Extracting Branching Object Geometry via Cores

by
Yonatan Fridman

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2004

Approved by:

_____
Advisor: Stephen Pizer

_____
Reader: Stephen Aylward

_____
Reader: James Damon

_____
Reader: Guido Gerig

_____
Reader: Jack Snoeyink

ABSTRACT

Yonatan Fridman

Extracting Branching Object Geometry via Cores

(Under the direction of Stephen M. Pizer)

The extraction of branching objects and their geometry from 3D medical images is the goal of the research described here. The methods I develop and analyze are based on optimization of medial strength embodied in cores, a type of multiscale medial axis. The cores research described here focuses on object branches.

The dissertation describes two sets of methods, one that computes 1D cores of tubular objects and another that computes 2D cores of non-tubular, or slab-like, objects. Each of these sets of methods uses a core-following technique that optimizes a medialness measure over position, radius, and orientation. I combine methods for branch-finding, branch-reseeding, and core-termination with core-following to produce a tool that can extract complex branching objects from 3D images without the need for user interaction.

I demonstrate the performance of the tubular core methods by extracting blood vessel trees from 3D head MR data, and I demonstrate the performance of the slab-like core methods by extracting kidneys from 3D abdominal CT data. I also integrate the two sets of methods with each other to extract kidneys and adjoining vasculature from CT data. Finally, I perform a thorough analysis of the effects of object geometry on cores using synthetic images. This analysis shows impressive resistance of the tubular core methods to image noise. It also shows that the slab-like core methods are not as robust but are still effective for extracting branching objects from medical images with relatively low noise.

ACKNOWLEDGEMENTS

I am tremendously grateful for the time and effort that my advisor, Steve Pizer, has invested in my education over the past six years. Not only have his technical knowledge and insight been invaluable to my academic growth, but his sincerity, integrity, and selflessness have been a wonderful influence on my personal growth.

My committee members, Stephen Aylward, Jim Damon, Guido Gerig, and Jack Snoeyink, have helped shape my research and lead me in worthwhile directions. I thank them for their interest in my research and education. I also extend my gratitude to Delphi Bull, without whose support I would have been lost attempting to finish my dissertation remotely.

I have been financially supported by the Office of Naval Research and the National Institutes of Health. I thank both Russell Taylor and Bob Goldstein for their time and expertise and for the opportunity to expand my academic horizons.

I thank my parents and my sister for urging me forward, for taking interest in my pursuits, and for always believing in me. I wouldn't be where I am now had my father not introduced me to programming so many years ago.

Mostly, I owe so much to my wife, Adelia. She often helped me advance my research with her mathematical intuition and valuable suggestions, she kept me moving forward with her unending love and support, and she was unbelievably patient the whole time.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1. Introduction

1.1 Motivation

Three-dimensional medical imaging provides a wealth of information not available to physicians until recently. Imaging devices and methods are advancing rapidly, improving the quality of images being produced and increasing the variety of organs and diseases being imaged. The diversity of imaging modalities being developed is broadening the field of medical imaging, with the dramatic improvement of magnetic resonance (MR) technology over the last decade, as well as the innovation of modalities such as functional magnetic resonance imaging (fMRI) that capture physiological function (as opposed to structure).

While physicians can obtain valuable knowledge by directly viewing such images, the medical image analysis community strives to advance the medical field through the computerized processing of digital images. The automated extraction of accurate structural data from digital images and the subsequent analysis of the data form the basis of medical image analysis. Automated extraction and analysis of data can facilitate diagnosis, therapy delivery, and therapy planning. An accurate segmentation of a tumor, for example, can aid in radiation treatment by helping to target the tumor while exposing surrounding tissue to minimal radiation. A segmentation of the cerebral vasculature can be used to create a 3D visual representation with which a physician can more easily guide a catheter toward a stenosis.

The segmentation of medical images is a key step in their analysis, as it provides quantifiable information on the anatomic structures contained within the images. Furthermore, other steps of medical image analysis, such as registration and motion tracking, often rely on prior segmentations [Duncan00]. Medical images are frequently segmented manually, but this process poses problems, particularly for three-dimensional images, on

which it is performed one slice at a time without making use of the volumetric information available. This process can be labor intensive and time consuming, and the resulting segmentations are generally not reproducible because they contain human bias [McInerney96]. Specifically, Metz showed that human observers performed inconsistently when viewing medical images under varying conditions [Metz85, Metz86]. Thus, the need exists to develop semi-automatic (interactive) and fully automatic segmentation methods.

In particular, this dissertation explores a method for segmenting branching structures in medical images. The ability to segment branching structures is crucial, notably in applications involving blood vessels. A 3D representation of cerebral vasculature that contains explicit branching and connection information and that is computed with little user interaction is of great value in both surgical planning and endovascular procedures [Bullitt01]. In the former, tumor treatment relies on the suspension of blood flow to the tumor while still allowing blood to reach healthy tissue. Thus, when cutting off blood flow to a given artery, a physician must identify child arteries that will subsequently be deprived of blood. In the latter, a 3D visual representation of a vessel tree can greatly facilitate the guiding of a catheter to a desired location, as the connections between vessels can be ambiguous given only the original image. The segmentation of vasculature in other parts of the body, including the lungs, the liver, and even the eyes, is also of significant interest in the medical community.

Other branching structures for which a segmentation is useful include bronchi and nerves, which are structurally similar to blood vessels, and bony objects such as vertebrae and the pelvis, which are qualitatively different in shape. Even combinations of anatomic objects can be segmented as single branching structures, for example, a kidney with its adjoining renal artery, renal vein, and/or ureter, or the liver along with its adjoining vasculature.

Although branching structures are prevalent in the human body, few segmentation methods are capable of handling them effectively and even fewer locate branches explicitly. The following paragraphs provide a literature survey that overviews medical image

segmentation methods, along with their strengths and weaknesses, and then focuses on existing methods for handling branching structures in medical images.

Boundary-finding deformable models are a widely researched class of methods for the automatic segmentation of medical images (see [McInerney96] for a thorough discussion of the development of deformable model methods through the mid 1990's). This class of methods has its roots in the "snakes" method originally developed in 2D by Kass *et al.* [Kass88] and then extended to 3D [Terzopoulos88, Cohen92]. Snakes, or active contour models, begin with a roughly initialized boundary curve or surface which then deforms according to a combination of image and shape constraints. Although active contour methods are limited by the topology of the initial boundary, they are less susceptible to image noise or boundary discontinuities than are more local methods [Kass88].

Other researchers built on these ideas by using models with more global descriptions. Staib and Duncan utilize Fourier parameterizations and incorporate prior shape information to further improve robustness [Staib96]. Cootes *et al.* use point distribution models and training images to describe the expected variability of a class of objects and then constrain model deformation according to the training set. Thus, they use statistical data on expected object shape and image appearance to extend active contour models to what they call active shape models [Cootes95] and active appearance models [Cootes98]. In [Cootes01] they compare the two. Székely *et al.* combine attributes of snakes, Fourier parameterizations, and models of shape variability to robustly segment 3D medical images using spherical harmonics as shape descriptors [Székely96, Kelemen99]. These methods suffer from the need for user interaction or user training, a trade-off between robustness and flexibility, and the need for a prior model, which imposes a fixed topology.

A related category of segmentation methods includes those based on level sets, which allow for implicit changes in topology. Methods from this class were originally developed by Osher and Sethian in the computer vision community [Osher88], first applied to 2D medical images by Malladi *et al.* [Malladi95], and extended to 3D by Caselles *et al.* [Caselles97]. These methods evolve a wave front according to a partial differential equation, stopping at image features such as edges. Lorigo *et al.* modified this level set approach to

segment tubular structures in 3D images by forcing the wave front to be 1D with co-dimension 2 [Lorigo99]. Though these unconstrained methods allow for variable topology, they are not meant to handle the level of noise or the types of object breaks and inconsistencies that deformable model methods are meant to handle.

Methods based on classification of tissue types are unrelated but historically significant. The finite Gaussian mixture model [Wells96] and the Gaussian hidden Markov model [Shattuck01, Zhang01] are two classification-based methods that are used primarily to segment images of the brain into white matter, gray matter, and cerebrospinal fluid. Since these methods classify voxels and do not produce shape information, I only allude to them.

Some of the segmentation methods described can handle only single-figure objects and not branching objects. Others can handle slight protrusions or branches but have difficulty with elongated figures such as blood vessels. Certain curve/surface evolution methods can handle arbitrarily complex branching structures, but they provide no explicit branching or connection information. This last class includes methods such as [Lorigo99], discussed earlier, and [Vasilevskiy02], which uses a geometric flow technique wherein a surface evolves under image-based constraints.

Published evaluation data [Bullitt01, Aylward02] together with my own experience suggest to me that, in dealing specifically with blood vessels or other tubular objects, the most effective method to date for producing representations that explicitly handle bifurcations is that of Aylward and Bullitt [Aylward96, Bullitt01]. Their method segments relevant tubular objects using a multiscale image intensity ridge traversal method. It then uses a post-processing step to link the segmented tubes in a hierarchical manner, to produce information on branch locations and parent-child connectivity, and to recognize and discard tubes that do not belong to the complete structure. This method has shown excellent results on a wide variety of organs and imaging modalities; it nonetheless requires multiple seed points in order to segment each branch individually and then a post-processing step to merge the branches into one structure. The need remains for a method capable of segmenting a branching tubular structure and identifying all of its bifurcations in a single pass.

The problem of segmenting general (non-tubular) objects in 3D medical images and, in the process, extracting branching information is an even more difficult one. Methods based on Voronoi skeletons [Ogniewicz92, Näf96] and shocks of evolving surfaces [Siddiqi99, Bouix00, Siddiqi02] (both to be discussed further in section 2.3) compute a medial description of branching structures that contains explicit information on bifurcations, but these methods require a boundary segmentation as input. Though they can be combined with an appropriate boundary-finding segmentation method to give the desired results, such a two-step process is connected through heavy propagation of error – minor differences in outputs from the segmentation step can lead to considerable discrepancies in the resulting medial descriptions. Another method, in which medial representations (m-reps) are deformed to match image data, is being developed to simultaneously segment and extract branching geometry from anatomic objects [Pizer03a]. M-reps are capable of effectively handling multi-figure objects but require a prior model of the object to be segmented, necessitating significant training and prior knowledge of the approximate shape of the object. These model-based methods are of little use when dealing with structures with a highly variable branching structure such as blood vessel trees.

Medial methods have come to the forefront of efforts to analyze branching objects in 3D medical images. Medial descriptions of biological objects have their roots in the Medial Axis Transform, introduced by Blum several decades ago [Blum67], which describes an object in terms of a locus of position-radius pairs making up the medial axis, or backbone, of the object. The main drawback of the Medial Axis Transform is its sensitivity to shape perturbations. However, medial descriptions benefit from their ability to change in biologically natural ways such as bending and widening. Medial descriptions also benefit from the structural information on which they are based. The natural association between opposing walls of a blood vessel, for example, is not inherent in the more commonly used boundary descriptions but is evident when an object is represented medially. Burbeck *et al.* have shown that this association has close ties to the way in which humans perceive objects [Burbeck96]. From an algorithmic point of view, this association also provides a constraint on possible object shapes, potentially making medial methods resistant to image noise.

The Medical Image Display and Analysis Group at the University of North Carolina at Chapel Hill has attempted to exploit the benefits of medial shape descriptions while avoiding the Medial Axis Transform's over-sensitivity to shape perturbations. One result is a medial axis at scale, *i.e.*, a medial axis of an object as it appears in a blurred image, known as a "core" [Morse94, Furst99]. Not only are cores insensitive to image noise and small perturbations [Morse98], but they are computed directly from image grayscale information, allowing them to be used as an automated segmentation tool. Cores carry with them radius and orientation information of the object(s) they represent, providing additional geometric information [Pizer98]. Cores were originally developed for use with 2D images as manifolds of dimension one and co-dimension one, but they have since been extended to 3D images in the form of both 1D and 2D manifolds (having co-dimension 2 and 1, respectively).

This dissertation further develops these cores to make use of the desirable attributes of medial descriptions while eliminating the shortcomings of medial methods that rely on either a prior model or a boundary segmentation. To be of general applicability, the core method must be able to handle a wide range of anatomic objects, including branching structures. A branching structure is defined in this dissertation as a structure whose medial description consists of more than one medial manifold. In other words, we desire a method capable of producing multiple medial manifolds, along with associated connectivity information, for describing branching structures.

It seems desirable to take advantage of the qualitative difference in shape between, say, a finger and the palm of a hand. A finger is an example of a tubular object whereas a hand is an example of a slab-like object. A tubular object, in three-dimensional Euclidean space, is one whose medial manifold is one-dimensional, *i.e.*, a space curve. Thought about another way, it is an object with circular cross-sections. A slab-like object is one whose medial manifold is two-dimensional. In this dissertation I relax the term "tubular" and use it to describe objects such as blood vessels whose cross-sections can be elliptical or otherwise distorted.

