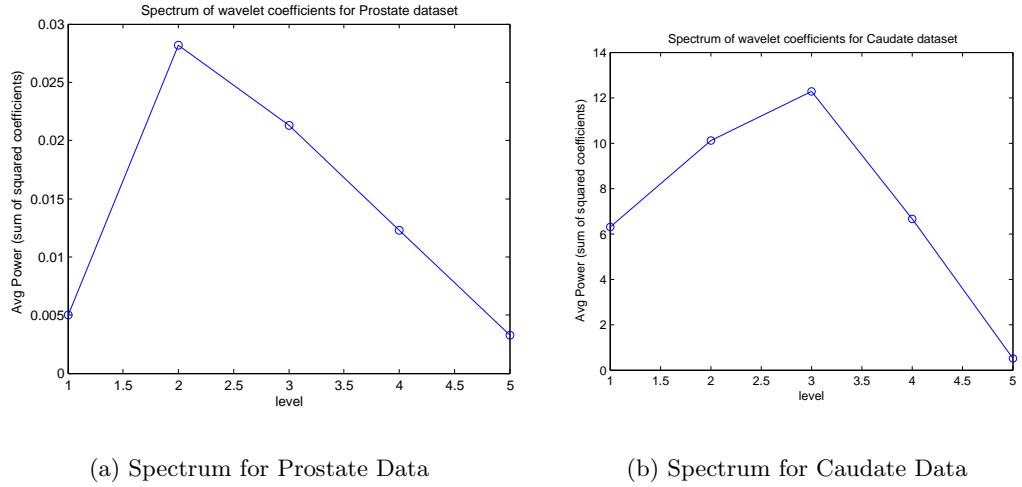
One advantage of using wavelets is that if $f(\mathbf{x})$ is mostly smooth, with a few local features located at different scales and spatial locations, then many of the wavelet coefficients will be small, except for the ones whose corresponding wavelet function is located close to the feature both in space and scale. This means that the wavelet corresponding to the smaller coefficients can be dropped from the expansion altogether without significantly affecting the function approximation. This leads to a compression property since the transform can match a variety of shape signals using a small number of bases.

4.4.1 Spectrum Analysis

A power spectrum can be calculated from the result of a wavelet transform. Plotting the power spectrum provides a useful graphical representation for analyzing the frequency content of the signal [?]. The wavelet power spectrum is calculated by summing the squares of the coefficient values for each resolution level j :

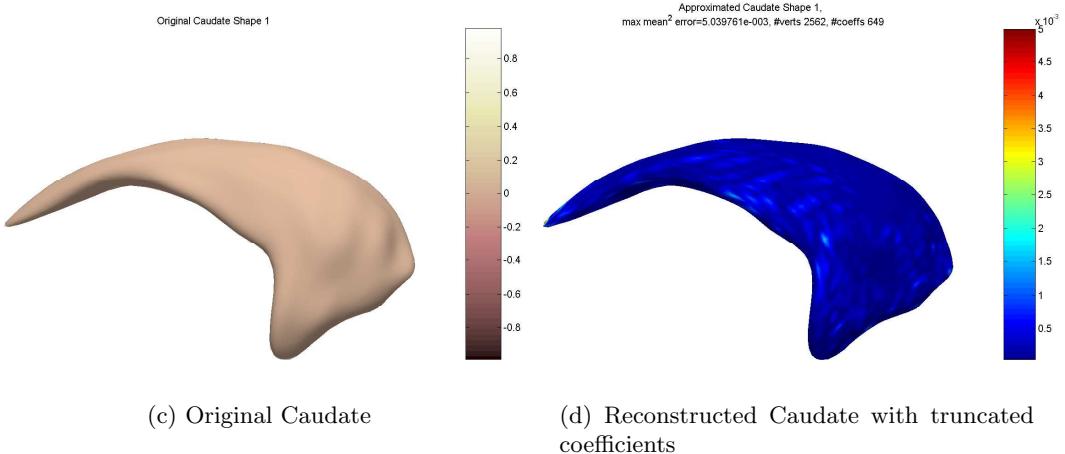
$$\text{spectrum}(j) = \sum_m \gamma_{j,m}^x + \gamma_{j,m}^y + \gamma_{j,m}^z \quad (53)$$

Figure 36(a) shows the average wavelet power spectrum for the prostate data and Figure 36(b) shows the average wavelet power spectrum for the caudate data. For each plot, the resolution level is plotted on the x-axis, from 1 to 5. The y-axis plots the spectrum power according to the above equation. The resolution level farthest out on the x-axis (plotted at x-axis point 5) contains differences between the wavelet function and the signal.



The better the approximation provided by the wavelet function, the smaller these values will be. We see that the spherical wavelet with the butterfly interpolation scheme that we are using provides a good approximation since band 5 has a small value. The peaks in the power spectrum plot also show at which frequency level the signal has the most power. We notice that the prostate dataset peaks at level 2, while the caudate dataset peaks at level 3, indicating that the prostate surfaces are generally “smoother” and most of the details on the surface appear at a coarser scale than for the caudate dataset.

Using the power spectrum, we can also *compress* the signal by removing the coefficients



with the least power. A common nonlinear approximation technique is to rank the coefficients by their power (square of each coefficient), and truncate (set value to zero) the last $x\%$ coefficients, where $x = 2$ or 5 typically [?]. In the spherical harmonics case, the coefficients corresponding to the highest frequency band are typically removed, since removing coefficients that correspond to coarser scale affects the full surface due to the global support of the harmonic functions. As a comparison, the truncation in the spherical wavelet case can remove coefficients at any scale due to their local support. Figure 36(d) shows a shape reconstructed after truncation, where 74% of the coefficients were truncated (using a truncation level of 5%). The mean error between the reconstructed surface and the ground truth (shown in Figure 36(c)) is 5×10^{-3} .

4.5 Scale-Space Spherical Wavelet Prior for Statistical Population Analysis

We adopt a hierarchical statistical approach to wavelet-based shape analysis, as in described in Section 4.2.2.1. A shape described by a vector of wavelet coefficients is then a random realization from a multivariate shape probability distribution $P(S)$.

As previously described, Davatzikos et. al. [13] used 1D wavelet to analyze shape contours with a scale-space decomposition of the wavelet coefficients to build a more accurate prior. Recall that the basic idea is to group wavelet coefficients at the same scale that are close in space using a tree decomposition shown in Figure 29. This means that the scaling coefficients are grouped in one band, wavelet coefficients of level $j = 0$ are grouped in $2^0 = 1$ band, coefficients of level $j = 1$ are grouped in 2^1 bands etc. To do this at level j , the contour is split into 2^j subsegments and the coefficients that are associated with the wavelet functions centered on a subsegment are grouped in a band. The splitting of the curve is arbitrary, as long as the subsegments have the same length.

The 3D equivalent is to group spherical wavelet coefficients by resolution level and by spatial proximity. This can be done by “cutting” the mean surface into N parts, where N is the desired number of bands. However, we propose a more principle approach where for each scale, we cluster highly correlated coefficients into a *band*, with the constraint that

coefficients across bands have minimum cross-correlation.

4.5.1 Adaptive Band Selection by scale-space decomposition

In this section, we describe our technique to cluster coefficients according to correlation to pick meaningful bands that indicate areas of variation. Such a decomposition can in itself be interesting for shape analysis.

To cluster correlated wavelet coefficients, we use a spectral graph partitioning technique [43]. We use a fully connected undirected graph $G = (V, E)$ where nodes V are wavelet coefficients for a particular scale. The weight on each edge $w(i, j)$ is a function of similarity between nodes i and j . To set $w(i, j)$, we calculate the covariance between the wavelet coefficients at nodes i and j for the x , y and z signal of the shape and set $w(i, j)$ to be the average covariance for all three signals. A cut

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (54)$$

is the optimal partitioning of V into two disjoint sets A and B such that nodes within a partition have the highest covariance and nodes across partitions have the lowest covariance. An optimal partitioning of the nodes is one that minimizes equation 54. To prevent unbalanced partitions, Shi and Malik [43] define the *normalized cut*. Given a partition of nodes of a graph V into 2 sets A and B , let x be an $N = |V|$ dimensional indicator vector, $x_i = 1$ if node i is in A and -1 otherwise. The normalized cut is then defined as:

$$\text{Ncut} = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(B, A)}{\text{assoc}(B, V)} \quad (55)$$

where $\text{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph. The solution to the normalized cut can be found by solving a generalized eigenvalue problem. More details about this technique can be found in [43].

Using this technique, we do a recursive partitioning of the coefficients into bands. We use a stopping criteria based on the quality of the decomposition of each set, validating whether the subdivided band correspond to two independent distributions. Indeed, if we start with a graph G where we consider each coefficient to be a random variable and find a partition $A \cup B = G$, $A \cap B = \emptyset$, then it is a good partition if $P(G) = P(A)P(B)$. We

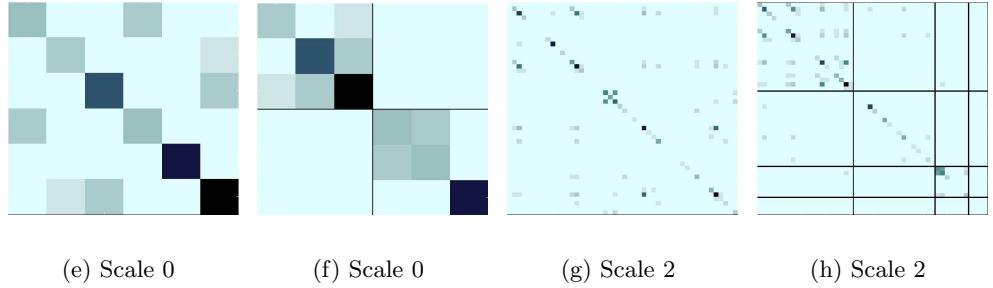


Figure 36: Covariances Matrices, see text for detail

can test this with the Kullback-Leibler (KL) divergence between the joint and the product of the marginals [35].

In this work, we assume a multivariate Gaussian distribution for each partition and derived the KL divergence [35]:

$$D(P(G)||P(A)P(B)) = 1/2 \log(|\Sigma_A||\Sigma_B|) - 1/2 \log(|\Sigma_G|) \quad (56)$$

where Σ_A is the covariance matrix of $P(A)$ and $|.|$ is the determinant ². If the distributions $P(A)$ and $P(B)$ are independent, then their KL divergence is 0. In practice, we do not accept a partition if $D(P(G)||P(A)P(B)) > 0.1$.

4.5.2 Decomposition Results

Figure 36 shows the result of this recursive decomposition for the Prostate Data. Figure 36(e) shows the covariance matrix for the coefficients at scale 0, ordered by their (arbitrary) index. The darker the entry, the more correlation. In Figure 36(f) the coefficients are re-ordered according to the output bands, where the black lines indicate the transition between bands. The intra-band variation is shown by the diagonal blocks. As expected, the off-diagonal inter-band covariance is minimal. The same is shown for resolution 2 in Figures 36(g)- 36(h). There are various band sizes, due to the fact that new bands are only recursively divided if eq. 56 is close to zero.

²In practice since Σ_A is often singular, we decompose it using SVD as $\Sigma_A = U\Sigma'_AU^T$ and estimate $\log|\Sigma_A| = \text{trace}(\log(\Sigma'_A))$, using only the positive entries of Σ'_A

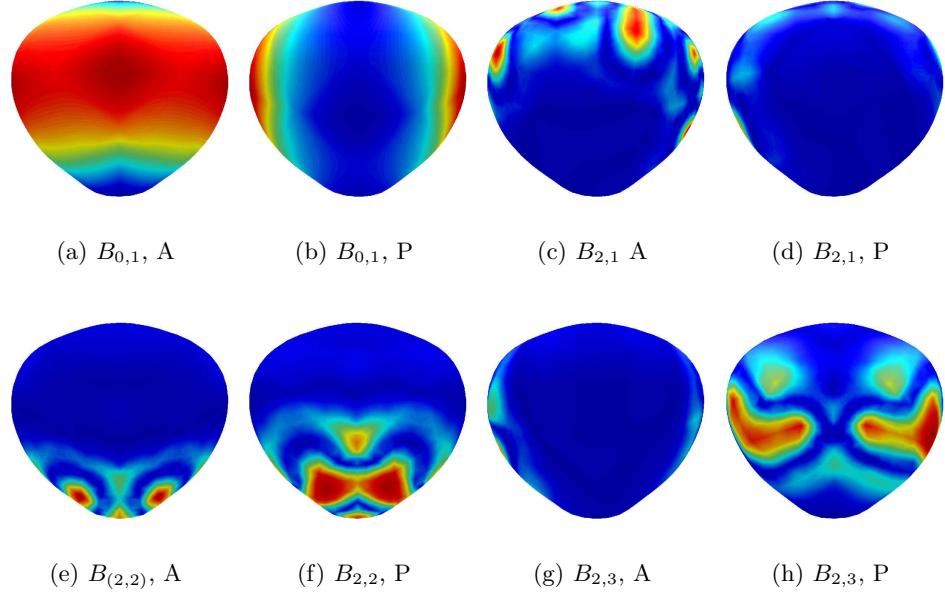


Figure 37: Band Decomposition: various bands $B_{j,i}$, where j is the resolution and i is the band number, shown in Anterior view (A) and Posterior view (P), see text for color

4.5.3 Band Visualization

To visualize the bands, we calculate the influence of all wavelet coefficient in band $B_{j,i}$ on each point \mathbf{x} of the surface. To do so, we set $\gamma_{j,m} = 1$ if $m \in B_{j,i}$ and 0 otherwise and apply the inverse wavelet transform given in Equation 50 (where $f(\mathbf{x}) : \mathbb{S}^2 \rightarrow \mathbb{R}$ is a function that measures the influence of the non-zero coefficients on a surface point \mathbf{x}).

If $f(\mathbf{x}) = 0$, then the point is not affected and if $f(\mathbf{x}) > 0$ it is affected according to the magnitude of $f(\mathbf{x})$. Using this function as a colormap (blue= 0, red= 1), Figures 37(a)-37(b) show the first band for the lowest scale. The second band is the complement of the first. As expected each band has a large spatial extent and indicate two uncorrelated shape processes on the prostate data: the variation of the anterior wall of the prostate (typically rounded) and the variation of the posterior wall of the prostate (typically flatter). Figures 37(c)- 37(h) show three bands for the scale 3. These bands are more localized. These are uncorrelated variations of the superior and inferior walls of the shape, as well as an uncorrelated variations of the anterior wall at that scale. Bands have compact support,

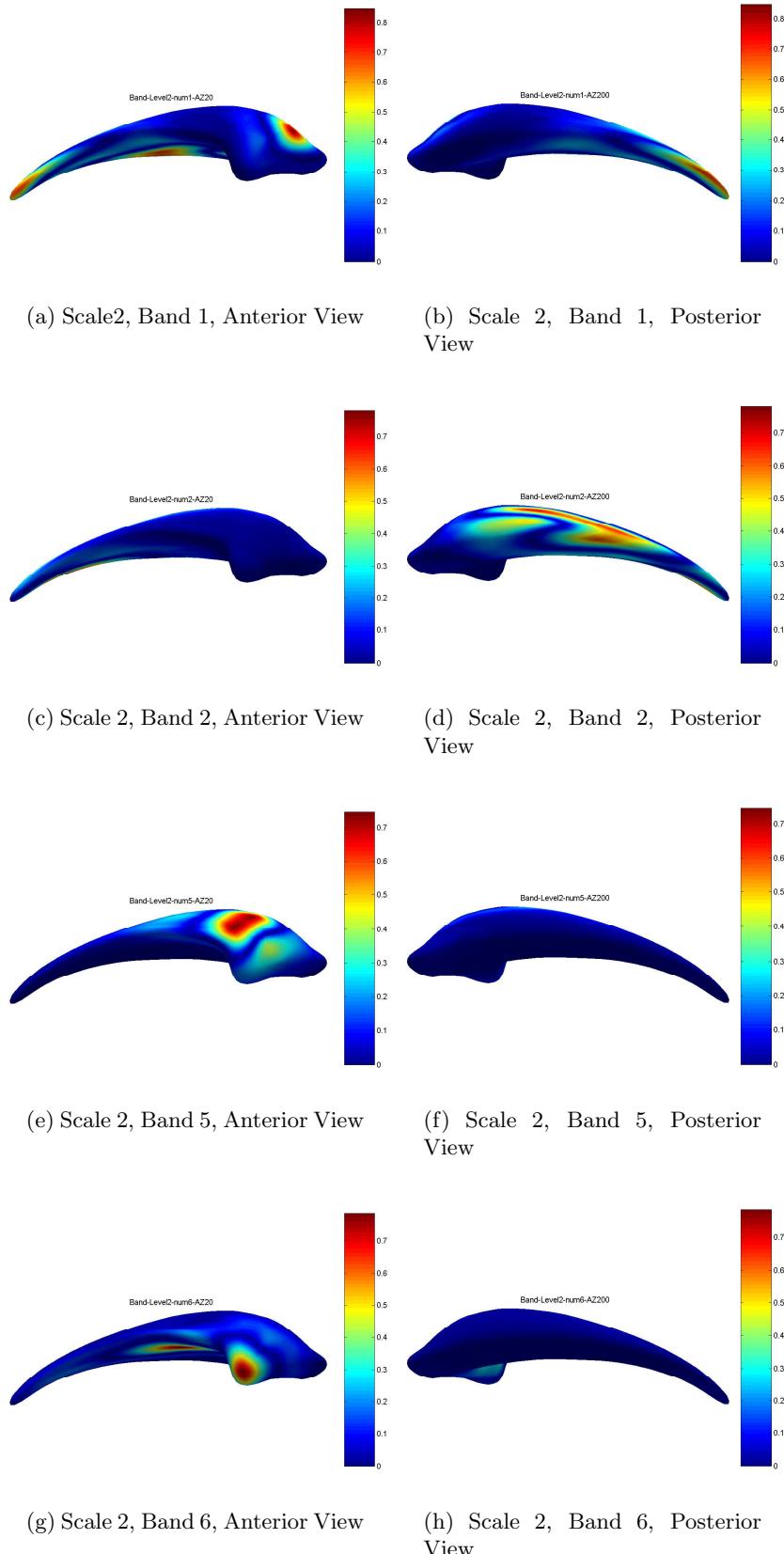


Figure 38: Band Decomposition, Caudate Nucleus dataset: various bands shown for scale 2 (shown in Anterior view and Posterior view), see text for color

though this is not a constraint of our technique. The symmetry in bands 2 and 3 is also interesting, showing that both the right and left side tend to co-vary similarly. This symmetry of variation is plausible for the prostate, and we plan to investigate this further. Notably a diseased organ could possibly be detected if there is a lack of symmetry. Figures 38 show the band decomposition of the caudate population for scale 2. Again, the bands seem to be well localized in space.

4.5.4 Building the Prior

Each band of coefficients is modeled as a multivariate Gaussian that encodes a set of shape variations. To estimate their distribution from the data, we apply PCA analysis to the coefficients in each band, as shown in Algorithm 1, thus obtaining a total of $b(N - 1)$ bands, where b is the number of bands and N is the number of shapes in the training set. The eigenvectors and eigenvalues of lower scale bands represent relatively global aspects of shape variability, whereas bands at higher scales represent higher frequency and more localized aspects of shape variability. Hence, our technique discovers shape processes *at every scale*, where the processes are all the eigenvectors of all the bands, and does not favor the discovery of global processes over local processes. Additionally, our prior accurately encodes finer details even with small training sets, since there exists at least $b(N - 1)$ eigenvectors, as opposed to just $N - 1$ eigenvectors when PCA is conducted directly on the vertices of the shape.