Given these challenges, the goal of this dissertation is to present a robust medial method for extracting geometric information from anatomic objects in 3D medical images.

The method is designed to extract relevant structures directly from such images without the need for *a priori* knowledge of the structures or images. It is intended to handle complex (albeit loop-free) branching structures, both tubular and slab-like.

1.2 Problem

The method presented in this dissertation aims to accurately extract complex branching objects from 3D medical images. While the analysis of medical images and the recovery of anatomic objects from the images is a well developed area of research, the automatic extraction of branching structures in particular is still a difficult problem, exacerbated by image noise. Although imaging devices are improving daily, images still contain nonnegligible levels of noise due to the acquisition process, image sampling, and motion artifacts. Noise can be reduced during image acquisition by increasing exposure time, but in images involving radiation this has the negative effects of exposing the patient to additional harmful radiation and of increasing the chances for motion artifacts in the image. Furthermore, certain tissues display little contrast when imaged, hampering the task of automatic object extraction since the effect of noise is relative to the difference in signal intensities between the object and the adjacent background. For example, in abdominal CT the kidney and liver appear only slightly brighter than their surroundings, resulting in a low signal-to-noise ratio. The fact that these two organs are displayed at almost identical intensities causes them to appear as a single object when pressed side by side.

This dissertation attempts to deal with these challenges to extract the branching geometry of anatomic objects, and to extract the objects themselves, via cores. The cores presented in this document are computed by a manually initialized predictor-corrector marching algorithm. A "medial atom" is fit to the image data using a multi-dimensional optimization process in which spokes radiating from the center of the medial atom – two spokes when extracting slab-like objects, a larger set of concentric spokes when dealing with tubular objects – attempt to situate their tips normal to and located at edges within the image. The agreement of the medial atom with the image data is maximized by varying the atom's location, radius, and orientation. Once the medial atom has been fit in this manner, the algorithm takes a step forward by predicting a new core location and then correcting this

location using the described optimization process. This process computes a variant of what is called an optimum parameter core.

One problem that arises in this marching algorithm is that cores of branching objects do not intersect at a common branch point. Consequently, when a bifurcation in the object is encountered, the core is terminated, and two new cores are automatically initialized at the bases of the two new branches. Object bifurcations are detected using an affine-invariant corner detector, false positives produced by the corner detector are rejected based on a set of heuristics, and the two new cores are initialized using geometric information from the existing core. The image in the area local to each core is constantly probed using statistical measures to decide when to terminate core extraction due to a lack of image information.

1.3 Thesis Statement

Optimum parameter cores with branch-handling and end-detection provide an automated means for computing the branching geometry of tubular structures in 3D medical images and for computing the branching geometry of general structures in relatively low noise 3D medical images.

1.4 Contributions of This Work

The five main contributions of this dissertation are introduced here. They are presented in an intuitive order such that each builds on ones already described. The primary impact of this dissertation is in the area of branching, particularly contributions 2 and 4.

CONTRIBUTION 1. *A robust implementation of 1D cores in 3D*

I describe and implement a method for computing one-dimensional cores of tubular objects in three-dimensional images. The method is robust and resistant to high levels of image noise as shown in subsections 7.2.1 and 7.3.1.1 by experiments on both clinical and synthetic images. Furst's implementation of 1D cores in 3D [Furst99] has difficulty following cores through areas of weak image information or high background noise and is limited by a predefined sampling rate. Aylward's method [Aylward02] deals well with noise

but uses separate filters to determine the location and radius of tubular objects and defines their orientation implicitly. In this dissertation I introduce an implementation of a 1D optimum parameter core in 3D that uses a single filter to optimize a medial atom's location and parameters. I also give descriptions of the characteristics that make my implementation work where its predecessors failed.

CONTRIBUTION 2. *A system for handling branches using 1D cores in 3D*

I develop a system for handling branching tubular objects using cores, which mathematically do not branch. The system combines a set of image and shape metrics, some of which have been previously developed by others, in a way that effectively locates object branches. Contributions of this system include

1. the use of an affine-invariant corner detector for locating object branches;
2. a method based on core geometry for predicting the locations and attributes of the two child branches once a branching region has been located; and
3. a means for segmenting tubular tree structures without any post-processing.

CONTRIBUTION 3. *A method for detecting ends of cores*

I develop a statistical method for terminating the core-following algorithm that handles both explicit (capped) object ends and implicit (tapered) object ends, such as blood vessels. The method is robust enough to correctly determine when image information no longer warrants following the core of an object yet can still traverse small breaks in that object.

CONTRIBUTION 4. *An implementation of 2D cores with branch-handling in 3D*

I modify the 1D core-following, branch-handling, and end-detection methods to compute 2D cores in 3D. This extension allows the segmentation of general objects, not only tubular objects, in volume images. For example, I consider the kidney as a non-tubular object and segment it using 2D cores. Furthermore, I integrate 1D and 2D cores to segment anatomic structures such as the kidney and adjoining renal artery. This is the first image-

based medial method that is capable of automatically segmenting general branching objects in 3D, as shown in Chapter 7.

CONTRIBUTION 5. *An analysis of the effects of object geometry on the performance of cores*

I vary certain geometric properties of branching objects and analyze how these variations affect the performance of image analysis via cores. Specifically, I examine how resistant to image noise core extraction is over a range of object widths, object curvatures, and branching angles in section 7.3.

1.5 Overview of This Document

The remainder of this dissertation addresses the following topics.

Chapter 2 presents background material on medial methods in medical image analysis. It introduces fundamental mathematical concepts that are later adapted for my purposes.

Chapter 3 describes the core method for segmenting tubular objects as it is implemented in this dissertation.

Chapter 4 presents changes and additions that enable cores to be used for the segmentation of slab-like objects.

Chapter 5 discusses properties of cores at object branches and presents techniques for handling branching objects using cores. It also conveys the ability of cores, as described in this dissertation, to handle structures that combine slabs and tubes, for example, a kidney and adjoining renal artery.

Chapter 6 presents a method for deciding when to terminate the core-following algorithm due to a lack of image information.

Chapter 7 demonstrates results of the core methods applied to clinical images and presents an analysis of the effects of object geometry on the performance of cores.

Chapter 8 gives a discussion and review of the work presented and suggests possibilities for future work.

Chapter 2. Background

2.1 Introduction

The work presented in this dissertation focuses on the extraction of branching objects from medical images using cores, which are medial axes at scale. This chapter presents an overview of the history of medial methods for image analysis, from the Blum Medial Axis Transform to current multiscale medial methods. After presenting medial methods for analyzing general anatomic objects, it discusses approaches, both medial and non-medial, for analyzing tubular objects, such as blood vessels, in medical images. It then provides a detailed background that presents mathematical properties of cores, different types of cores, and extensions to cores for use in analyzing both tubular and slab-like objects. The chapter concludes with a discussion of behavioral aspects of medial methods at object branches and ends.

2.2 Blum Medial Axis

Blum defined the Medial Axis Transform in early work meant to describe object shape [Blum67, Blum78]. Blum proposed computing a 2D object's medial axis as the locus of centers of maximal disks contained within that object. To elaborate, consider a disk that fits entirely within the boundary of an object. If that disk is not contained in any other disks that also fit within the object's boundary then it is called a maximal disk. Such a maximal disk must be tangential to the object's boundary in at least two locations (fig. 2.1). These locations of tangency are known as involutes and are associated with the medial point located at the center of the maximal disk. Blum referred to the set of centers of an object's maximal disks as that object's medial axis. Furthermore, he called the collection of these centers along with the radii of the associated maximal disks the Medial Axis Transform (MAT). The MAT is a transform of the original object in the sense that it describes the object completely – in the limit as the number of disks defined by the MAT approaches infinity, the union of

the interiors of all disks is exactly the interior of the object. Thus the MAT provides both geometric information, such as object width, and a complete and compact representation of the object it describes.

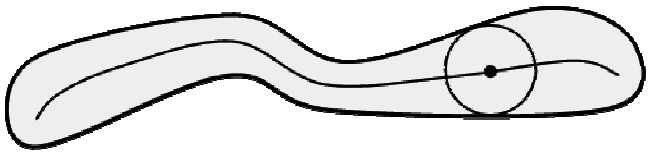

Fig. 2.1. The medial axis of a simple 2D object along with an inscribed maximal disk centered on the medial axis and tangent to the object boundary at two involute locations.

Some properties of the Blum medial axis in 2D are of particular relevance to the work presented here. At places where an object has either a significant protrusion or a bifurcation, a corresponding singular medial location exists having three medial axis neighbors, *i.e.*, three medial curves intersect at the specified medial location (fig. 2.2). The associated radii of these three incoming medial curves converge to a shared radius at the intersection point. A maximal disk centered at the intersection point generically has three involutes; the maximal disk can have more than three involutes in non-generic cases where more than three medial curves share an intersection point [Yomdin81, Mather83]. At places where an object boundary achieves a local maximum in curvature, a corresponding singular medial location has a single medial axis neighbor. The maximal disk associated with the singular medial location osculates the object's boundary, meaning that the radius of curvature of the boundary at the location of tangency is equal to the radius of the maximal disk (fig. 2.2). The two involutes necessarily converge at this location of tangency [Leyton87].

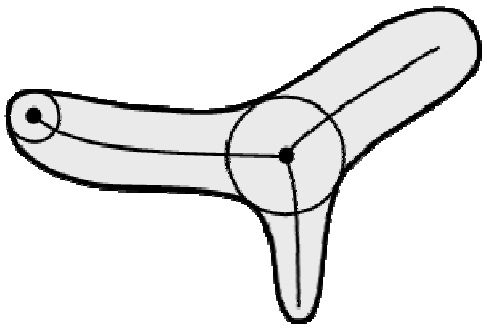

Fig. 2.2. The medial axis of a branching 2D object. One maximal disk is shown centered at the intersection of three medial curves and having three points of tangency with the object boundary. The other maximal disk is centered at the end of a medial curve and has a single point of double tangency (*i.e.*, osculation).

This 2D MAT can be extended to 3D in an intuitive manner, with the key distinction being that maximal disks are replaced by maximal spheres. These maximal spheres, like the maximal disks in 2D, are generically bi-tangent to an object boundary, but the centers of all maximal spheres generate medial manifolds that are two-dimensional rather than the one-manifolds occurring in 2D images. Thus, the medial axis of a generic object in 3D is a set of intersecting two-manifolds with associated radius information.

The intersections of these manifolds are no longer individual points, as they were in 2D, but rather space curves. Moreover, an explicit asymmetry is generally created that was not present in 2D. In the medial axis of a 2D object there exists symmetry in that three medial curves intersecting at a point are all fundamentally the same – it is not clear which curve is the parent and which two are the children. Katz & Pizer [Katz03] argue that each of the three branches is somewhat like a parent and somewhat like a child, and they define the relationships between the branches mathematically. In the generic 3D case, on the other hand, only two medial manifolds intersect. In most cases, one of these two manifolds is unmistakably the parent and the other is the child (fig. 2.3), and their intersection is a curve that makes up a portion of the boundary of the child and that acts as a crease in the parent [Yushkevich02]. The tangent planes to the medial manifolds are discontinuous along this crease, unlike elsewhere on the manifolds. Some uncommon situations do arise, however, in which symmetry is formed between medial manifolds. This is the case, for example, when

two figures are joined in such a way that each figure intersects the other's crest, where a crest location is defined as a local maximum of surface curvature. When this happens, one figure can be arbitrarily chosen as the parent of the other. Another case of symmetry occurs when more than two medial manifolds intersect at a single location. These symmetric situations will be discussed further in section 2.6.

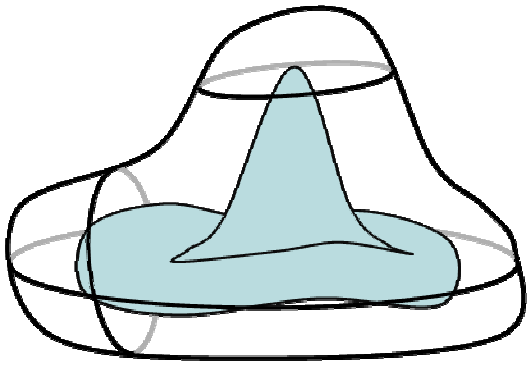

Fig. 2.3. A common form for the medial axis of a branching 3D object.

Like the intersections of medial manifolds, ends of medial manifolds in 3D are space curves. These end curves have properties similar to those described for medial end points in 2D. In particular, the maximal spheres centered at end curves osculate an object surface at crest locations [Bruce96].

Although two-manifolds form the generic medial axes of 3D objects, a medial axis in 3D can degenerate to a one-manifold or even to a point. A medial axis degenerates to a one-manifold in the singular case where an object is tubular, *i.e.*, has circular cross-sections. This situation results in a full circle of points on the object's surface that are each other's involutes and a conical volume created by the line segments connecting the involutes and the corresponding medial axis location. A medial axis degenerates to a point when dealing with a sphere, in which case the center of the sphere is the sole medial point and all points on the sphere's boundary are each other's involutes. Medial representations of tubular objects will be discussed in detail later; representations of spheres will not be considered further.

Additional properties and analysis of the 3D medial axis can be found in [Yomdin81, Nackman82, Bruce96, Giblin00].

2.2.1 Smoothed Local Symmetries

A variant of Blum's medial axis, Smooth Local Symmetries (SLS), was first studied in 2D by Brady and Asada [Brady84]. Unlike the 2D version of Blum's medial axis, which is the locus of centers of maximal disks contained within an object, the SLS is the locus of midpoints of maximal disk chords whose endpoints are the involutes of the maximal disks (fig. 2.4).

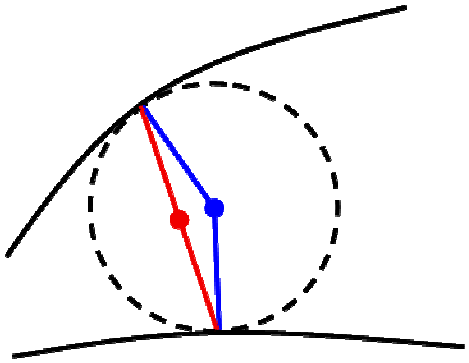

Fig. 2.4. Opposing boundaries of a 2D object (*solid black*) along with a maximal disk (*dotted black*). The location and "spokes" of the MAT corresponding to the maximal disk are shown in blue. The location and "spokes" of the SLS corresponding to the same maximal disk are shown in red.

In 2D, the SLS has similar properties to Blum's medial axis at interior locations, *i.e.*, locations not corresponding to object branches or object ends. In the vicinity of object branches, however, two separate SLS curves generically intersect. This differs from the generic behavior of Blum's medial axis at branching locations, where three curves converge to a branch point. The behavior of these two types of axes also differs near object ends. While Blum's medial axis ends inside objects at centers of curvature of the objects' boundaries, the SLS continues past these locations and ends at object boundaries.

In 3D, the SLS generalizes to the locus of midpoints of maximal sphere chords whose endpoints are the pair of involutes of the maximal spheres. When an object in 3D is tubular, each SLS axis location is at the center of a disk defined by a circle of involutes. This differs from the conical region formed by the line segments connecting each tubular Blum medial location and its involutes. While the SLS has been less completely studied than Blum's medial axis, certain singularity theoretic studies have suggested that Blum's axis is more appropriate than the SLS for general objects in 3D, but that the disks generated by the SLS are more appropriate than the Blum cones in the tubular case [Damon04]. The behavior of the 3D SLS is discussed further in subsection 5.3.1.

2.3 Other Medial Methods

Blum's MAT has many desirable properties as a shape descriptor but suffers from its sensitivity to object boundary perturbations. Minute perturbations in an object's boundary lead to slight shifting of the resulting medial axis, as expected, but they often produce a new branch of the medial axis as well. This is an undesirable property of shape descriptors in many real world applications where it is detrimental for the topology of the medial axis to change as a result of boundary or image noise. In an attempt to overcome this drawback of the MAT, a variety of medial methods have combined the concept of Blum's medial axis with a branch of image analysis known as scale space analysis. The theory of scale space, first developed by [Iijima62] and later re-developed by [Marr80], [Witkin83], and [Koenderink84], among others, is based on the notion that different parts of a scene become apparent when viewed at different scales, or resolutions. For example, when viewing a tree from nearby you can see individual leaves, but when the same tree is viewed from farther away or with a different area of focus you see the tree as a whole [Koenderink90]. Similarly, when viewed at the appropriate scale, desired large scale features of an object are apparent but noise and other small features are not. This notion, when applied to medial methods, results in multiscale medial axes that have reduced sensitivity to image noise and boundary detail as compared with Blum's medial axis.

An example of a multiscale medial method, presented in [Näf96], utilizes the concept of the Voronoi diagram [Ogniewicz90, Ogniewicz92]. Given a sampled object boundary, the

Voronoi diagram of that discrete boundary is a partitioning of the ambient space such that each region of the partitioning contains exactly one boundary sample point and every location within a region is closer to its contained sample point than to any other. To relate this to Blum's medial axis, a circle of appropriate radius centered along a Voronoi edge is tangential to two of the object's boundary sample points (or more if the circle is centered on an intersection of Voronoi edges). The resulting Voronoi edges can be partitioned into two disjoint sets. The first set contains edges traced out by centers of circles that are tangent to medial involutes on opposing sides of the object boundary; these edges approximate portions of the corresponding medial axis. The other set contains edges traced out by centers of circles that are tangent to adjacent boundary samples; these edges are a result of the discretization of the object boundary and are therefore discarded. As the distance between an object's boundary samples approaches zero, the set of edges of the Voronoi diagram, excluding those edges that separate adjacent samples, converges to the medial axis of the object [Schmitt89]. Once a Voronoi diagram is produced from the object's sampled boundary, diagram edges that correspond to small details are pruned and a topology is imposed to create a coherent medial axis. To produce medial axes at progressively finer levels of scale, edges that correspond to successively smaller scale details are pruned.

In another multiscale medial method, curve/surface evolution is performed using a Hamilton-Jacobi equation [Siddiqi99, Bouix00, Siddiqi02]. Given an object boundary, the Euclidean distance map of the boundary is computed. The boundary then evolves inward in the direction of the negative gradient vector field of the distance map. Medial locations are given as those places where shocks, or collisions, of the propagating curve occur. A location is identified as a shock if the average outward flux per area or volume, depending on image dimensionality, of the gradient vector field over a small region surrounding that location is negative and large in magnitude. This technique works because the average outward flux per area/volume over any non-medial region in the limit as the region shrinks to a point is zero, but at a medial location this quantity approaches a value that is negative and proportional in magnitude to the dot product of the normal to the medial manifold and the gradient vector field [Siddiqi02]. Since this method alone is not enough to guarantee a thin and topologically correct medial axis, a thinning process is applied whereby points are removed in reverse

order of their flux values, starting from the most positive flux value, while taking measures not to discard desired medial end points whose flux is below a chosen threshold. The choice of this threshold affects the scale of the resulting medial axis, where higher thresholds produce axes at finer scale. A nice comparison of these two methods with cores is given in [Pizer03b].

2.4 Tube Segmentation Methods

The medial methods described in the previous section are capable of extracting valuable geometric information from anatomic objects in medical images but rely on a prior segmentation. The computation of segmentations is a major challenge in the field of medical image analysis, due in part to the variability of both anatomic objects and the attributes of the medical images themselves. Methods that aim to segment structures with particular geometric properties have an advantage over general segmentation methods in that they can exploit the known structure geometry. In Chapter 3 of this dissertation I describe such a method designed specifically for the segmentation of tubular objects, which are of particular interest in the medical community due to their prevalence in the human body in the form of blood vessels, nerves, bronchi, and even some bones.

Here I present existing methods, both medially and non-medially based, that have been developed for the quantification and analysis of tubular objects in 3D medical images using the known geometry. These methods fall into five categories, which are sketched in the next paragraph and then detailed in subsections 2.4.1 – 2.4.5.

Intensity thresholding methods use expectation maximization, adaptive region growing, or other statistically-based schemes to select which voxels are part of the object of interest and which are background. They do not normally compute connectivity information between voxels and as a result produce no structural information. *Vessel enhancement filtering*, a particular form of anisotropic filtering that aims to enhance blood vessels and other tubular objects while suppressing irrelevant image information, also produces no structural information. *Level set evolution methods* are based on an explicit surface that evolves according to geometric and image-based forces in an attempt to conform to

structures in the image. *Deformable model fitting methods* for tubular object segmentation generally find approximate medial axes of the tubes being sought and then locate the tubes' boundaries by fitting models based on the medial axes to the image data. Finally, *core methods* march along a height ridge of image intensity or other such function that responds strongly at middles of tubes, computing radius information along the way.

2.4.1 Intensity Thresholding

Intensity thresholding methods decide which voxels of an image belong to a given object, region, or class based on statistical measures. Since voxels are normally classified but not explicitly structured by intensity thresholding methods, geometric information is not readily available.

One type of intensity thresholding method uses region growing, in which a seed point is placed within the region of interest and then expands to encompass new voxels based on homogeneity criteria and other information often defined by the user. Pohle and Toennies [Pohle01] present a region growing scheme that learns its homogeneity criteria from characteristics of the region being grown, decreasing the need for user input.

Another type of intensity thresholding method classifies tissue using a Gaussian mixture model defined on image intensities [Wilson97, Wilson99]. The classification of tissue is achieved using an iterative expectation maximization algorithm in which the most likely Gaussian parameters are estimated. Once desired tissues are segmented using the determined classification, the segmentation can optionally be refined using structural criteria.

2.4.2 Vessel Enhancement Filtering

Vessel enhancement filtering is a form of anisotropic filtering in which narrow tubular objects in particular are enhanced. In anisotropic filtering, an effort is made to reduce noise using a blurring process while minimizing the blurring of edges and other relevant image structure. In other words, these schemes aim to smooth images within homogeneous regions but not smooth them across regions. As with intensity thresholding, anisotropic filtering does not produce explicit geometric information. Many vessel enhancement filtering methods have been developed using a variety of filters.

Chen and Hale [Chen95], Du and Parker [Du95], and Orkisz *et al.* [Orkisz97] independently developed vessel enhancement filtering to improve MRA data as a pre-processing step to maximum intensity projection. Chen and Hale use directed low-pass filters. Du and Parker use a nonlinear adaptation of a Laplacian of a Gaussian filter to enhance small tubular structures. Orkisz *et al.* use a two-step method that determines local image structure orientation information and then performs nonlinear smoothing based on the determined orientations.

Koller *et al.* [Koller95] and Poli and Valli [Poli96] employ a variant in which a nonlinear combination of linear filters is used to search for tubular objects across scales. Lorenz *et al.* [Lorenz97] and Sato *et al.* [Sato98] perform vessel enhancement filtering using the eigenvalues of the Hessian matrix.

2.4.3 Level Set Evolution

Level set methods use a manifold to explicitly represent an object's surface. They are traditionally based on a manifold of co-dimension unity that is initialized within an object. The manifold is then attracted toward relevant image features, evolving according to a partial differential equation that imposes curvature constraints. Level set methods have the advantage that they explicitly provide structural information, yet they are not constrained by a predefined topology.

Lorigo *et al.* [Lorigo99, Lorigo01] were the first to successfully segment tubular structures in volume images using a level set method of co-dimension 2. Their implementation begins with a rough initial surface created either by thresholding an image or by generating a predefined surface. This initial surface need not be a good approximation. Given this initialization, the surface is evolved using a combination of the mean curvature flow, which influences the surface toward decreased curvature, and image forces that attract the surface; specifically, correspondence between the image gradient and the surface normals is favored. Finally, radii are estimated using surface principal curvature information.

One weakness of level set methods is that the surfaces they produce can leak from the desired object boundaries in places of weak image information. Although level set methods

have the ability to handle branching tubular structures, branching information is not generated explicitly.

Since Lorigo, other researchers have advanced the segmentation of tubular structures using level set evolution. Westin *et al.* [Westin00] improve Lorigo's method by using more sophisticated image properties, described in more detail in section 3.2, to attract the evolving surface. Deschamps and Cohen [Deschamps01, Deschamps02] augment existing methods with stopping criteria and techniques for freezing desired portions of the evolving manifold in an effort to prevent leaking. Quek and Kirbas [Quek01] somewhat modify level sets to create a method in which a wave propagates through a volume image, clinging to image locations that appear as tubular objects.

Vasilevskiy and Siddiqi [Vasilevskiy02] have developed an impressive geometric flow method in which a surface evolves to maximize inward flux. The method begins with a set of seeds located within the desired objects and grows those seeds based on the image gradient vector field. The direction of growth is the direction in which the inward flux of the vector field increases most rapidly. A current shortcoming of this method is that it is unclear when to terminate flow. In addition, as with level set methods, branches are not located explicitly.

2.4.4 Medial Axis-Based Deformable Model Fitting

Deformable model fitting methods are often more robust than the other types of methods due to prior knowledge built into a model. When designed specifically for tubular objects, deformable model fitting methods normally employ a model that's based on the medial axis of the desired tube, accompanied by local radius information. These tubular deformable model fitting methods can handle tubular structures with abnormalities, as several researchers have shown. Van Bemmel *et al.* [van Bemmel02] use a method based on medial axes to handle stenosed arteries, which appear as pinched tubes. De Bruijne [de Bruijne03] uses a method based on active shape models (see section 1.1) to analyze abdominal aortic aneurysms, which appear as bulged tubes.