4.6 Experiments

In this Section, we compare the previously presented shape prior based on spectral partitioning to standard ASM, and other scale-space decompositions for a reconstruction task. We first describe the types of shape priors that we are comparing, then the reconstruction task and finally the results.

4.6.1 Shapes Priors used in the experiment

With our experiment, we wanted to achieve three goals:

Algorithm 1 Algorithm for Building Shape Prior

```
1: Calculate the mean  $\bar{S}$  from all shapes in the population
2: for each shape  $S_k$  in the population do
3:    $\tilde{S}_k = S_k - \bar{S}$ 
4:    $\Gamma_k^x = \text{Spherical\_Wavelet\_Transform}(\tilde{S}_k^x)$  where  $S^x$  is the x coordinate of S
5:    $\Gamma_k^y = \text{Spherical\_Wavelet\_Transform}(\tilde{S}_k^y)$  where  $S^y$  is the y coordinate of S
6:    $\Gamma_k^z = \text{Spherical\_Wavelet\_Transform}(\tilde{S}_k^z)$  where  $S^z$  is the z coordinate of S
7:    $\Gamma_k = [\Gamma_k^x; \Gamma_k^y; \Gamma_k^z]$ 
8:   (optional) truncate  $\Gamma_k$  to keep highest  $x\%$  squared coefficients
9:   for each scale  $j$  do
10:    cluster the wavelet coefficients  $\Gamma_{k,j}$  into  $b$  bands ( $B_{j,i} | i \in 1, \dots, b$ )
11:    for each band  $B(j,i)$  do
12:      Calculate the mean  $\bar{B}_{j,i}$  and covariance matrix  $\frac{1}{n} B_{j,i} B_{j,i}^T$  where  $n$  is the number
         of coefficients in band  $B_{j,i}$ 
13:      Apply PCA thus obtaining an eigenvector matrix  $U^{(j,i)}$  and eigenvalues  $\Sigma^{(j,i)}$ 
14:    end for
15:  end for
16: end for
```

1. Compare the prior based on the spherical wavelet spectral partitioning scale-space decomposition (WAV-SP) to other wavelet scale-space decompositions
2. See the impact of coefficient truncation on the accuracy of the wavelet shape priors
3. Compare the best wavelet shape prior to standard ASM

To achieve goal 1 and 2, we use the following scale-space decompositions of the spherical wavelet coefficients:

- WAV-SP: spectral partitioning scale-space decomposition presented in section 4.5 without truncation
- WAV-SP-trunc: spectral partitioning scale-space decomposition presented in section 4.5 with a 2% truncation level
- WAV-1band: no scale-space decomposition, all coefficients are placed in 1 band
- WAV-1band-trunc: no scale-space decomposition with 2% truncation, *non-truncated* coefficients are placed in 1 band
- WAV-scale : only scale decomposition, coefficients from the same scale are grouped in 1 band

- WAV-scale-trunc : only scale decomposition with 2% truncation, *non-truncated* from the same scale are grouped in 1 band
- WAV-prev: scale-space decomposition following [13], where the coefficients are clustered in bands at each scale according to Figure 29. At each scale, we cluster coefficients arbitrarily by using a recursive subdivision of the surface points according to their mapping to the sphere. The first subdivision is along the plane that connects the 0 and 180 meridian. The second subdivision is along the plane that connects the 90 and 270 meridian. The third subdivision is along the plane that cuts through the equator.
- WAV-prev-trunc: similar to WAV-prev, but with a 2% truncation.

For each decomposition, the shape prior consists of a set of eigenvectors and eigenvalues for each band. To achieve goal 3, we compare all the WAV techniques to ASM for the task of reconstruction.

4.6.2 Test Shape Reconstruction

In this experiment, we partition our dataset of N samples randomly into T training samples and $N - T$ testing samples, where $T = [5, 10, 25]$ and learn a shape prior from the training samples for each technique described in the previous section.

To test the priors, we project the test shape s unto the eigenvector U_b for each band B (where b is the band number):

$$\alpha_b = U_b^T(s - \bar{B}) \quad (57)$$

and ensure that the projection lies within ± 2 standard deviation of the values observed for the training set:

$$\alpha'_b(m) = \begin{cases} \alpha_b(m) & \text{if } -3\sqrt{(\sigma_m)} \leq \alpha_b(m) \leq 3\sqrt{(\sigma_m)} \\ -3\sqrt{(\sigma_m)} & \text{if } \alpha_b(m) < -3\sqrt{(\sigma_m)} \\ 3\sqrt{(\sigma_m)} & \text{if } \alpha_b(m) > 3\sqrt{(\sigma_m)} \end{cases} \quad (58)$$

s_n is corrected to s'_n by applying

$$\alpha'_b = \bar{B} + U_b \alpha'_n \quad (59)$$

Note that for ASM, there is only 1 band that contains the (x, y, z) coordinates of the shape. For the spherical wavelet priors, the number of bands depends on the type of decomposition and the bands contain the wavelet coefficients.

To test the accuracy of each prior, we calculate an average mean squared error between each reconstructed test shape s' and the ground truth test shape s .

To test the robustness of each prior, we also test the reconstruction in the presence of noise. To add noise to the test shape, we displace each vertex according to a Gaussian distribution with mean 0 and a standard deviation that is 5% of the bounding box of the object, as shown in Figure 43(b), producing a shape with noise s_n . Ideally, we would want the prior to not be affected by the noise and the reconstructed shape to be close to the ground truth (the shape without noise). To test this, we project the noisy shape s_n onto the priors, and calculate the mean squared error between the reconstructed shape s' and the *ground truth* shape s .

4.6.3 Results

Figure 39 shows the mean squared reconstruction error and max error, averaged over all the shapes in the training set, for the various shape priors and various training set sizes of the Prostate dataset. Figure 40 shows the same information for the Caudate dataset. As we can see in all graphs, both the WAV-SP and WAV-SP-trunc outperform the other techniques, even when the training set size is large (25 shapes). It is also interesting to see that if we compare a particular decomposition with and without truncation, the error is approximately the same. This means that truncation does not affect the accuracy of the prior. This allows us to use truncation to have an even more compact set of parameters that remains descriptive enough.

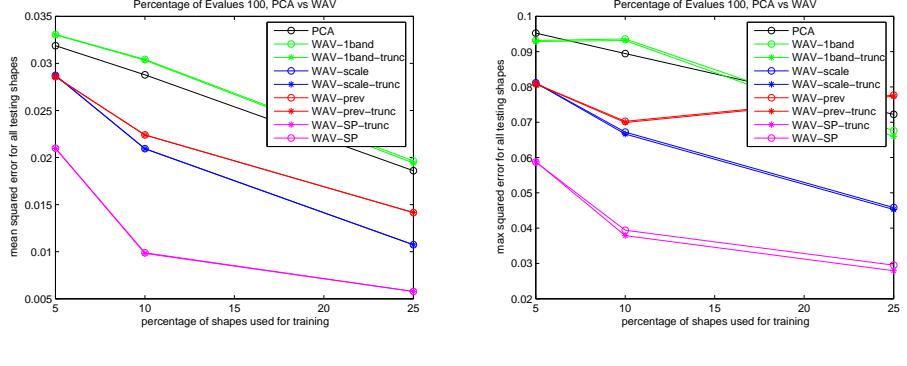
Figure 41 and Figure 42 show the effect of using a noisy shape as a projection onto the priors, but then comparing the reconstruction to the original ground truth, to see if

the prior effectively suppresses noise. As we can see, the results are very similar to the case without noise, and the WAV-SP-trunc prior outperforms the other priors. It is also interesting to see that both PCA and the wavelet priors are not affected by Gaussian noise. Therefore, although the WAV-SP-trunc prior is more specific than PCA (meaning it represents a population more accurately), it is not more sensitive to noise.

As an example, Figures 43 and 44 show the Ground Truth shape (GT), Noisy shape (Noise), and reconstruction with PCA and Wavelet shape priors with **5 training samples** for the prostate dataset and **25 training samples** for the caudate dataset. The figures show the reconstruction when both the **GT** is projected onto the prior and the **Noise** is projected onto the prior. We see that details are lacking in the PCA reconstruction. The WAV-SP technique incorporates both local and global details that PCA does not encode.

4.7 Conclusions

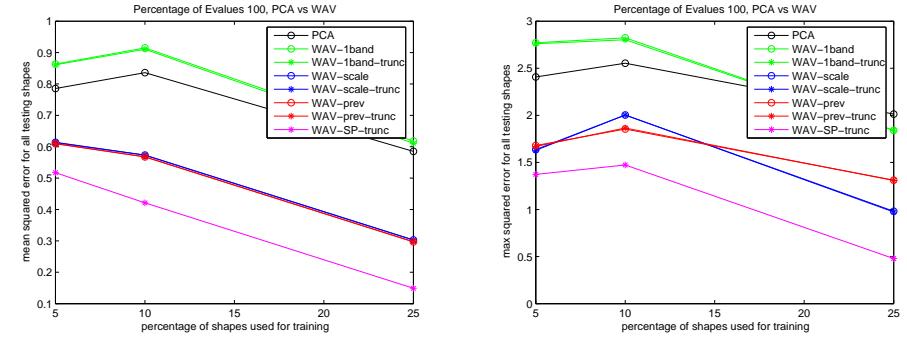
We have demonstrated that our spherical wavelet based technique is a better shape prior than ordinary ASM when it is important to represent finer, more localized shape variation. Our novel method finds independent shape variation processes at multiple scales and multiple locations by adaptively clustering correlated wavelet coefficients. From our results we have shown that this technique outperforms other types of scale-space decompositions, as well as standard ASM. We have also shown that coefficient truncation does not impact the accuracy of the wavelet shape priors, therefore allowing us to have a more compact shape representation. In our proposed work, we will explain how we plan to use this shape representation and prior into an active surface segmentation framework.



(a) Mean Squared Reconstruction Error, averaged over training shapes

(b) Max Error, averaged over training shapes

Figure 39: [Prostate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the **Normal** Test Shapes



(a) Mean Squared Reconstruction Error, averaged over training shapes

(b) Max Error, averaged over training shapes

Figure 40: [Caudate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the **Normal** Test Shapes

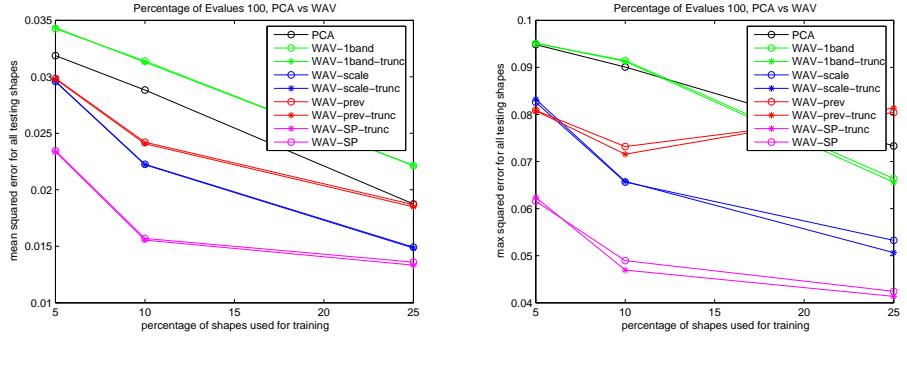


Figure 41: [Prostate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the **Noise** Test Shapes

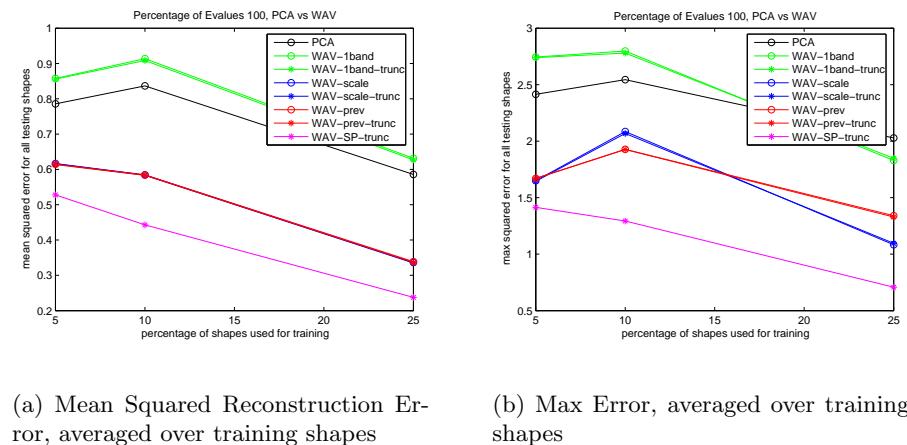


Figure 42: [Caudate Dataset] Mean Squared Reconstruction Error and Max Error for various training set sizes for the **Noise** Test Shapes

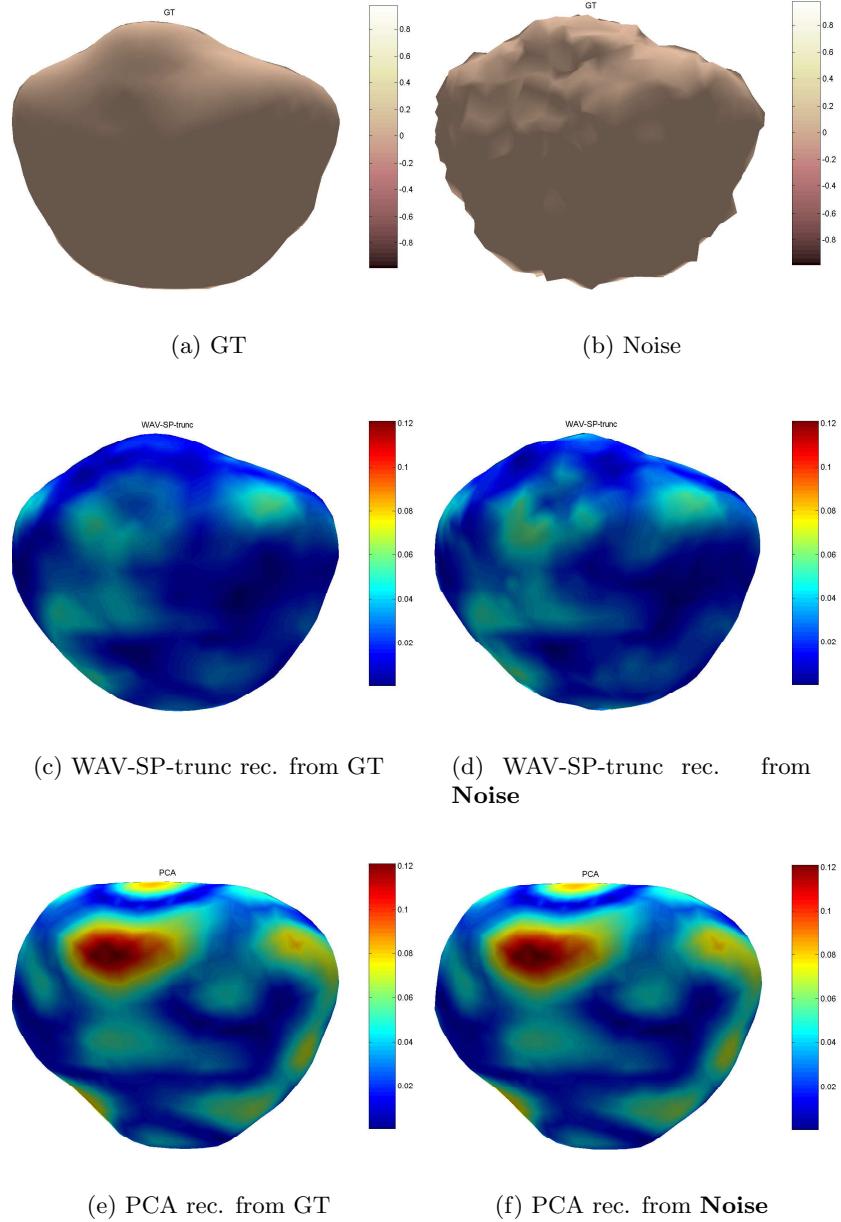


Figure 43: Prostate dataset: Ground Truth (GT), Noisy Shape (Noise), and reconstruction with PCA and Wavelet shape prior with **5 training samples** when both the **GT** is projected onto the prior and the **Noise** is projected onto the prior. Color is error from blue (lowest) to red.

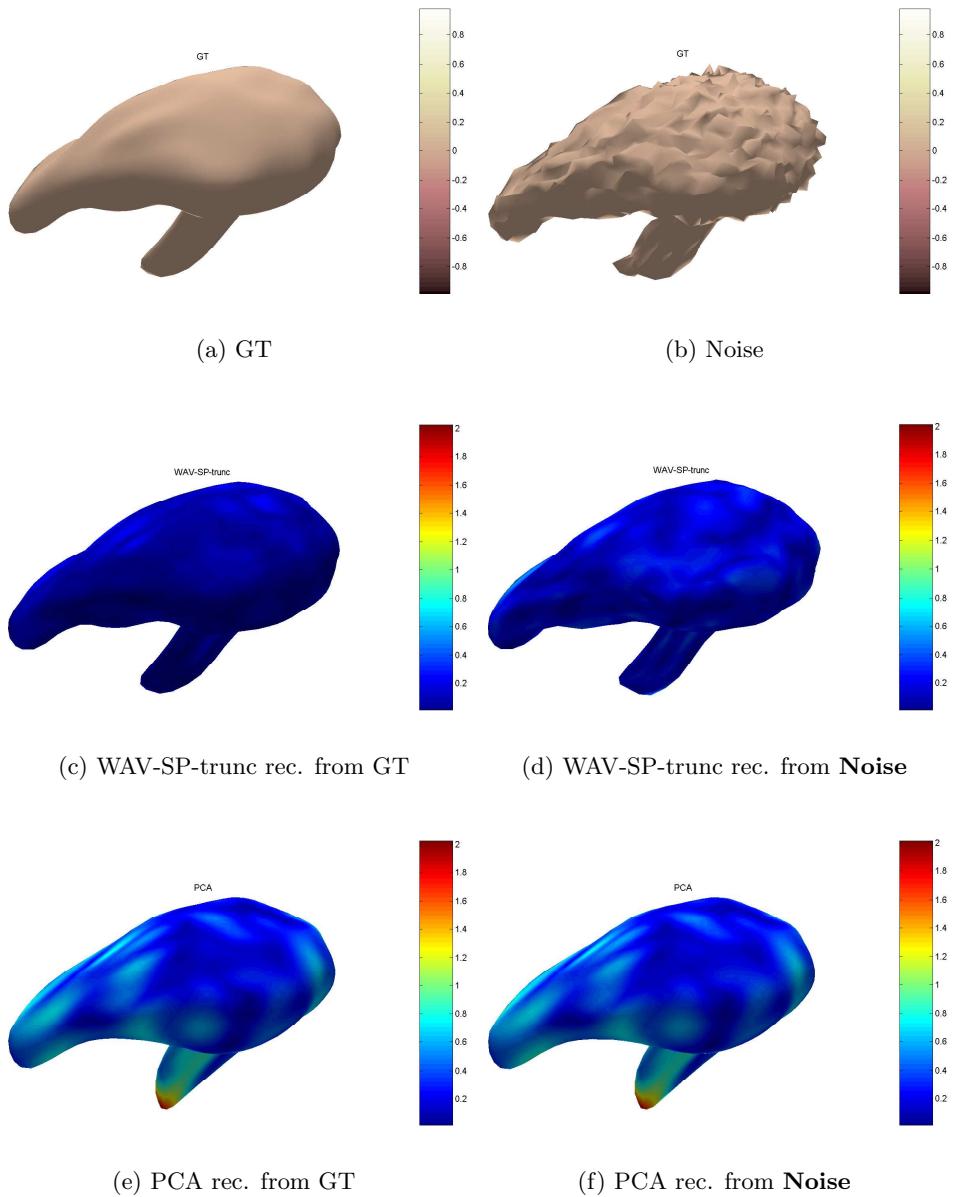


Figure 44: Caudate dataset: Ground Truth (GT), Noisy Shape (Noise), and reconstruction with PCA and Wavelet shape prior with **25 training samples** when both the **GT** is projected onto the prior and the **Noise** is projected onto the prior. Color is error from blue (lowest) to red.