A particularly successful example of a tubular deformable model fitting method that can handle both regular and deformed tubular objects, proposed by Frangi *et al.* [Frangi98, Frangi99], computes an approximate medial axis that is coupled to an object's boundary and then refined. In their method, an approximate medial axis, modeled using a B-spline representation, is initialized using a semi-automated approach based on an iso-surface rendering of the vasculature. This medial axis is then deformed to the image data in a snake-like manner using local Hessian information. The vessel wall is initialized as a swept surface [Piegl96] coupled to the medial axis and then deformed using both image data and prior knowledge of the image acquisition process. This method is capable of effectively locating tubular objects but is hampered by its need for user interaction during the medial axis initialization step and its inability to handle bifurcations. In general, deformable model fitting methods have the shortcoming that the topology of the object being segmented is fixed at the time the model is created.

2.4.5 Core Methods

The final category includes the method presented in this dissertation. Core methods are based on medial axes of objects, but they are not constrained to a fixed topology by a predefined model. Rather, core methods extract a medial axis, at scale (*i.e.*, in a blurred image), by following a height ridge either of intensity or of medialness. Medialness is a scalar function that measures how much an image location is like the middle of an object.

Perhaps the most complete method for automatically segmenting branching tubular structures in 3D images is the core method of Aylward and Bullitt [Aylward02], which uses multiscale image intensity ridge traversal with branch-handling. Their work is most closely related to the work presented in this dissertation, as both have grown from research carried out at the University of North Carolina [Morse93, Fritsch95, Aylward96, Eberly96, Pizer98]. In their intensity ridge traversal method they search for the position of a tube's medial axis using a Hessian operator applied to image intensities, then determine the tube's width information using a tubular kernel, and finally define orientation implicitly according to the Hessian's eigenvectors. In contrast, the approach of this paper is to simultaneously determine position, width, and orientation information as a ridge of medialness. Their

implementation is impressive in its combination of speed of computation and resistance to image noise.

The work I present suffers compared to the work of Aylward and Bullitt in terms of running time but benefits from its ability to extract a tubular tree structure starting from a single seed point. Their method combines individual branches as a post-processing step. The method presented in this dissertation is further extended to handle general (non-tubular) branching structures, as discussed in Chapter 4.

2.5 Cores

Cores were introduced in the previous section as a method for segmenting tubular structures in 3D images. These cores, which are 1D manifolds, have been extended to 2D cores capable of segmenting non-tubular structures [Furst99, Damon03a, Fridman04]. Since 1D and 2D cores in 3D images are the focus of this dissertation, a detailed background of cores is presented in this section.

2.5.1 Medial Atoms

A core, whether 1D or 2D, is composed of discrete samples called medial atoms that are fit to image data. A medial atom is a structure having a central hub and two equal length spokes originating from that hub (fig. 2.5). The hub is located medially within an object, defining a location on the core, and the spokes extend outward to touch opposing object boundary positions.

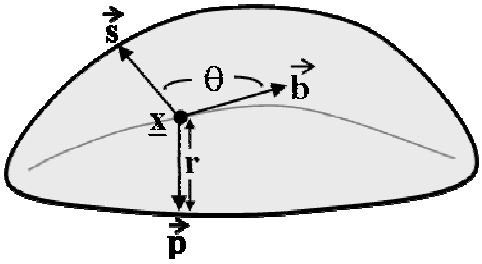

Fig. 2.5. The geometry of a medial atom shown with a cross-section of a 3D object. $\underline{x}$ is the medial atom's location; r is its radius; θ is its object angle; $\vec{b}$ is the spoke bisector, one of three orthogonal vectors that define $\underline{\vec{F}}$; and $\vec{p}$ and $\vec{s}$ are the atom's spokes.

Mathematically, a medial atom $\underline{m}$ in 3D is a structure defined by four parameters $(\underline{x}, r, \underline{\vec{F}}, \theta)$ having eight degrees of freedom (fig. 2.5). The vector $\underline{x}$ is a set of coordinates in 3-space that defines the location of the medial atom hub and corresponds to the center of Blum's maximal sphere. The scalar r is the length of the atom spokes at the location $\underline{x}$ and corresponds to the radius of the maximal sphere. Beyond these two parameters, medial atoms extend the paradigm of the maximal sphere to include orientation information.

The orientation information is included via a frame $\underline{\vec{F}}$ derived from two spokes, $\vec{p}$ and $\vec{s}$, that extend from the medial location $\underline{x}$ to the two involute locations (see section 2.2). Definition of a unique frame $\underline{\vec{F}}$ requires three scalars. I define two of those scalars to be the azimuth and altitude of $\vec{b}$, the bisector of the spokes $\vec{p}$ and $\vec{s}$ that lies in the subspace spanned by $\vec{p}$ and $\vec{s}$, in spherical coordinates from the positive x-axis. I define the third scalar to be a spin angle φ that is zero if $\vec{b}$, $\vec{p}$, and $\vec{s}$ have the same azimuth and that increases as the spokes rotate counter-clockwise about $\vec{b}$. To prevent $\vec{p}$ and $\vec{s}$ from flipping, I require $\vec{p} \times \vec{b}$ to point in the positive y-axis direction before $\underline{\vec{F}}$ is rotated by the azimuth, altitude, and spin angle. These choices are explained in section 4.3. Frame $\underline{\vec{F}}$ has generally been represented by a set of three orthogonal vectors, $\vec{b}$, $\vec{n}$, and $\vec{b}^{\perp}$, where $\vec{n}$ is perpendicular to $\vec{b}$ in the subspace spanned by $\vec{p}$ and $\vec{s}$, and where $\vec{b}^{\perp} = \vec{n} \times \vec{b}$ [Pizer03b].

Assuming the surface of the object is locally smooth, $\vec{n}$ is normal to the core tangent space at the location $\underline{x}$, so $\vec{b}$ and $\vec{b}^{\perp}$ span the core tangent space. The definition of $\vec{F}$ used in this dissertation facilitates the medial atom optimization described in sections 3.4 and 4.3.

The final medial atom parameter, θ, is the angle between $\vec{b}$ and each of the spokes in the subspace spanned by the spokes. Angle θ is known as the object angle since it describes the local rate of widening or narrowing of the object, but it is defined somewhat differently in this dissertation than it is elsewhere. A medial atom has no inherent forward/backward orientation, so most of the work led by Pizer [Pizer03b, among others] defines the forward direction as the direction of narrowing of the object, thereby imposing the constraint $0 \le \theta \le \pi/2$. In this document the forward direction of a medial atom is defined by the marching algorithms described in sections 3.4 and 4.3 and can be either the direction of narrowing or widening of the object. Thus, the constraint imposed in this dissertation is $0 \le \theta \le \pi$. When θ equals $\pi/2$, the resulting medial atom is circularly symmetric about the two collinear spokes. In this situation, the image information at the involute locations defines $\vec{n}$, but $\vec{b}$ is no longer well defined in the subspace spanned by $\vec{b}$ and $\vec{b}^{\perp}$. This has consequences for the computation of cores and will be discussed further in section 4.3.

Cores composed of generic medial atoms have co-dimension 1, so a core in a three-dimensional space is generally a two-dimensional manifold. But as an object approaches tubularity this medial manifold narrows. The extraction of cores of nearly tubular objects can be numerically problematic because minor narrowing or widening of the object can cause instability in core computation at singular locations where the object is perfectly tubular, *i.e.*, where its cross-section is circular. If the object is known to be nearly tubular, the core can be constrained to be one-dimensional (fig. 2.6). The medial atoms used to extract 1D cores from 3D images are defined by the same set of four parameters described above but they differ from general atoms in that they have a set of more than two concentric spokes. Each of these spokes still has its tail located at $\underline{x}$, is of length r, and forms an angle θ with the spoke bisector $\vec{b}$. Such 1D medial atoms are defined in subsection 2.5.4.

Fig. 2.6. A 1D core of a tubular object (*left*) and a 2D core of a slab-like object (*right*) in 3D.

2.5.2 Medialness

As mentioned in subsection 2.5.1, a core is composed of medial atoms fit to image data. The core is computed by optimizing the medialness $M(\underline{m})$ of each atom $\underline{m}$, where medialness is a scalar function that measures the fit of the medial atom to image data. Medialness is measured by a weighted sum over the image according to a chosen medialness kernel. Medialness kernels are built on the structure of a medial atom, *i.e.*, based on the parameters $(\underline{x}, r, \underline{\vec{F}}, \theta)$. An effective medialness kernel is designed to respond to an image most strongly when located at an object middle; for a bright object on a darker background this is generally accomplished using a kernel whose maximum intensity values are located near the center of the kernel. In addition, an effective medialness kernel should respond to image information that indicates opposing object boundaries without being overly sensitive to image noise or to intensity variations between the opposing boundaries.

In the early stages of research on cores in 2D images, Fritsch [Fritsch93] used the negative Laplacian of a Gaussian (LoG) (left side of fig. 2.7) as a medialness kernel. Its properties are defined by the parameters of a 2D medial atom in the following manner. The center of the medialness kernel, which corresponds to the maximum value of the negative LoG, is defined as the location $\underline{x}$ of a corresponding medial atom. The standard deviation of the Gaussian from which the kernel is computed is defined as the radius r of the corresponding atom. Since the LoG is isotropic, there is no use for the medial atom orientation parameters. This so called "Mexican hat" function has the desired bright center and dark surround structure mentioned in the previous paragraph. Furthermore, a Gaussian derivative seems the appropriate choice due to its properties as an edge detector at scale.

When applied as image filters, Gaussian derivatives blur the image while performing edge detection, without introducing any extraneous image structure in the process [Marr80]. While this kernel is resistant to image noise, it suffers from its sensitivity to image intensity variations interior to the object due to the strong weighting at the kernel's center. In an attempt to place more emphasis on the region near an object's boundaries and less on the object's interior Morse *et al.* [Morse93] used a kernel defined as a derivative of a Gaussian taken in a radial direction, integrated over a circle of radius r centered at $\underline{x}$ (right side of fig. 2.7). Morse defined the Gaussian to have a standard deviation of ρr where ρ is a multiplicative factor he often set at 0.25, as chosen through experimentation and user studies. As can be seen in fig. 2.7, this kernel places emphasis on object boundaries while responding minimally interior to the object. Eberly [Eberly96] showed that for $\rho = 1$ this kernel is equivalent to the LoG and as ρ approaches 0 the resulting core approaches Blum's medial axis.

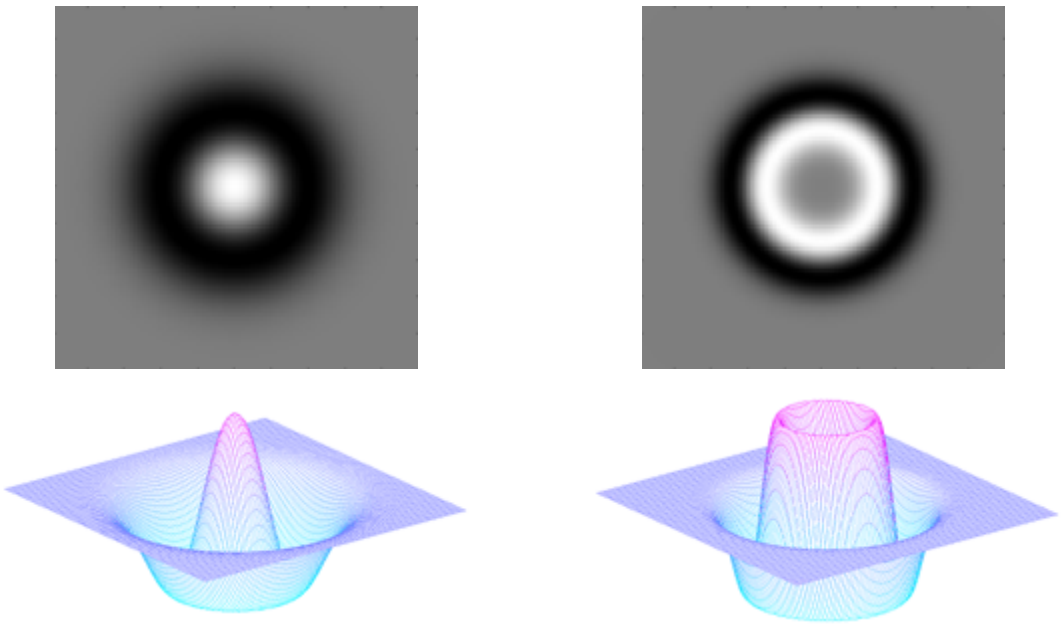

Fig. 2.7. Isotropic 2D medialness kernels of Fritsch and Morse.

Fritsch's kernel and Morse's kernel are both isotropic, but further cores research showed that more consistent results could be achieved with anisotropic medialness kernels

that probe only at two opposing boundary locations rather than over a full circle. Kernels that probe at opposing boundary locations emphasize object boundary information, whereas isotropic kernels are susceptible to noise at other locations. To this end, Fritsch [Fritsch93] created an oriented version of his Laplacian kernel by computing the second derivative in a single direction (left side of fig. 2.8). As with his isotropic kernel, $\underline{x}$ and r define the location and standard deviation, respectively, of the function. This oriented kernel makes use of the medial atom orientation by taking the derivative orthogonal to the vector $\vec{b}$. Puff [Puff95] then combined the advantages of Fritsch's oriented Laplacian kernel and Morse's boundary-based kernel by integrating over two opposing regions, each of which is weighted by a directional derivative of a 2D Gaussian (right side of fig. 2.8). In this case, a derivative of a Gaussian is placed at each of the two medial atom spoke tips, where the derivative is taken in the direction of the spokes. Since the object angle θ is still undefined, the spokes are assumed to be collinear and orthogonal to $\vec{b}$.