CHAPTER V

PROPOSED WORK

For this thesis, we propose the following three remaining research topics:

1. Segmentation of medical imagery using spherical wavelets
2. Testing the robustness of the shape registration and its influence on the segmentation results
3. Shape classification using spherical wavelets

We now give more detail on each topic.

5.1 Segmentation of medical imagery using spherical wavelets

For this project, the end goal is to derive a parametric surface evolution equation by evolving the wavelet coefficients directly so that we can include the shape prior defined in chapter 4 directly in the flow (recall that the shape prior is a probability distribution defined over the wavelet coefficients themselves). We will derive a surface evolution equation by minimizing an energy function based on image forces as well as a shape prior defined in the wavelet model space. The resulting algorithm should be computationally efficient due to the low number of parameters and should encode higher frequency shape variations when compared to active shape models due to the spherical wavelet shape prior.

The parameters of our model will be shape parameters, as well as the parameters of a similarity transformation, called pose parameters.

5.1.1 Pose parameters

Given a surface $\vec{\mathcal{S}}(u, v) : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$, a transformed surface of $\vec{\mathcal{S}}$, based on the pose parameters \mathbf{p} , is denoted $\tilde{\vec{\mathcal{S}}}$, and is defined as:

$$\tilde{\vec{S}}(u, v) = \begin{bmatrix} \tilde{x}(u, v) \\ \tilde{y}(u, v) \\ \tilde{z}(u, v) \\ 1 \end{bmatrix} = T[\mathbf{p}] \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \\ 1 \end{bmatrix} \quad (60)$$

where

$$T[\mathbf{p}] = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \exp(\Omega(w_x, w_y, w_z)) \quad (61)$$

The transformation matrix $T[p]$ is the product of three matrices: a translation matrix, a scaling matrix and a rotation matrix. The formulation for the rotation matrix, $\exp(\Omega)$, is the exponential map formulation that requires only three parameters to describe a rotation [27]. Let $\vec{w} = [w_x, w_y, w_z]^T$ be a 3D vector and $\theta = \|\vec{w}\|$ be its norm. An angular rotation θ around an axis of direction \vec{w} can be represented by:

$$\begin{aligned} \exp(\Omega) &= I + \Omega + \frac{1}{2!}\Omega^2 + \frac{1}{3!}\Omega^3 + \dots \\ \exp(\Omega) &= I + \frac{\sin\theta}{\theta}\Omega + \frac{(1-\cos\theta)}{\theta^2}\Omega^2 \end{aligned} \quad (62)$$

where Ω is the skew-symmetric matrix:

$$\Omega = \begin{bmatrix} 0 & -w_z & w_y & 0 \\ w_z & 0 & -w_z & 0 \\ -w_y & w_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (63)$$

This formulation avoids the gimbal lock problem of Euler angles and does not require an additional constraint to preserve the norm of a quaternion.

5.1.2 Surface parameters using spherical wavelets

As we have seen in chapter 4, the surface \vec{S} can be represented in the wavelet basis function using the mean population shape $\overline{\vec{S}}(u, v)$ plus a vector of wavelet coefficients Γ :

$$\vec{\mathcal{S}}(u, v) = \overline{\vec{\mathcal{S}}(u, v)} + \Phi(u, v)\Gamma \quad (64)$$

where Γ is the vector of coefficients of size $N \times 3$ and Φ is the matrix of basis functions of size $N \times N$, where each column of Φ is a basis function. Here therefore, the shape parameters are the Γ coefficients.

5.1.3 Proposed Surface Evolution Derivation: case with no prior

We will first derive a region-based surface evolution equation without the shape prior, using the notation of Section 2.4.4, and adapting the derivation in Appendix A.1.

Recall that the basic underlying mathematical idea is to write down an energy that minimizes a particular function f inside a region R , enclosed inside the surface $\vec{\mathcal{S}}$:

$$E(\vec{\mathcal{S}}) = \int_R f(\mathbf{x}) d\mathbf{x} = \oint_{\vec{\mathcal{S}}} \langle F, S_u \times S_v \rangle du dv \quad (65)$$

In the derivation, we will represent the surface $\vec{\mathcal{S}}$ as:

$$\vec{\mathcal{S}}(\mathbf{p}, \Gamma) = T[p] \begin{bmatrix} \bar{x}(u, v) + \Phi(u, v)\Gamma_x \\ \bar{y}(u, v) + \Phi(u, v)\Gamma_y \\ \bar{z}(u, v) + \Phi(u, v)\Gamma_z \\ 1 \end{bmatrix} \quad (66)$$

where the mean surface $\overline{\vec{\mathcal{S}}} = (\bar{x}, \bar{y}, \bar{z})$ and Γ_x represents the x portion of the wavelet coefficients.

The goal is then to find a gradient flow of $\vec{\mathcal{S}}$ that minimizes the previously defined energy, *in terms of the pose \mathbf{p} and shape parameters Γ* .

We will derive $\frac{dE}{dp}$ and $\frac{dE}{d\Gamma}$. We will then conduct a parameter optimization via gradient descent, in similar spirit to [50]. The update equations for the shape and pose parameters in the gradient descent approach are given by:

$$\begin{aligned} \Gamma^{t+1} &= \Gamma^t - \Delta_\Gamma \frac{dE}{d\Gamma} \\ \mathbf{p}^{t+1} &= \mathbf{p}^t - \Delta_\mathbf{p} \frac{dE}{d\mathbf{p}} \end{aligned} \quad (67)$$

where Δ_Γ and $\Delta_{\mathbf{p}}$ are positive step size parameters, and Γ^{t+1} , \mathbf{p}^{t+1} denote the values of the parameters Γ and \mathbf{p} at the $t + 1^{th}$ iteration, respectively. The updated shape and pose parameters will then be used to implicitly determine the updated location of the segmenting surface.

It is important to note that no special numerics are required as the update equations do not involve partial differential equation, only ordinary differential equations over a limited parameter set. This should result in a fast and simple implementation of the surface evolution.

5.1.4 Proposed Experiments for surface evolution with no shape prior

1. We will first test the surface evolution with no shape prior on a synthetic binary labelmap. In this case, the function f to minimize is -1 inside an object in the labelmap and 1 outside the object. Therefore, to minimize the energy, the surface should fit at the boundary of the object.
 - (a) We will first test the evolution with the pose and shape parameters independently. For the pose parameter, we will start with an initial surface that has the same shape than the object to be extracted, but is scaled, rotated and translated. We will then initialize the surface such that at least some of its interior overlaps with the interior of the object in the labelmap. We will run the surface evolution equation updating the pose parameters and verify that it converges to the boundary of the object in the labelmap.
 - (b) For the shape parameter, we will start with a sphere as an initial surface. We will then initialize the surface such that at least some of its interior overlaps with the interior of the object in the labelmap. We will run the surface evolution equation updating the shape parameters and verify that it converges to the boundary of the object in the labelmap.
 - (c) We will then test the evolution with the combined pose and shape parameters. One issue we will have to be careful about is the determination of the Δs for each parameter. Since the magnitude of each parameter is different, we will have

to choose the Δ for each parameter such that one iteration produces a similar magnitude of change on the surface (for example a unit displacement).

2. We will then test the surface evolution with no shape prior on a the caudate dataset.

In this case, we will have to determine the optimal function f to use based on the type of imagery. We propose to start with a Gaussian model of the interior of the object to be segmented vs. its exterior. We will determine the mean and standard deviation of the voxels that lie inside and outside the object in the image and use those statistics to drive our curve evolution. We will again test the shape and pose parameters evolution separately and then combined

5.1.5 Proposed Surface Evolution Derivation: case *with* prior

We will then derive a surface evolution equation with the shape prior derived in chapter 4, using the notation of Section 2.4.4, and adapting the derivation in the appendix.

With the shape prior, we change our set of shape parameters to be the weights of the shape prior defined in chapter 4. Recall that the set of Γ coefficients is represented as:

$$\Gamma = \bar{\Gamma} + \sum_{b=1}^{B-1} \sum_{i=1}^{K_b} \alpha_{b,i} U_{b,i} \quad (68)$$

where $\bar{\Gamma}$ is the mean of the coefficients of the training set, B is the total number of bands discovered during the training phase, K_b is the number of eigenvectors for band b , U_i is the i^{th} eigenvector for band b (learned during the training phase) and $\alpha_{b,i}$ is the weight parameter.

The weight parameters $\alpha = [\alpha_{0,1}, \dots, \alpha_{B,K_B}]$ are now the shape parameters of our model, since we learned their mean and standard deviation during the training phase and can limit the weights for a new shape to lie within ± 2 standard deviation of their mean in order to ensure that the new shape is close to other shapes in the population.