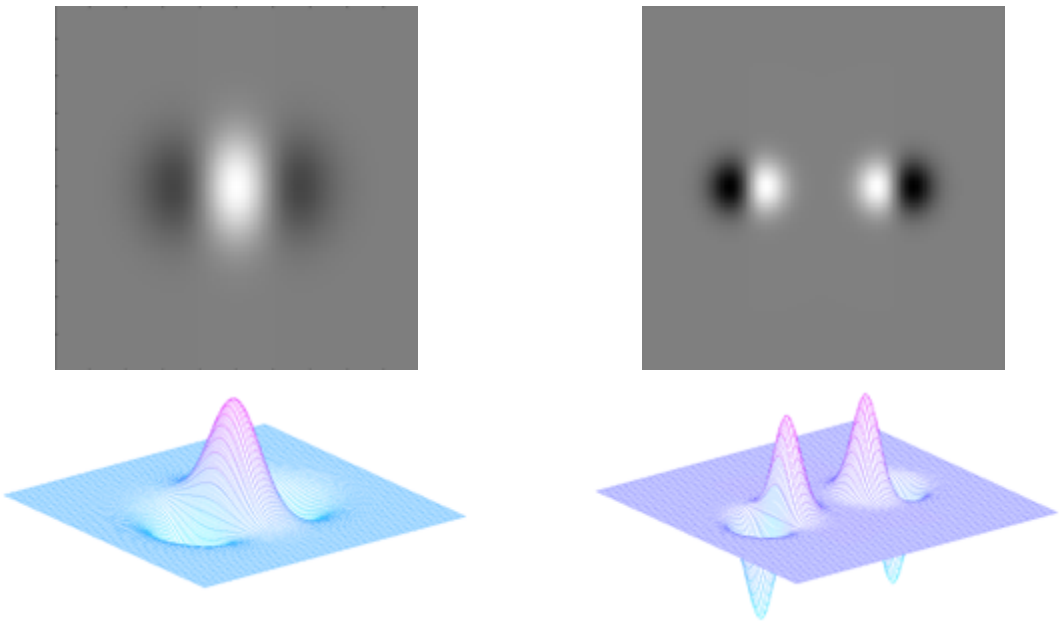

Fig. 2.8. Oriented 2D medialness kernels of Fritsch and Puff.

Furst [Furst99] further augmented Puff's medialness kernel by introducing the object angle parameter θ, thereby removing the restriction that the two Gaussian derivatives must

be collinear and allowing them to drift in a constrained manner. When the two spokes are collinear, *i.e.*, θ equals $\pi/2$, this is identical to Puff's kernel.

Furst extended his kernel from 2D to 3D by replacing each directional derivative of a 2D Gaussian with a directional derivative of a 3D (volumetric) Gaussian. He placed these two Gaussian derivatives at the tips of the spokes of a 3D medial atom, described in subsection 2.5.1. This medialness kernel is then applied as a weighting to the image of interest (fig. 2.9) and the result is integrated to give the desired medialness value. In both 2D and 3D this is given by the equation

$$M(\underline{m}) = D_{\hat{\vec{p}}} I(\underline{x} + \vec{p}, \sigma) + D_{\hat{\vec{s}}} I(\underline{x} + \vec{s}, \sigma) \tag{2.1}$$

where $D_{\hat{\vec{v}}} I(\underline{x}, \sigma)$ is the derivative of the image taken in the direction of a vector $\vec{v}$, at a location $\underline{x}$, and at a scale of interrogation σ. The choice of scale used in this document is discussed further in section 6.2. Such a 3D Furst kernel is used in this dissertation to compute medialness. The kernel is modified for the purposes of this dissertation by constraining the object angle to $\pi/2$ (resulting in a 3D version of Puff's kernel) and/or increasing the number of spokes, as described in section 3.3. More details of the aforementioned 2D kernels and their extensions to 3D can be found in [Pizer98] and [Furst99].

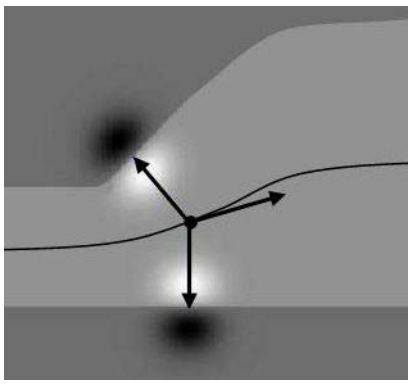

Fig. 2.9. A slice through Furst's 3D medialness kernel overlaid on a corresponding slice through a 3D image. Also shown are the medial atom's spokes and spoke bisector, as well as the object's core.

2.5.3 Medial Atom Optimization

Given a medialness kernel that responds most strongly when placed at image locations that resemble the middle of an object, we desire to automatically determine the location and parameters of the corresponding medial atom that maximize the kernel's medialness response. Medial atoms are thus fit to image data by computing an optimization of the corresponding kernel's medialness response over both Euclidean space and parameter space, resulting in atoms located along a height ridge of medialness, or a core. In a 3D image a core can be either 1D or 2D, as described in subsection 2.5.1, with a co-dimension of 2 or 1, respectively. For a 1D core there are 2 Euclidean directions of optimization, *i.e.*, linearly independent directions transverse to the core, while for a 2D core there is a single Euclidean direction of optimization. In either case, computation of a medial atom also requires optimization over the 5D parameter space $(r, \vec{F}, \theta)$. To generalize this notion, when computing a d-dimensional core within an n-dimensional space, the medialness of each medial atom must be optimized over $n - d$ linearly independent directions transverse to the core. In this dissertation d can be either 1 or 2 and n is 8, resulting in a 6- or 7-dimensional optimization space. We must decide how to choose the 6 or 7 linearly independent directions spanning this optimization space.

2.5.3.1 Optimum Parameter Cores

Two different possibilities for choosing these optimization directions have been studied. One is the optimum parameter method, in which the two subspaces, parameter space and Euclidean space, are treated separately. In parameter space, the directions of optimization are chosen as the parameters of the medial atom being optimized. While r and θ are obvious choices as directions of optimization, we must choose three optimization directions to cover the three degrees of freedom provided by $\vec{F}$. In this dissertation I choose those directions of optimization as the altitude of the spoke bisector $\vec{b}$, the azimuth of $\vec{b}$, and the spin angle of the medial atom about $\vec{b}$ as defined by the atom's spokes. The choice of these three optimization directions provides geometrically desirable results, as discussed in section 3.4. In Euclidean space, early research at the University of North Carolina chose the optimization directions as the eigenvectors of the Hessian matrix of the image corresponding

to its most negative eigenvalues (see subsection 2.5.4). With the development of oriented medialness kernels, it has since become more natural to choose the optimization directions as those directions orthogonal to the core's tangent space. This is straightforward when computing a core whose Euclidean co-dimension is 1. For a core with co-dimension 2, such as a 1D core in a 3D image, it is easy to choose a plane of optimization, but it is not obvious which two directions should be chosen to span that plane. As described further in section 3.4, the choice of these two directions is arbitrary as long as they are chosen in some consistent manner over the extent of a core.

Cores computed using this choice of optimization space, in which parameters are used explicitly as optimization directions, are an example of what are called optimum parameter cores [Fritsch95, Furst98]. They seem to represent anatomic objects in a more natural manner than maximum convexity cores, discussed next.

2.5.3.2 Maximum Convexity Cores

Earlier core research focused on maximum convexity cores [Morse94, Eberly96]. The directions of optimization of these cores are chosen as the $n-d$ directions of sharpest negative curvature of medialness, *i.e.*, the most medially convex directions. These are defined as the directions of greatest second derivative of the chosen medialness function, given by the eigenvectors of the Hessian matrix that correspond to its $n-d$ most negative eigenvalues [Eberly96]. The remaining d eigenvectors define the core's tangent space.

Maximum convexity cores differ from optimum parameter cores in that no explicit differentiation is made between Euclidean space and parameter space during optimization. This difference makes maximum convexity cores mathematically simpler than optimum parameter cores. Furthermore, their singularity-theoretic properties have been researched extensively in [Miller98, Keller99] and their mathematical properties are much better understood and defined than are those of optimum parameter cores. Still, recent research has shied away from maximum convexity cores and rather focused on optimum parameter cores for three main reasons outlined by Furst and Pizer [Furst98]:

1. The separation of parameters from Euclidean directions allows for the use of the parameters in modeling natural shape attributes such as widening/narrowing and bending.

2. The same separation of parameters allows for easier handling of spaces of higher dimensions. Maximum convexity cores have never been extended successfully beyond a 4D domain, limiting their applicability.

3. The incommensurability of position, radius, and orientation can lead to mathematically undesirable results when computing maximum convexity cores. Since optimum parameter cores treat these as separate directions of optimization, such difficulties are avoided.

The separation of parameters in optimum parameter cores also allows for the $(n-d)$-dimensional optimization space to be dealt with using multiple optimizations of lower dimension. This technique of breaking down the optimization space, used in [Aylward02] and subsequently in this dissertation (section 3.4), significantly reduces computation costs.

2.5.3.2.1 Connectors

Maximum convexity cores are computed as height ridges of medialness, meaning that every point $\underline{x}$ on the core is a maximum of medialness in the core's co-space. In other words, given a d-dimensional core residing in an n-dimensional co-space, for any choice of $n-d$ linearly independent directions $\vec{v}_i$, for $1 \le i \le n-d$, transverse to the core at $\underline{x}$, point $\underline{x}$ is at a relative maximum of medialness in each direction $\vec{v}_i$ [Furst98].

Damon generalized the height ridge structure by describing a larger relative critical set structure [Damon98, Damon99] which includes height ridges (or cores, when the height ridge is applied to a medialness function), height valleys, and height saddles. Damon's results hold for a variety of functions, but I restrict the discussion to medialness functions only. A location $\underline{x}$ is defined as a height valley if it is a minimum of medialness in the

core's co-space, *i.e.*, if there is a relative minimum of medialness at $\underline{x}$ in each direction $\vec{v}_i$. Height ridges and height valleys can be considered complementary in that a location $\underline{x}$ is a height ridge of a medialness function M if and only if it is a height valley of $-M$. A location $\underline{x}$ is defined as a height saddle if it is a relative maximum of medialness in some directions $\vec{v}_i$ and a relative minimum of medialness in the other directions. Damon showed that at locations where height ridges and height valleys of medialness end, height saddles of medialness continue, and thus he called these height saddles of medialness "connectors".

Damon and others [Damon98, Miller98, Keller99] have defined a variety of properties of maximum convexity ridges, saddles, and valleys. The relative maximum property discussed two paragraphs ago requires that at a location $\underline{x}$ on the core there exist $n-d$ negative eigenvalues of the Hessian matrix (see beginning of subsection 2.5.3.2). There are singular locations at the ends of maximum convexity cores where one of the following two conditions occurs: 1) the negative eigenvalue with the smallest magnitude becomes equal to zero, corresponding to a singular point of the Hessian matrix; 2) two eigenvalues become equal to each other, corresponding to a partial umbilic point of the Hessian matrix. In both of these cases the maximum convexity core ends and a connector begins. Similarly, Fritsch [Fritsch97] observed locations where optimum parameter cores end and connectors begin.

In this dissertation cores are computed using a modified optimum parameter technique that determines the location of a medial atom before determining its parameters (see section 3.4). These modified optimum parameter cores continue indefinitely, never transitioning to connectors due to the order of optimization specified in section 3.4. Using this order of optimization, a local relative maximum can be found in all directions transverse to the relative critical set starting from any location within an image. In other words, an approximate medial atom placed anywhere in an image with any initial parameters can be optimized to a mathematically valid core sample which may or may not represent an object of interest. For this reason, the cores presented in this dissertation do not end within an image, and consequently connectors need not be considered further.

2.5.4 Tubular Cores

Cores, as presented here, can have co-dimension of either 1 or 2. Generically, cores are of co-dimension 1, so cores of objects in 3D images are normally 2-manifolds. The situation can arise, however, in which a core passes through a singular location where the object it describes has a circular cross-section. If the involutes of a medial atom align with a circular cross-section of the object, the core becomes 1D, *i.e.*, a space curve, over an infinitesimal interval, and the frame of the medial atom is ill-defined. When extracting cores of approximately tubular objects like blood vessels, involutes are likely to align with nearly circular cross-sections, causing the extraction of 2D cores to be unstable. In order to stabilize this process, we can force cores to be 1D along their whole length when extracting objects that we know to be nearly tubular. For the remainder of this dissertation I will use the term "tubular" to refer to any object extracted using a 1D core, whether it is in fact tubular or not according to the mathematical definition. Similarly, I will use the term "slab-like" to refer to any object extracted using a 2D core. The choice of whether to use a 1D core or a 2D core may be somewhat arbitrary and is based on the perceived closeness to tubularity of the object being extracted.

When computing the core of a generic slab-like object in 3D, a medial atom traditionally has two spokes, as shown in fig. 2.5. If the core is constrained to 1D, the two spokes are ideally replaced by an infinite number of concentric spokes but in practice are replaced by a finite number of spokes greater than or equal to two, as shown in fig. 2.10. I discuss the choice of how many spokes to use in section 3.3. Medialness is computed in this dissertation by placing a directional derivative of a 3D Gaussian at the tip of each spoke and integrating image values as weighted by the Gaussian derivatives. For 3D tubular cores this computation is given by the equation

$$M(\underline{m}) = \sum_{\vec{v} \in V} D_{\hat{\vec{v}}} I(\underline{x} + \vec{v}, \sigma) \tag{2.2}$$

where **V** is the set of concentric vectors obtained by rotating the first spoke, $\vec{p}$ (fig. 2.5), about the vector, $\vec{b}$. σ is the scale of interrogation, *i.e.*, the standard deviation of the

Gaussians. The details of tubular medial atoms and tubular cores are presented in sections 3.3 and 3.4.

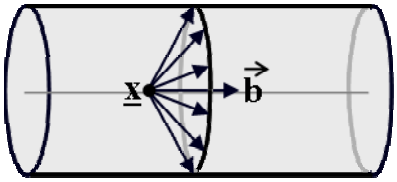

Fig. 2.10. The geometry of a tubular medial atom in 3D. $\underline{x}$ is the medial atom's location and $\vec{b}$ is the spoke bisector at $\underline{x}$. The spokes are concentric with uniform radius r, object angle θ, and tails located at $\underline{x}$.

2.5.5 Core-Marching

Previous subsections of section 2.5 have described medial atoms, both slab-like and tubular, and methods to determine the location and parameters of such atoms by optimizing their medialness values. The next step is to develop a method to locate multiple medial atoms and combine them into a single core that represents a 3D figure. The particular method for core extraction depends on whether the figure is tubular or slab-like, *i.e.*, whether the core being extracted is 1D or 2D.

The most straightforward and commonly used methods for extracting a 1D core of a tubular object rely on marching [Eberly94, Aylward96]. Marching methods begin with a seed medial atom located along the desired core and step a small distance in the core's tangent direction, provided by the seed atom's $\vec{b}$ vector, to predict a new medial atom. This predicted atom is then optimized as described earlier in order to correct both its location transverse to the core and its parameters. This alternating prediction and correction is repeated in an iterative manner, thereby extracting the sampled core one medial atom at a time. The details of the particular marching method used to extract tubular cores are given in section 3.4.

When extracting a 2D core of a slab-like object there is no intuitive means for marching along the core since any potential method must cover a 2D space, and it is not clear in which direction to march. The only existing solution to this problem is the Marching Ridges algorithm [Furst99] for extracting slab cores from 3D images, based on the earlier Marching Cubes algorithm [Lorensen87]. Marching Ridges discretizes the Euclidean space, creating implicit cubes whose vertices are adjacent voxel centers. Given a cube that contains a portion of a core, Marching Ridges computes derivatives of the medialness function at the cube's vertices, edges, and faces to determine where the core exits the cube. Each face through which the core exits indicates an adjacent cube containing a portion of the same core. All adjacent cubes identified are examined to extract the whole core. Marching Ridges has a variant for extracting tubular cores from 3D images, based on the earlier Marching Lines algorithm [Thirion96], which similarly uses derivatives of medialness to determine the two faces through which a 1D core exits a given cube. The sampling rate of Marching Ridges, though, is limited by the prior discretization of the image being analyzed. Section 4.3 of this dissertation presents a novel method for marching along a 2D core, based loosely on the 1D core-marching method described above, that has no inherent sampling limitations. Regardless of the particular method used, the core-marching method is likely to encounter object branches. This situation is discussed next.

2.6 Behavior of Medial Methods at Branches

While cores share numerous properties with Blum's MAT, the two differ significantly in the way they handle object branches due to cores' computation at scale. In the MAT, each segment of a medial axis represents a single figure, or branch, and several medial axis segments meet at a branch point. Furthermore, the corresponding radii of these medial axis segments converge to a shared value at the branch point. In contrast, cores have no explicit branch point. When a 2D object branches, three separate cores generically cross in the vicinity of the branch but not at a mutual intersection point, and the parameters of the cores are generally unequal at the locations where they cross. In 3D images, 1D cores generically do not touch at all, even in the vicinity of branches. 2D cores do intersect near branches but there is little consistency in the geometry and attributes of these intersections.

To better understand the branching behavior of medial axes computed without scale, such as the MAT, I first describe the behavior in the simple 2D situation and then extend the description to the 3D situation, which is more relevant for this dissertation. A 2D object's medial axis is generically composed of points located at the centers of disks that are bi-tangent to the object's boundary (fig. 2.1). Giblin and Kimia refer to such a medial location as an interior point and assign it the label A_1^2 [Giblin03]. In general, Giblin and Kimia describe all medial locations using the notation A_k^n, where n is the number of places at which the inscribed disk is tangent to the object's boundary, or the number of involute locations, and k is the order of tangency at those involute locations. Since an interior medial point has two involutes, each having first-order tangency between the inscribed disk and the object boundary, it is labeled A_1^2. When $n=1$, the superscript is removed from this notation. When a medial location has involutes with different orders of tangency, each involute is described separately. For example, a medial location with one involute of first-order tangency and another involute of third-order tangency is labeled $A_1 A_3$ (see below). This notation of Giblin and Kimia is only used in sections 2.6 and 2.7 of this dissertation to help classify types of medial points. The types of medial points are relevant for the rest of this dissertation but the notation itself is not.

At a generic medial branching location in 2D, where three medial curves intersect, there is a single A_1^3 point (fig. 2.2). In other words, there are three involutes each having order 1 tangency. Thus, medial axes in 2D are generically composed of A_1^2 curves that intersect at A_1^3 points. Branch points with more than three involutes are also possible, but are not generic. For example, an A_1^4 point can occur when four branches intersect, but an infinitesimal perturbation of the object boundary will destroy the A_1^4 point and create two new A_1^3 branch points instead.

In 3D, the same generic medial locations are present and occur along manifolds of dimension one higher than in 2D (fig. 2.11). That is, A_1^2 interior points generically form sheets in 3D (rather than the curves they form in 2D) and A_1^3 branch points generically form

curves at the intersections of medial sheets [Giblin00]. In 3D, there are two other generic medial locations that are not generic in 2D: $A_1 A_3$ points occur at the intersection of an A_1^3 curve with an A_3 curve (see section 2.7) and A_1^4 points (not shown in fig. 2.11) occur at the intersection of four A_1^3 curves.

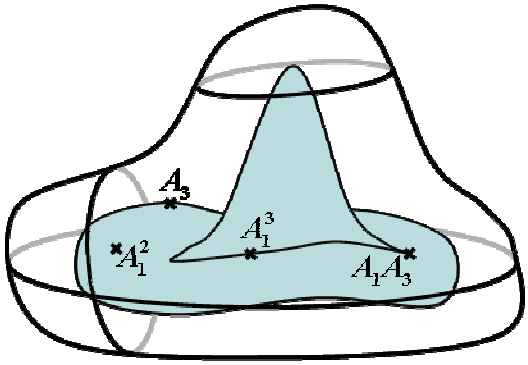

Fig. 2.11. The medial axis of a branching 3D object showing four of the five different types of generic points as defined in [Giblin00].

Cores, as mentioned, are computed at scale and do not have branch points. As a result, any marching method used to extract a core (see subsection 2.5.5) cannot simply march along the core until a branch point is reached and then follow the resulting medial manifolds. Rather, indications must be sought during core-marching that a branch is nearby, and when such an indication is found the locations of the branching medial manifolds must be determined. Cornerness of the object boundary, discussed next, is an effective indicator of branching.

2.6.1 Cornerness

Certain features of an image, for example, edges or ridges of intensity, are of particular interest in computer vision and image processing. To automatically filter image features, Lindeberg [Lindeberg98] locates the singularities of differential geometric expressions. This method has the desirable property that it is invariant to translation, rotation, and scaling, so the same features will be detected regardless of how these

transformations are applied to the image. A well known example of such a feature detector is Canny's non-maximum suppression edge detector [Canny86].

A feature detector that is of interest in this dissertation due to its ability to locate branches is a junction or corner detector [Lindeberg94, ter Haar Romeny02]. In early research Kitchen and Rosenfeld, among others, used the curvature of level curves as a corner detector in 2D images [Kitchen82]. I extend their corner detector to 3D. In gauge coordinates this detector can be written as

$$\frac{I_{\vec{u}\vec{u}}}{I_{\vec{v}}} \tag{2.3}$$

where $\vec{v}$ is the image gradient direction. In 2D, $\vec{u}$ is the direction orthogonal to $\vec{v}$. In 3D, $\vec{u}$ is the eigenvector corresponding to the largest eigenvalue of the Hessian matrix in the plane normal to the gradient direction $\vec{v}$ [Fridman03]. In Cartesian coordinates this corner detector is written as

$$\frac{I_{\vec{x}}^2 I_{\vec{y}\vec{y}} - 2I_{\vec{x}} I_{\vec{y}} I_{\vec{x}\vec{y}} + I_{\vec{y}}^2 I_{\vec{x}\vec{x}}}{\left(I_{\vec{x}}^2 + I_{\vec{y}}^2\right)^{3/2}}. \tag{2.4}$$

Blom extended the 2D version of this detector to give a strong response at sharply curved edges by multiplying it by the gradient magnitude cubed [Blom92], giving

$$I_{\vec{u}\vec{u}} I_{\vec{v}}^2 \tag{2.5}$$

in gauge coordinates and

$$I_{\vec{x}}^2 I_{\vec{y}\vec{y}} - 2I_{\vec{x}} I_{\vec{y}} I_{\vec{x}\vec{y}} + I_{\vec{y}}^2 I_{\vec{x}\vec{x}} \tag{2.6}$$

in Cartesian coordinates. This detector is affine and skew invariant [Lindeberg94]. It is the same in 3D as it is in 2D in the sense that applying the 3D version to a volume image yields

an identical response as applying the 2D version to each slice of the 3D image individually [Fridman03].

The magnitude of the response of this detector, a measure which I will call "cornerness," is an indicator of the curvature of edges in an image. This measure responds to edges of objects at locations of concave branching in 2D or hyperbolic branching in 3D, where the response is proportional to edge curvature. In this dissertation, cornerness is constantly probed during core-marching at medial atom spoke tips, which are located at putative edges in the image being analyzed, and local maxima in cornerness are viewed as indicators that an object branch is present.

2.7 Behavior of Medial Methods at Ends

The use of scale does not have the same effect on medial axis ends as it does on branches. Still, the MAT and the cores described in this dissertation behave differently at their respective ends. First I describe Giblin and Kimia's medial point classification system as it applies to ends of medial axes. Recall that 2D medial axes are described as A_1^2 curves that intersect at A_1^3 points. The only other type of medial point generically found in 2D is the medial axis end point, A_3 (fig. 2.2) [Giblin03]. The maximal disk centered at a medial axis end point has a single location of third-order tangency with the object's boundary. At this location of tangency the maximal disk osculates the boundary, *i.e.*, has the same local curvature as the boundary. Moreover, the location of tangency is a local maximum of curvature of the boundary.

In 3D, medial sheets, A_1^2, are bounded on all sides by end curves, A_3 (fig. 2.11) [Giblin00]. Analogous to the 2D case where a maximal disk centered at an A_3 point osculates the object's boundary at a maximum of curvature, a maximal sphere centered along an A_3 curve in 3D osculates the object's surface at a crest point, where one of the two principal curvatures of the surface achieves a local maximum.

Cores, in theory, also end where a maximal sphere osculates an object surface. Since cores are computed at scale, however, they only display this behavior in situations with little or no noise and a smoothly rounded object end. In these situations, medial atom object angles become small near core ends as the atoms' involutes near each other. In practice, the involutes often fail to converge and the core marches past the desired object end. In this dissertation, cores are computed using a fixed object angle of $\pi/2$, not allowing the involutes to converge. The effects of using a fixed object angle are discussed further in section 4.2, and a method for explicitly locating core ends is described in Chapter 6.

Chapter 3. Core-Following for Tubular Objects

3.1 Introduction

This chapter presents the use of 1D cores for the automated extraction of non-branching tubular objects from 3D images. Cores are height ridges of a graded measure of medial strength called medialness, computed at scale. As discussed in section 2.5, cores have the ability to robustly extract object geometry, as well as the objects themselves, directly from noisy grayscale image information. When the objects are known to be tubular or nearly tubular, we can constrain cores to be 1-manifolds (as opposed to generalized 2D cores; see subsection 2.5.4). The constraints applied to these tubular cores improve their robustness, allowing them to accurately and consistently track tubular objects in images with a high level of noise.