Again, the basic underlying mathematical idea is to write down an energy that minimizes a particular function f inside a region R , enclosed inside the surface \vec{S} :

$$E(\vec{S}) = \int_R f(\mathbf{x}) d\mathbf{x} = \oint_{\vec{S}} \langle F, S_u \times S_v \rangle du dv \quad (69)$$

In the derivation, we will represent the surface $\vec{\mathcal{S}}$ as:

$$\vec{\mathcal{S}}(\mathbf{p}, \Gamma) = T[p] \begin{bmatrix} \bar{x}(u, v) + \bar{\Gamma}_x + \sum_{b=1}^{b=B} \sum_{i=1}^{i=K_b} \Phi(u, v)(\alpha_{b,i}^x U_{b,i}^x) \\ \bar{y}(u, v) + \bar{\Gamma}_y + \sum_{b=1}^{b=B} \sum_{i=1}^{i=K_b} \Phi(u, v)(\alpha_{b,i}^y U_{b,i}^y) \\ \bar{z}(u, v) + \bar{\Gamma}_z + \sum_{b=1}^{b=B} \sum_{i=1}^{i=K_b} \Phi(u, v)(\alpha_{b,i}^z U_{b,i}^z) \\ 1 \end{bmatrix} \quad (70)$$

The goal is then to find a gradient flow of $\vec{\mathcal{S}}$ that minimizes the following energy, *in terms of the pose \mathbf{p} and shape parameters α* .

We will derive $\frac{dE}{dp}$ and $\frac{dE}{d\alpha}$. We will then conduct a parameter optimization via gradient descent, in similar spirit to [50]. The update equations for the shape and pose parameters in the gradient descent approach are given by:

$$\begin{aligned} \alpha^{t+1} &= \alpha^t - \Delta_\alpha \frac{dE}{d\alpha} \\ \mathbf{p}^{t+1} &= \mathbf{p}^t - \Delta_\mathbf{p} \frac{dE}{dp} \end{aligned} \quad (71)$$

where Δ_α and $\Delta_\mathbf{p}$ are positive step size parameters, and α^{t+1} , \mathbf{p}^{t+1} denote the values of the parameters α and \mathbf{p} at the $t + 1^{th}$ iteration, respectively. The updated shape and pose parameters will then be used to implicitly determine the updated location of the segmenting surface.

Again no special numerics are required.

5.1.6 Proposed Experiments for surface evolution with shape prior

We propose the same set of experiments as in section 5.1.4, but using the full evolution equation with the prior. We will first use a synthetic dataset that consists of simple similar shapes (spheres with bumps on their surface that have a Gaussian distribution for their location and height). We will then use the caudate dataset, as well as one other dataset from the NAMIC repository (such as hippocampus structures). Since we have ground truth segmentations for all structures in the NAMIC repository, we will be able to calculate the mean squared error between the ground truth and the result of our segmentation technique.

We will then compare our technique to the Active Shape Model in 3D, using the same datasets, the same training and testing sets. We will compare both techniques with the

mean squared error between the ground truth and the segmented surface.

We propose to investigate how registration of the shapes in the population in the training set influences the accuracy of the shape prior learned during the training phase or our algorithm.

In chapter 4, we present two techniques to register shapes to the training set. Both techniques use a mapping to the sphere: the Brechbuhler et. al technique uses an area constraint, the Haker et. al. technique uses landmarks. Figure 24 summarizes the mapping and registration process for the Haker et. al technique. Once the coordinates of each shape are mapped to the sphere, the coordinates of the original vertices of the shape are interpolated so that they are defined for each vertex of a high resolution mesh on the sphere (last column of Figure 24). Using this new mesh for all shapes (that is all shapes now have the same triangulation and the same number of vertices), the shapes are then registered with Procrustes alignment so that the same vertex on the mesh corresponds to a similar point on each shape (where similarity is measured by euclidean distance among points), as shown in Figure 25. During this process, if two identical shapes are mapped to the sphere with a slight rotation in the u and v direction (by translating the landmarks with some Gaussian noise for example), the coordinates defined on the new spherical mesh will be slightly misaligned.

We plan to investigate the sensitivity of the segmentation results such misalignment during the registration process for both techniques. For the Haker et. al technique, we will translate the position of the landmarks picked on each surface both in the u and v direction (so that the landmarks are translated along the surface). The most misalignment happens when the landmarks are translated by $du/2$ and $dv/2$, where du is the average distance in the u direction between two vertices on the spherical mesh and dv is the average distance in the v direction. For the Brechbuhler technique, we will rotate the (θ, ϕ) parameterization on the sphere according to the same rule.

For both techniques, we will then compare the segmentation error for different levels of noise to the case without noise, for the caudate dataset. In particular, it would be interesting to find out whether there is a correlation between the magnitude of the segmentation error

and the amount of misalignment.

5.2 *Shape classification using spherical wavelets*

Once a segmentation is obtained, a desired task is to conduct an analysis of shape differences among two groups of segmented structures. This can be useful for longitudinal studies, as well as disease detection studies, where one group contains structures from patients with a disease and the other group contains structures from healthy patients. To test for shape differences, each shape needs to be described by a set of features. A simple example is to test for the difference in volume (a single feature) for brain ventricles of patients with schizophrenia vs. brain ventricles of healthy patients. One way to test for such a difference is to assume that the volume measurement for each shape is a random variable V and assume that V is normally distributed in each population. Then one can test statistically whether the distribution of the volume for each population are different using analysis of variance technique.

However often features in a population are not necessarily normally distributed. To address this issue, non-parametric permutation tests do not assume the form of the distribution in order to detect significant differences among the population. In permutation tests, the difference between groups is tested against distances provided by all permutations of samples from both groups. Using this technique, Styner et. al have shown that there is a difference of width in the caudate head for patients with schizotopal disorder v.s. healthy patients [46]. They used as features the coordinate of registered vertices surfaces of the shapes.

We are collaborating with Professor Martin Styner from University of North Carolina to include our spherical wavelet features into their non-parametric testing pipeline, provided as part of the NAMIC toolkit. We would like to first test whether the results found with spherical wavelet features correlate with the results found using vertex coordinates as features. Given the different nature of the wavelet feature set, we might also find some disparities in the results. Since the wavelet coefficients represent an integral over a local neighborhood of vertices (where the size of the neighborhood depends on the scale), one

possibility is that the wavelet coefficients at a large scale smooth out shape differences by integrating over many vertices. In this case, coefficients at a higher scale might be better indicators of group differences. However, if there are significant global differences among groups, it is also possible that coefficients at a coarse scale would be good indicators of group difference. We plan to use coarse scale coefficients only as features for our first experiment, then gradually add coefficients of higher scale to the feature set to test for differences at varying scales. We plan to collaborate with Professor Styner to understand and explain our results.

APPENDIX A

APPENDIX

A.1 Region-Based Flow derivation

In this section, we show the derivation for the region-based flow. The basic underlying mathematical idea is to write down an energy that minimizes a particular function f inside a region R , enclosed inside the curve $\vec{\mathcal{C}}$. The goal is then to find a gradient flow of $\vec{\mathcal{C}}$ that minimizes the following energy:

$$E(\vec{\mathcal{C}}) = \int_R f(\mathbf{x}) \, d\mathbf{x} = \oint_{\vec{\mathcal{C}}} \langle \vec{F}, \vec{\mathcal{N}} \rangle \, ds \quad (72)$$

where $\vec{\mathcal{N}}$ denotes the unit normal of $\vec{\mathcal{C}}$, ds is the Euclidean arclength element, and $\vec{F}(\mathbf{x})$ is a vector field chosen so that $\nabla \cdot \vec{F}(\mathbf{x}) = f(\mathbf{x})$. For example, given $\mathbf{x} = (x, y)$

$$\vec{F}(x, y) = \begin{pmatrix} F^x(x, y) \\ F^y(x, y) \end{pmatrix}$$

where:

$$\begin{aligned} F^x(x, y) &= \frac{1}{2} \int_0^x f(\lambda, y) \, d\lambda \\ F^y(x, y) &= \frac{1}{2} \int_0^y f(x, \lambda) \, d\lambda \end{aligned}$$

so that:

$$\nabla \cdot \vec{F}(x, y) = \frac{\partial F^x(x, y)}{\partial x} + \frac{\partial F^y(x, y)}{\partial y} = \frac{1}{2} f(x, y) + \frac{1}{2} f(x, y) = f(x, y) \quad (73)$$

The equivalence between the region integral based on f and the contour integral based on \vec{F} in Equation 72 follows from the divergence theorem.

To derive the gradient flow, we start by considering a fixed parameterization $p \in [0, 1]$ of the curve $\vec{\mathcal{C}}$ which does not vary as the curve evolves in time t so that (p, t) are independent variables. By a change of variable we may write E as follows:

$$E(\vec{\mathcal{C}}) = \int_0^1 \langle \vec{F}, J\vec{\mathcal{C}}_p \rangle \, dp \quad (74)$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ denotes a -90° rotation matrix. Differentiating with respect to t yields

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_t, J\vec{\mathcal{C}}_p \right\rangle + \left\langle \vec{F}, J\vec{\mathcal{C}}_{pt} \right\rangle \, dp \quad (75)$$

Where $\frac{d\vec{F}}{dx}$ denotes the Jacobian matrix of \vec{F} with respect to \mathbf{x} , that is if $\mathbf{x} = (x, y)$

and

$$\frac{d\vec{F}}{dx} = \begin{bmatrix} \frac{\partial F_x}{\partial dx} \frac{\partial F_x}{\partial dy} \\ \frac{\partial F_y}{\partial dx} \frac{\partial F_y}{\partial dy} \end{bmatrix} \quad (76)$$