The chapter begins with a brief description of a filtering technique [Westin01] which has been applied as a pre-processing step to core extraction through collaboration with Dr. Carl-Fredrik Westin and Raúl San José Estépar at Brigham and Women's Hospital and Harvard Medical School. Section 3.3 then details how tubular medial atoms (fig. 2.10), the discrete elements from which cores are composed, are implemented in this dissertation. Finally, section 3.4 describes the extraction of 1D cores via a marching method that iteratively locates medial atoms. These last two sections build heavily on other work done at the University of North Carolina on cores, medialness, and ridge marching [Eberly94, Morse94, Fritsch95, Aylward96, Furst98, Morse98, Pizer98, Furst99, Aylward02].

3.2 Image Pre-processing

The analysis of medical images is hindered by the noise present in those images. Small structures are often obscured and mistaken for image noise by automated methods. Conversely, randomly aligned noise can be incorrectly identified as structure. Simple filters

such as the low-pass Gaussian filter effectively reduce noise but also blur edges within an image, making small objects and weak edges difficult if not impossible to detect. A variety of approaches have been developed with the goal of suppressing image noise while maintaining or even enhancing relevant structure, including anisotropic diffusion [Perona90, Gerig92], nonlinear filtering [Du95], and directed low-pass filtering [Chen95].

A recent approach to this problem, based on anisotropic adaptive filtering [Knutsson92], has been developed by Westin *et al.* [Westin01] for processing magnetic resonance angiography (MRA) data. This method uses three steps to reduce noise in low frequency regions and enhance relevant structures in the image. First, it employs quadrature filters [Knutsson89] to estimate local orientations of image structure at every place within the image, exploiting the assumption that local orientation can be described by lines and planes. It represents these local orientations using tensors. Second, it stabilizes the estimates from step 1 by applying a Gaussian low-pass filter to the computed orientation tensors. Third, it filters the original image in an adaptive fashion, using a combination of one isotropic low-pass filter and six directed high-pass filters, where the application of the directed filters is controlled by the orientation tensors from the first two steps. This method has shown excellent results in reducing high frequency noise in MRA data while enhancing small tubular objects such as blood vessels.

Before tubular core computation, I apply this adaptive filtering technique to the image. Pre-processing the image increases the signal-to-noise ratio and improves the success of the core-following algorithm.

3.3 Tube Medial Atoms as Implemented

Tube medial atoms in 3D were introduced in subsection 2.5.4 as the basic elements of 1D cores. Each tube medial atom is a structure having a hub at a given location $\underline{x}$ and a set of concentric spokes defined by the parameters $(r, \vec{\underline{F}}, \theta)$ – a radius, a frame, and an object angle. An image kernel can be created from a tube medial atom by centering a directional derivative of a 3D Gaussian at each spoke tip of the atom, with the derivative taken in the spoke direction. This kernel, when applied as a weighting function to an image, acts as a

robust medialness filter, responding strongly at image locations that appear as middles of tubular objects, at scale. Such medial atom kernels can be used to automatically compute the medial location, radius, and orientation information of tubular objects in 3D grayscale images.

Medialness computations on tubular objects are more stable and robust when using a kernel built from an explicit multi-spoked tube medial atom (eq. 2.2; fig. 3.2) than when using a kernel built from a 2-spoked medial atom (see eq. 2.1 and fig. 2.9). The spin angle of a 2-spoked atom is undefined on a perfectly tubular object, and it is defined but computationally unstable on a nearly tubular object (see subsection 2.5.1). Furthermore, the multi-spoked atom is more robust than the 2-spoked atom due to the additional constraints imposed on it – the concentricity of its spokes makes it less likely to lose track of an object in the presence of image noise than the 2-spoked atom.

The multi-spoked atom is also more robust since the Gaussians placed at the spoke tips during medialness computation extend further along the boundary of the object than do the pair of Gaussians in the 2-spoked atom, thereby acting more effectively as edge detectors. The pair of Gaussians in the 2-spoked atom can be adjusted to cover the same along-boundary area as the set of Gaussians in the multi-spoked atom by increasing their standard deviations, but this creates two undesirable side effects: 1) the across-boundary extent of the Gaussians is increased, making the medial atom more vulnerable to effects from nearby objects; and 2) the fit of the Gaussians to an object boundary deteriorates more rapidly at locations where the boundary is sharply curved. Experiments have shown that the tube medial atom correctly identifies significantly more tubes than does the 2-spoked atom, particularly in high noise situations, regardless of the extent of the Gaussians. For example, the experiment of subsection 7.3.1.1.1 for tubular core-following was repeated using 2-spoked atoms rather than multi-spoked atoms. When using 2-spoked atoms, 367 out of 480 cores were correctly followed (with no deviations of more than one voxel from the true medial axis), whereas when using 8-spoked tube atoms, 455 out of 480 cores were correctly followed. The choice to use 8 spokes is motivated later in this subsection.

The multi-spoked tube medial atoms used in this dissertation are an adaptation of other medial atoms used to track cores at UNC. They can be compared with the medialness kernels that Furst extended from 2D to 3D for finding tubes [Furst99]. He extended both the Fritsch medialness kernel and the Morse medialness kernel (left and right sides of fig. 2.7, respectively) to 3D by "stretching" them along the z-axis so as to have a Gaussian profile in that direction. Furst found that the extended Fritsch kernel, as in 2D, is overly sensitive to image intensities within an object due to the strong response at the kernel center. The extended Morse kernel proved much more adept at sensing object boundaries, but it suffers from a high computational cost.

Aylward developed a different type of medialness kernel for determining widths of tubular objects in 3D images as part of his intensity ridge tracking method that separately searches for position and width information [Aylward02]. His kernel is a medial atom having eight spokes with the object angle θ fixed at $\pi/2$ and a pair of spheres aligned at each spoke tip (fig. 3.1). The inner spheres are positive in value, responding strongly to bright regions in the image, while the outer spheres are negative, preferring dark, or background, regions. Aylward has shown that this kernel is effective for determining tubular object widths once the object's medial location has been found but too sensitive to image noise to be used for finding medial locations.

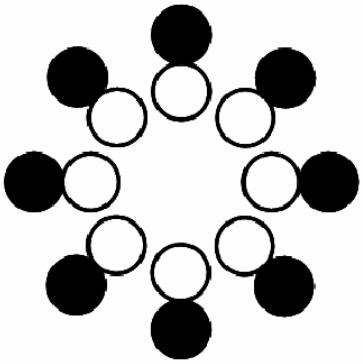

Fig. 3.1. A 2D representation of the binary 3D medialness kernel used in [Aylward02] to determine widths of tubular objects.

This dissertation combines the ideas of Furst and Aylward to create a medial atom capable of effectively locating tubular objects in 3D. The atom is much like Aylward's kernel in that it has eight concentric spokes and an object angle fixed at $\pi/2$. It draws from Furst's efforts in that each pair of spheres from Aylward's kernel is replaced by a volumetric directional derivative of a Gaussian, where the derivative is taken in the along-spoke direction (fig. 3.2). I define the Gaussians' standard deviations to be 0.25 times the atom's radius, with a minimum standard deviation of one voxel, for reasons discussed in section 6.2. The Gaussian derivatives act as edge detectors, much like the pairs of spheres in fig. 3.1, while at the same time blurring the image to reduce the effects of noise. This kernel is more expensive to compute than that of Aylward but is preferred when trying to locate objects in noisy images. It avoids the oversensitivity to intensities within an object present in the extension of Fritsch's kernel by interrogating the image near object boundaries only. At the same time it is more efficient than the extension of Morse's kernel since its computation is simpler, and it is mathematically equivalent to that kernel in the limit as the number of spokes approaches infinity [Furst99].

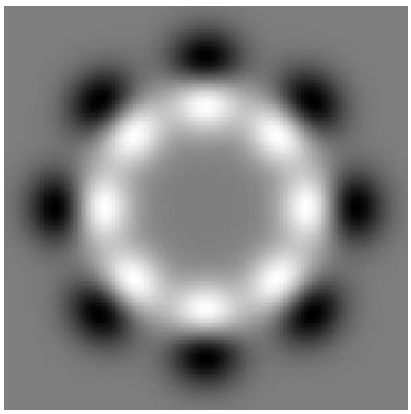

Fig. 3.2. A slice through the middle of the 3D tubular medialness kernel used in this dissertation.

While this tubular kernel is computationally more efficient than the extended version of Morse's kernel, the concern exists that it is less accurate, as it is an approximation of the Morse kernel when implemented using a finite number of spokes. The more spokes are used, the better the approximation, but computation time increases almost linearly with the number

of spokes. To decide how many spokes to use, kernels with differing numbers of spokes were used in test runs to extract hundreds of tubes of varying curvatures, such as the ones described in subsection 7.3.1.1, from noisy synthetic images. Of the tubes correctly extracted using the (non-approximating) extended Morse kernel, the 8-spoked kernel from fig. 3.2 failed less than 1% of the time, and all of those failures occurred in images with the highest level of noise tested. When a kernel having only four spokes was used, the failure rate increased to 14%, which was deemed unacceptable. All further results shown in this dissertation were achieved using a kernel with eight spokes.

Another approximation employed here is the constraining of the medial atom object angle θ to $\pi/2$ radians. This approximation was originally implemented according to the suggestion of Dr. Stephen Aylward, UNC-Chapel Hill, for the sake of improving running time since it reduces the dimensionality of optimization by 1 (see subsection 2.5.3). This constraint corresponds, strictly speaking, to no narrowing or widening of the tubular object being tracked, but in practice it has an insignificant mathematical effect on the fit of the medial atom to a tubular object with a slow rate of widening or narrowing, such as a blood vessel. This effect is discussed more precisely in section 4.2. Furthermore, this constraint was found to improve the robustness of the method. When the object angle is unconstrained, the medial atom is more sensitive to the effects of image noise, particularly in cases of narrow objects and low signal-to-noise. Unless the object of interest narrows or widens rapidly, a medial atom with a constrained object angle produces more consistent results than an unconstrained atom as it is less susceptible to noise. The code written for this dissertation has a switch that allows for θ to be either constrained or unconstrained, but all results shown in this dissertation were achieved with θ constrained to $\pi/2$.

Similarly, I have chosen to constrain the spokes to all be of the same length. Since I use a set of concentric spokes of equal length with Gaussian derivatives taken in the directions of the spokes, medial atom optimization favors perfectly tubular objects. In practice, the method handles objects that are not quite tubular, but the more the object deviates from tubularity, the less likely it is to be correctly located. Two other possible options that have not been tested are as follows. First, the constraint can be loosened to

allow the arrangement of the spokes to vary in an elliptical manner. In other words, the spoke tips can be constrained to lie along an ellipse rather than a circle, and the eccentricity and orientation of this ellipse can be introduced as new parameters for optimization. While this change would increase computation time, it could potentially improve performance in situations where object cross-sections can become significantly elongated. It might, on the other hand, decrease overall robustness due to the loosened constraint, as described in the previous paragraph for the case where θ is unconstrained.

Finally, although the medial atoms, as implemented here, are effective at segmenting tubes in noisy images, they can be inaccurate on heavily curved tubular sections. At locations where a tube bends, the medial atoms consistently err in locating the tube in the direction of the tube's center of curvature. This inaccuracy is due to the blurring effect of the Gaussians that make up the medialness kernel, as Gaussian blurring tends to straighten image edges. The larger the standard deviation of the Gaussians used, the stronger this effect becomes.

These tube medial atoms, as presented, can be used to automatically determine the medial location, radius, and orientation information of a tubular object within an image. Each such medial atom represents a discrete point along the core of the object, and a 1D core is found via a marching algorithm, presented next.

3.4 Marching Along a 1D Core

The core of a non-branching tubular object is, mathematically, a continuous curve that is the medial axis, at scale, of the object. In practice, such a core is composed of a series of medial atoms, each of which has been fit to the image data. A seed medial atom is normally initialized at one end point of the desired core and a marching algorithm is initiated at that seed point, iteratively taking steps forward in the core tangent direction and re-locating these new medial atoms by climbing the medialness function to its ridge. This optimization is performed with respect to a medial atom's position, radius, and orientation. Bifurcations are handled separately, as will be described in Chapter 5.

In the method presented here, the user initializes the core-marching algorithm by inputting an estimated medial atom at one end of a tubular object. This requires seven parameters – the medial atom's x, y, and z coordinates, its radius, and the altitude, azimuth, and spin angle that define its frame $\underline{\vec{F}}$ (see subsection 2.5.1). The particular initial value of the spin angle is irrelevant, as explained later, so it is automatically initialized to 0. The six remaining user-defined parameters are selected in an interactive fashion. Due to the effectiveness of the optimization process described below, these initial values need not be accurate. Depending on the local properties of the image, it is enough to initialize the location of the seed medial atom anywhere within the desired cross-sectional area of the tube. In other words, the location can be off by as much as the radius of the tube. The initial radius estimate can generally be off by a factor of at least 2. The initial altitude and azimuth that define the core tangent direction need only describe a vector in the same hemisphere as the actual core tangent, specifying which marching direction is forward and which is backward. The selection of these six initial parameters is the only user interaction required in the entire tube extraction process presented in this dissertation, whether the tubular object branches or not.