If we integrate by parts the second term of Equation 75, we obtain

$$\int_0^1 \left\langle \vec{F}, J\vec{\mathcal{C}}_{pt} \right\rangle \, dp = \left\langle \vec{F}, J\vec{\mathcal{C}}_t \right\rangle |_0^1 - \int_0^1 \left\langle \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_p, J\vec{\mathcal{C}}_t \right\rangle \, dp = - \int_0^1 \left\langle \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_p, J\vec{\mathcal{C}}_t \right\rangle \, dp \quad (77)$$

where we are using the fact that for a parametrized closed curve, $\vec{\mathcal{C}}(0) = \vec{\mathcal{C}}(1)$, therefore $\vec{\mathcal{C}}_t(0) = \vec{\mathcal{C}}_t(1)$. Now Equation 75 becomes:

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_t, J\vec{\mathcal{C}}_p \right\rangle - \left\langle \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_p, J\vec{\mathcal{C}}_t \right\rangle \, dp \quad (78)$$

Reaaranging terms leads to:

$$\frac{dE}{dt} = \int_0^1 \left\langle \vec{\mathcal{C}}_t, \frac{d\vec{F}}{dx}^T J\vec{\mathcal{C}}_p \right\rangle - \left\langle J^T \frac{d\vec{F}}{dx} \vec{\mathcal{C}}_p, \vec{\mathcal{C}}_t \right\rangle \, dp \quad (79)$$

$$\frac{dE}{dt} = \int_0^1 \left\langle \vec{\mathcal{C}}_t, \left(\frac{d\vec{F}}{dx}^T J - J^T \frac{d\vec{F}}{dx} \right) \vec{\mathcal{C}}_p \right\rangle \, dp \quad (80)$$

Since the expression

$$\frac{d\vec{F}}{dx}^T J - J^T \frac{d\vec{F}}{dx} = \begin{bmatrix} 0 \frac{\partial F_x}{\partial dx} + \frac{\partial F_y}{\partial dy} \\ -(\frac{\partial F_x}{\partial dx} + \frac{\partial F_y}{\partial dy})0 \end{bmatrix} = \begin{bmatrix} 0 \nabla \cdot \vec{F} \\ -\nabla \cdot \vec{F} 0 \end{bmatrix} = fJ$$

, Equation 80 becomes:

$$\frac{dE}{dt} = \int_0^1 \left\langle \vec{\mathcal{C}}_t, f J \vec{\mathcal{C}}_p \right\rangle dp = \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, f \vec{\mathcal{N}} \right\rangle ds \quad (81)$$

. We see then that the form of the gradient flow for $\vec{\mathcal{C}}$ (the negative of the gradient so that the region integral decreases most rapidly) is revealed to be

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = -f \vec{\mathcal{N}} \quad (82)$$

Thus the flow depends only upon f , not upon our particular choice for F .

A.2 Soft Shape Prior Region-Based Flow derivation

In this section, we consider a more general class of region-based energy functionals where the integrand f depends upon another family of region integrals over R . From Chapter 3, Equation 40, we have the following energy:

$$E(\vec{\mathcal{C}}) = - \int_R \phi \, d\mathbf{x} + \int_{\vec{\mathcal{C}}} ds + \alpha \int_R \epsilon_1^p(\mathbf{x}) \, d\mathbf{x} \quad (83)$$

we focus on the third term, and therefore want to solve for the minimum of the following energy:

$$E(\vec{\mathcal{C}}) = \int_R f(\epsilon_1(\mathbf{x}, r, t)) \, d\mathbf{x} \quad (84)$$

where $f(x) = x^p$ and

$$\epsilon_1(\mathbf{x}, r, t) = \int_R \mathcal{X}(\mathbf{x}, r, \mathbf{y}) d\mathbf{y} \quad \text{where } \mathcal{X}(\mathbf{x}, r, \mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \in B(\mathbf{x}, r) \\ 0 & \text{if } \mathbf{y} \notin B(\mathbf{x}, r) \end{cases} \quad (85)$$

Note that we make $\epsilon_1^p(\mathbf{x}, t)$ depend on the artificial time parameter t since R (the interior of $\vec{\mathcal{C}}$) changes as the curve evolves over time.

We start out as in the previous section to rewrite the integral as a contour integral

$$E(\vec{\mathcal{C}}) = \int_R f(\epsilon_1(\mathbf{x}, r, t)) \, d\mathbf{x} = \oint_{\vec{\mathcal{C}}} \langle \vec{F}(\mathbf{x}, t), \vec{\mathcal{N}} \rangle \, ds \quad (86)$$

where $\vec{\mathcal{N}}$ denotes the unit normal of $\vec{\mathcal{C}}$, ds is the Euclidean arclength element, and $\vec{F}(\mathbf{x}, t)$ is a vector field chosen so that $\nabla_{\mathbf{x}} \cdot \vec{F}(\mathbf{x}) = f(\epsilon_1(\mathbf{x}, r, t))$ (note that $\nabla_{\mathbf{x}}$ means the divergence operator with respect to \mathbf{x} only).

As in the previous section A.1, to derive the gradient flow, we start by considering a fixed parameterization $p \in [0, 1]$ of the curve $\vec{\mathcal{C}}$ which does not vary as the curve evolves in time t so that (p, t) are independent variables. By a change of variable we may write E as follows:

$$E(\vec{\mathcal{C}}) = \int_0^1 \langle \vec{F}(\mathbf{x}, t), J \vec{\mathcal{C}}_p \rangle \, dp \quad (87)$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ denotes a -90° rotation matrix. Differentiating with respect to t yields

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{d\vec{F}}{d\mathbf{x}} \vec{\mathcal{C}}_t, J \vec{\mathcal{C}}_p \right\rangle + \left\langle \vec{F}, J \vec{\mathcal{C}}_{pt} \right\rangle + \left\langle \vec{F}_t, J \vec{\mathcal{C}}_p \right\rangle \, dp \quad (88)$$

Where $\frac{d\vec{F}}{d\mathbf{x}}$ denotes the Jacobian matrix of \vec{F} with respect to \mathbf{x} . As seen in the previous section A.1, the first two terms of Equation 75 simplify to $[\oint_{\vec{\mathcal{C}}} \langle \vec{\mathcal{C}}_t \cdot f(\epsilon_1) \vec{\mathcal{N}} \rangle \, ds]$ so Equation 88 becomes (with a change of variable for the third term):

$$\frac{dE}{dt} = \oint_{\vec{\mathcal{C}}} \langle \vec{\mathcal{C}}_t \cdot f(\epsilon_1) \vec{\mathcal{N}} \rangle + \langle \vec{F}_t, \vec{\mathcal{N}} \rangle \, ds \quad (89)$$

Using Green's theorem,

$$\oint_{\vec{\mathcal{C}}} \langle \vec{F}_t, \vec{\mathcal{N}} \rangle \, ds = \int_R \nabla_{\mathbf{x}} \cdot \vec{F}_t(\mathbf{x}) \, d\mathbf{x} = \int_R \frac{\partial f(\epsilon_1(\mathbf{x}, t))}{\partial t} \, d\mathbf{x} = \int_R f'(\epsilon_1(\mathbf{x}, t)) \epsilon_{1t}(\mathbf{x}, t) \, d\mathbf{x}$$

. So

$$\frac{dE}{dt} = \oint_{\vec{\mathcal{C}}} \langle \vec{\mathcal{C}}_t \cdot f(\epsilon_1) \vec{\mathcal{N}} \rangle \, ds + \int_R f'(\epsilon_1(\mathbf{x}, t)) \epsilon_{1t}(\mathbf{x}, t) \, d\mathbf{x} \quad (90)$$

The second term does not immediately reveal the form of the gradient for $\vec{\mathcal{C}}$ (recall that we need to have an inner product with $\langle \vec{\mathcal{C}}_t, . \rangle$). We further manipulate this term

by noticing that ϵ_1 itself has the form of an energy (region integral whose integrand does not depend on $\vec{\mathcal{C}}$) and so taking its derivative with respect to t is equivalent to (using the technique of Section A.1):

$$\epsilon_{1t} = \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) \vec{\mathcal{N}} \right\rangle ds \quad (91)$$

Plugging this in to Equation 90, we obtain:

$$\frac{dE}{dt} = \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, f(\epsilon_1) \vec{\mathcal{N}} \right\rangle ds + \int_R f'(\epsilon_1(\mathbf{x}, t)) \left[\oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) \vec{\mathcal{N}} \right\rangle ds \right] d\mathbf{x} \quad (92)$$

$$\frac{dE}{dt} = \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, f(\epsilon_1) \vec{\mathcal{N}} \right\rangle ds + \int_R \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, f'(\epsilon_1(\mathbf{x}, t)) \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) \vec{\mathcal{N}} \right\rangle ds d\mathbf{x} \quad (93)$$

$$\frac{dE}{dt} = \oint_{\vec{\mathcal{C}}} \left\langle \vec{\mathcal{C}}_t, \left[f(\epsilon_1) + \int_R f'(\epsilon_1(\mathbf{x}, t)) \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) d\mathbf{x} \right] \vec{\mathcal{N}} \right\rangle ds \quad (94)$$

Therefore the final gradient flow is:

$$\frac{\partial \vec{\mathcal{C}}}{\partial t} = - \left[f(\epsilon_1(\vec{\mathcal{C}})) + \int_R f'(\epsilon_1(\mathbf{x}, t)) \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) d\mathbf{x} \right] \vec{\mathcal{N}} = \epsilon_1^p(\vec{\mathcal{C}}, r) + p \int_R \epsilon_1^{p-1}(\vec{\mathcal{C}}) \mathcal{X}(\mathbf{x}, r, \vec{\mathcal{C}}) d\mathbf{x} \quad (95)$$

REFERENCES

- [1] ANGENENT, S., HAKER, S., TANNENBAUM, A., , and KIKINIS, R., “Laplace-beltrami operator and brain surface flattening,” *IEEE Trans. Medical Imaging*, vol. 18, pp. 700–711, 1999.
- [2] AUBERT, G., BARLAUD, M., FAUGERAS, O., and JEHAN-BESSON, S., “Image segmentation using active contours: Calculus of variations or shape gradients?,” *SIAM Journal on Applied Mathematics*, vol. 63, no. 6, 2003.
- [3] AYACHE, N., “Medical computer visioni, virtual reality and robotics,” *Image and Vision Computing*, vol. 13, no. 4, pp. 295–313, 1995.
- [4] AYACHE, N., CINQUIN, P., COHEN, I., COHEN, L. D., HERLIN, I. L., LEITNER, F., and MONGA, O., “Segmentation of complex 3D medical objects: a challenge and a requirement for computer assisted surgery planning and performing,” in *Computer Integrated Surgery*, MIT-Press, 1995.
- [5] BRECHBHLLER, C., GERIG, G., and KBLER, O., “Parametrization of closed surfaces for 3-d shape description,” *CVIU*, vol. 61, pp. 154–170, March 1995.
- [6] CASELLES, V., CATTE, F., COLL, T., and DIBOS, F., “A geometric model for active contours in image processing,” *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.
- [7] CASELLES, V., KIMMEL, R., and SAPIRO, G., “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, no. 11, pp. 61–79, 1997.
- [8] CHAN, T. F., SHEN, J. J., and VESE, L., “Variational pde models in image processing,” *Notice of AMS*, vol. 50, pp. 14–26, Jan 2003.
- [9] COHEN, L. D., “On active contour models and balloons,” *CVGIP: Image Understanding*, vol. 53, pp. 211–218, March 1991.
- [10] COHEN, L. and COHEN, I., “Finite element methods for active contour models and balloons for 2d and 3d images,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1131–1147, 1993.
- [11] COOTES, T. F. and TAYLOR, C. J., “Combining point distribution models with shape models based on finite element analysis,” *Image Vis. Computing*, vol. 13, no. 5, p. 4039, 1995.
- [12] COURANT, R. and HILBERT, D., *Methods of Mathematical Physics*. Interscience, 1953.
- [13] DAVATZIKOS, C., TAO, X., and SHEN, D., “Hierarchical active shape models, using the wavelet transform,” *IEEE Trans. Medical Imaging*, vol. 22, no. 3, pp. 414–423, 2003.
- [14] DELINGETTE, H. and MONTAGNAT, J., “Topology and shape constraints on parametric active contours.” Technical report, INRIA, France, 2000.

- [15] DESCHAMPS, T. and COHEN, L., “Fast extraction of minimal paths in 3d images and application to virtual endoscopy,” *Med. Image Anal.*, vol. 5, no. 4, 2001.
- [16] FARIN, G., *Curves and Surfaces for CAGD*. New York: Academic Press.
- [17] FLASQUE, N., DESVIGNES, M., CONSTANS, J., and REVENU, M., “Acquisition, segmentation and tracking of the cerebral vascular tree on 3d magnetic resonance angiography images,” *Med Image Anal.*, vol. 5, pp. 173–83, Sept 2001.
- [18] FRANGI, A., NIJSEN, W., VINCKEN, K., and VIERGEVER, M., “Multiscale vessel enhancement filtering,” in *MICCAI'98*, pp. 130–137, 1998.
- [19] GERIG, G., STYNER, M., JONES, D., WEINBERGER, D., and LIEBERMAN, J., “Shape analysis of brain ventricles using spharm,” *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA01)*, 2001.
- [20] GOLLAND, P., GRIMSON, W. E. L., SHENTON, M. E., and KIKINIS, R., “Detection and analysis of statistical differences in anatomical shape,” *Medical Image Analysis*, vol. 9, no. 1, pp. 69–86, 2005.
- [21] GRAYSON, M., “The heat equation shrinks embedded plane curves to round points,” *J. Differential Geometry*, vol. 26, pp. 285–314, 1987.
- [22] HAKER, S., WARFIELD, S., and TEMPANY, C., “Landmark-guided surface matching and volumetric warping for improved prostate biopsy targeting,” *MICCAI*, 2004.
- [23] HAN, X., XU, C., and PRINCE, J., “A topology preserving level set method for geometric deformable models,” *IEEE Transactions on PAMI*, vol. 25, pp. 755–768, June 2003.
- [24] KASS, M., WITKIN, A., and TERZOPOULOS, D., “Snakes: Active contour models,” *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1987.
- [25] KICHENASSAMY, S., KUMAR, A., OLVER, P., TANNENBAUM, A., and YEZGI, A., “Conformal curvature flows: from phase transitions to active contours,” *Archive for Rational Mechanics and Analysis*, vol. 134, pp. 275–301, 1996.
- [26] KRISSIAN, K., MALANDAIN, G., and AYACHE, N., “Model-based detection of tubular structures in 3d images,” *Computer Vision and Image Understanding*, vol. 80, no. 2, pp. 130–171, 2000.
- [27] LEPETIT, V. and FUA, P., *Foundations and Trends in Computer Graphics and Vision*, vol. 1, ch. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey, pp. 1–89. Now Publishing, 2005.
- [28] LEVENTON, M., GRIMSON, E., and FAUGERAS., O., “Statistical shape influence in geodesic active contours,” in *Comp. Vision and Patt. Recon. (CVPR)*, June 2000.
- [29] LORIGO, L. M., FAUGERAS, O., GRIMSON, W. E. L., KERIVEN, R., KIKINIS, R., NABAVI, A., and WESTIN, C.-F., “Codimension-two geodesic active contours,” in *Comp. Vision and Patt. Recon. (CVPR)*, June 2000.
- [30] MALLAT, S., *Wavelet Tour of Signal Processing*. Academic Press, 1999.

- [31] MCINERNEY, T. and TERZOPOULOS, D., “A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis,” *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, pp. 69–83, 1995.
- [32] MCINERNEY, T. and TERZOPOULOS, D., “Deformable models in medical image analysis: a survey,” *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.
- [33] MCINERNEY, T. and TERZOPOULOS, D., “T-snakes:topology adaptive snakes,” *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, 2000.
- [34] OSHER, S. and FEDKIW, R., *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2002.
- [35] PAPOULIS, A., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.
- [36] PICHON, E., *Novel Methods for Multidimensional Image Segmentation*. PhD thesis, Georgia Institute of Technology, 2005.
- [37] PRESS, W., TEUKOLSKY, S., VETTERLING, W., and FLANNERY, B., *Numerical Recipes in C++*. Cambridge University Press.
- [38] R. MALLADI, J. S. and VEMURI, B., “Shape modeling with front propagation: a level set approach,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, 1995.
- [39] SATO, Y., S.NAKAJIMA, SHIRAGA, N., ATSUMI, H., YOSHIDA, S., KOLLER, T., GERIG, G., and KIKINIS., R., “Three dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images,” *Med. Imag Anal*, vol. 2, pp. 143–168, 1998.
- [40] SCHRÖDER, P. and SWELDENS, W., “Spherical wavelets: Texture processing,” in *Rendering Techniques '95* (HANRAHAN, P. and PURGATHOFER, W., eds.), Springer Verlag, 1995.
- [41] SCHRÖDER, P. and SWELDENS, W., “Spherical wavelets: Efficiently representing functions on the sphere,” *Computer Graphics Proceedings (SIGGRAPH 95)*, pp. 161–172, 1995.
- [42] SETHIAN, J., *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [43] SHI, J. and MALIK, J., “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [44] STAIB, L. and DUNCAN, J., “Boundary finding with parametrically deformable models,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, no. 11, pp. 1061–1075, 1992.
- [45] STAIB, L. and DUNCAN, J., “Deformable fourier models for surface finding in 3d images,” in *Proc. 2nd. Conf. on Vis. in Biomed. Computing* (ROBB, R., ed.), vol. 1808, pp. 90–104, SPIE, 1992.

- [46] S.VETSA, STYNER, M., PIZER, S., LIEBERMAN, J., and GERIG, G., “Caudate shape discrimination in schizophrenia using template-free non-parametric tests,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI*, pp. 661–669, 2003.
- [47] T. DESCHAMPS, L. C., “Fast extraction of tubular and tree 3d surfaces with front propagation methods,” *International Conference on Pattern Recognition*, 2002.
- [48] TAYLOR, P., “Invited review: computer aids for decision-making in diagnostic radiology - a literature review,” *British Journal of Radiology*, vol. 68, pp. 945–957, 1995.
- [49] TER HAAR ROMENY, B. M., ed., *Computational imaging and vision*.
- [50] TSAI, A., YEZZI, A. J., III, W. M. W., TEMPANY, C., TUCKER, D., FAN, A., GRIMSON, W. E. L., and WILLSKY, A. S., “A shape-based approach to the segmentation of medical imagery using level sets,” *IEEE Trans. Med. Imaging*, vol. 22, no. 2, pp. 137–154, 2003.
- [51] VASILEVSKIY, A. and SIDDIQI, K., “Flux-maximizing geometric flows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1565–1578, 2002.
- [52] WANG, K., DUTTON, R., and TAYLOR, C., “Improving geometric model construction for blood flow modeling,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 18, no. 6, pp. 58–69, 1999.
- [53] WEBSITE, N., “National alliance for medical image computing.” <http://www.namic.org>.
- [54] WIKIPEDIA, “Wavelet.” <http://en.wikipedia.org/wiki/Wavelet>.
- [55] XU, C., PHAM, D. L., and PRINCE, J. L., *andbook of Medical Imaging, Volume 2*, ch. Medical Image Segmentation Using Deformable Models, pp. 129–174. SPIE Press, May 2000.
- [56] YANG, Y., TANNENBAUM, A., and GIDDENS, D. P., “Knowledge-based 3d segmentation and reconstruction of coronary arteries using ct images,” in *IEEE EMBS*, pp. 1664–1666, 2004.
- [57] YEZZI, A. J., TSAI, A., and WILLSKY, A., “A fully global approach to image segmentation via coupled curve evolution equations,” *Journal of Visual Communication and Image Representation*, vol. 13, pp. 195–216, 2002.
- [58] ZHU, S. C. and YUILLE, A., “Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 884–900, September 1996.