Once the seed medial atom is roughly initialized, its parameters must be corrected. As described in subsection 2.5.3, this is done by maximizing the medialness of the atom, which, for a given set of parameter values, is computed by integrating the image intensities weighted by the chosen medialness kernel (eq. 2.2). When computing optimum parameter cores of tubes (see subsection 2.5.3.1), medialness is maximized over the specified set of non-Euclidean parameters and over two orthogonal directions spanning the plane orthogonal to the core tangent direction. The spin angle φ is included as an optimization parameter to account for the discretization error produced by using an 8-spoked atom as an approximation to an isotropic medialness kernel. It is sufficient to optimize over a spin angle of 45 degrees due to the 8-directional symmetry present. Qualitative experiments I conducted comparing the performance of medialness optimization with and without the inclusion of φ indicated that the optimization process is less stable when not optimizing over φ. Since the spokes necessarily lie in the core normal plane and are already computed, I arbitrarily choose two of the spokes that form a right angle as the two Euclidean optimization directions.

Given these two Euclidean optimization directions and four optimization parameters (radius, altitude, azimuth, and spin angle), optimum parameter cores are computed by simultaneously maximizing medialness over all six linearly independent directions transverse to the core tangent direction. I originally implemented such a six-dimensional optimization with one modification. Since the location of a medial atom is found by optimizing over the normal plane to the core tangent at the same time as the core tangent direction itself is being optimized, this is a coupled problem. This problem is solved by computing an approximation in which the Euclidean directions are chosen in the plane normal to the core tangent direction with which the optimization is initialized. In other words, as the tangent direction changes throughout the optimization process, the Euclidean optimization directions do not. This modification does not introduce error to the medial atom location, it simply finds a medial atom at a slightly different place along the length of the core than would have been found otherwise. Furthermore, the kernel used to compute medialness is recalculated on each iteration of the optimization to be normal to the current tangent direction.

This implementation proved to be computationally expensive due to the six-dimensional optimization and was modified to compute optimum parameter cores in two steps. First, the location of a medial atom is optimized while keeping the initial parameter approximations fixed. Once this two-dimensional optimization is complete and the atom location is set, a four-dimensional optimization is performed to maximize medialness with respect to the medial atom parameters. While this method computes an approximation of optimum parameter cores, trials have shown that results are not significantly poorer. Furthermore, this order of optimization obviates the need to compute connector curves (see subsection 2.5.3.2.1). Either the two step optimization or the full six-dimensional optimization can be selected using a switch, but all further results shown here were achieved using the two step optimization. In either case, medialness is optimized using a conjugate gradient method [Polak71, Golub89, Golub96] written in MATLAB by Hans Bruun Nielsen and available at http://www.imm.dtu.dk/~hbn/Software/conj_grad.m.

This medial atom optimization method, applied to the manually initialized seed atom, provides a starting location along the desired tubular core. The location of a second medial

atom is then predicted by taking a small step from the location of the initial atom in the direction of its computed tangent. The step size is initialized at one voxel but is decreased automatically at locations where the core bends sharply. Specifically, whenever the tangent directions of adjacent medial atoms differ by more than 15 degrees, the step size is halved. These values were chosen according to the results of the experiment described in subsection 7.3.1.1.2. This adjustment decreases the detrimental effects of the poor linear approximation of the core's shape by its tangent direction at places of high curvature, while at the same time not slowing program execution when the core is fairly straight. The parameters $(r, \vec{F})$ of the new medial atom are predicted by simply setting them equal to the parameters of the initial atom, a reasonable prediction given the small step sizes. The predicted atom is then corrected using the described conjugate gradient optimization. This prediction and correction process, when iterated, extracts a sampled core that is equal to the continuous core in the limit as the step size goes to zero. Using this marching algorithm, in combination with the branch-handling and core-termination methods described in Chapters 5 and 6, respectively, results in the accurate automated extraction of complex branching tubular structures from 3D images, as measured in Chapter 7.

Chapter 4. Core-Following for Slab-Like Objects

4.1 Introduction

This chapter extends the 1D core extraction method of Chapter 3 by utilizing 2D cores to extract non-tubular, or slab-like, objects from 3D images. This extension generalizes the core-following algorithm for use on arbitrarily-shaped single-figure objects, where a figure is a structure that can be represented by a non-branching core. The method for extracting slab-like cores differs from the method presented in Chapter 3 in that it employs a medial atom with two spokes rather than eight and produces a sampled medial sheet rather than a sampled medial curve. This method is less constrained than the tubular core-following method and consequently less robust. I add new constraints to improve the resistance of the slab-like core-following method to image noise.

Section 4.2 describes how the medial atoms used in following slab-like objects differ from the tube medial atoms discussed in section 3.3. Section 4.3 presents a novel coarse-to-fine grid-based marching scheme that utilizes the slab medial atoms to follow 2D cores. The last section of this chapter describes geometric penalties imposed on slab atoms that constrain the variability between adjacent atoms and improve the robustness of the slab-like core-following method.

4.2 Slab Medial Atoms as Implemented

Unlike the 8-spoked tube medial atoms presented in section 3.3, each slab medial atom has only a pair of spokes that extend from the atom's central hub to the two involute locations on opposing object boundaries. Otherwise, slab atoms are similar to tube atoms in that they are structures defined by the 4-tuple $(\underline{x}, r, \vec{\underline{F}}, \theta)$ that gives the atom location, radius, orientation, and object angle.

As with tube medial atoms, I fix the object angle θ of the slab atoms to $\pi/2$, resulting in a medial atom with two collinear spokes. This is crucial to improve the method's stability for slab atoms even more than for tube atoms since there are only two spokes. Without this constraint each spoke would be individually susceptible to image noise, and trials have shown that results are inconsistent when θ is unconstrained. With this constraint, the "forward" direction $\vec{b}$ (see subsection 2.5.1) is no longer defined since the medialness filter is symmetric about the axis created by the two spokes. This issue will be revisited in the next section.

Constraining θ to $\pi/2$ implies that opposing object boundaries are parallel; as the boundaries deviate from parallel, the fit of the Gaussian derivatives to the image data deteriorates and the resulting core-following algorithm becomes less robust. To be precise, the medialness response of a medial atom with two collinear spokes, assuming no discretization errors and an object that is symmetric about the core, is proportional to the cosine of the object's rate of narrowing. Accordingly, slow rates of narrowing have almost no detrimental effect on core-following. Near the ends of rounded objects such as kidneys, core-following does become significantly less robust, but even when an object narrows at a rate of 45 degrees, the response is still at $1/\sqrt{2}$ of its full strength – enough of a response for effective following in most situations.

I compute the medialness of this 2-spoked atom in a manner similar to the computation of tube medialness, as described in section 3.3. A directional derivative of a volumetric Gaussian is centered at each spoke tip, where the derivative is taken in the direction of the spoke. This results in a filter that, when applied as a weighting function to an image, returns a scalar medialness value (see eq. 2.1). The next section discusses the optimization of slab atoms with respect to medialness and the subsequent marching algorithm used to combine slab atoms into a 2D core.

4.3 Marching Along a 2D Core

As with tubular core-marching, slab-like core-marching requires a method for stepping from a computed medial atom to a new predicted atom and then correcting the location and parameters of that predicted atom. This marching process is more complex with slab-like cores, which are 2D, than it is with 1D tubular cores. While the tangent vector at a given location along a 1D core provides a clear forward direction, marching along a 2D core is ambiguous. At each location along a slab-like core there exists a tangent plane, and for a marching algorithm to extract the core of such an object, it must choose marching directions spanning the tangent plane in such a way that no part of the medial sheet is missed. The method presented in this dissertation covers the whole medial sheet in a discrete manner using a grid-based predictor-corrector marching algorithm. The method then refines the computed medial grid in a coarse-to-fine approach.

Given an approximate seed or parent medial atom that has been corrected, as described later in this section, the algorithm marches in four directions from that atom's location – call these directions North, South, East, and West – with the goal of finding four potential new atoms that will be placed on a "predicted atoms" queue. I define the North direction as the parent atom's $\vec{b}$ direction, arbitrarily chosen in the tangent plane of the parent. The East direction is then defined as being orthogonal to both the North direction and the direction of the spokes (left side of fig. 4.1). These two directions span the tangent plane to the core at the location of the parent atom. A step is taken from the location of the parent atom in each of the four compass directions to predict the locations of four potential new atoms. The parameters of these potential atoms are predicted as follows. Each atom is assigned the same radius as the parent atom from which it was predicted. This choice is the most obvious guess given an initial lack of knowledge about the new potential atom's radius. The frame of each atom is defined by setting its $\vec{b}$ vector as the direction of the step taken to it from the parent atom and setting its spokes parallel to the spokes of the parent atom. Again, the spoke orientation is left unchanged from the parent atom since there is typically no reason to assume a different orientation. The motivation for the chosen $\vec{b}$ vector is the desire to produce a nearly rectilinear medial grid to both improve its stability and simplify later computations. We wish to create implied connections from a medial atom to its 4

potential neighbors (right side of fig. 4.1). The same results could be achieved using a vector rotated 90 degrees from the $\vec{b}$ vector in the tangent plane to the core, but the choice defined here facilitates the determination of marching directions on subsequent iterations. Once the medial atom parameters are initialized as described, these four potential atoms are placed on the "predicted atoms" queue.

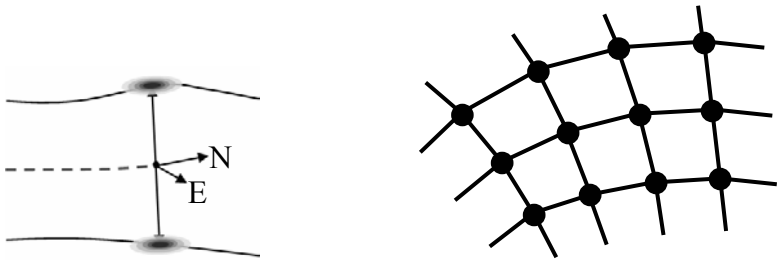

Fig. 4.1. Left: Marching along a slab-like core in 3D, in which case the core is 2D and each medial atom has two collinear spokes. The dotted curve represents part of the computed core, in profile. Right: A view of a computed slab-like core from above, where dots represent medial atoms and line segments represent implied connections between atoms. Each atom in the interior of an object is connected to 4 neighbors.

The corrector portion of the predictor-corrector method then takes a potential atom from the head of the queue and optimizes its position, radius, and orientation. This optimization is similar to that used for tube atoms in which medialness is maximized via a two step modification of the optimum parameter method (see section 3.4). The first step, Euclidean optimization of the medial atom location, is simpler for slab atoms than it is for tube atoms. Whereas in the case of tube atoms the method must choose two directions of optimization orthogonal to the core tangent, here there is a single predetermined direction orthogonal to the core tangent, *i.e.*, the direction of the spokes. The second step is akin to that of tube atoms in which the four atom parameters are optimized for medialness using a multi-dimensional conjugate gradient method. For slab atoms, however, the optimization is underconstrained since rotation of the medial atom about the spokes results in unchanged medialness values. Fortunately, all directions of optimization are transverse to this rotation, so the conjugate gradient optimization process is stable given an initial vector $\vec{b}$.

Once the predicted atom is corrected, it is tested to determine if it meets two conditions: 1) it is still on the core, *i.e.*, has not fallen off the end of the core; and 2) it is not too close in position to an existing atom. The first test is the subject of Chapter 6. The second test considers a new atom too close to an existing atom if it is within a distance 0.75 times the marching method's step size. The factor of 0.75 is somewhat arbitrary, although it must be less than 1, so that valid atoms are not discarded, without being too far below 1, which would make the chosen step size meaningless. If the atom fails either of these two tests, it is discarded. Otherwise, the atom is saved as part of the core and its own North, East, and West atoms are predicted and placed on the queue. Except for the first iteration of marching, a medial atom $\underline{m}$ need not predict a new atom in the South direction since that new atom would nearly duplicate the atom from which $\underline{m}$ was predicted. Other predicted atoms will also duplicate existing atoms, but those duplicates will be discarded later by the second test described earlier in this paragraph.

This grid-based marching scheme effectively covers cores of single-figure slab-like objects. It works well in noise-free images, but the introduction of noise leads to the introduction of error. Since slab-like cores are less stable than tubular cores (see section 3.3), minor inaccuracies arise in the computation of slab atoms in the presence of noise, and after a large number of steps these inaccuracies compound and become unacceptable. This effect can be reduced by augmenting the grid-based marching scheme to work in a coarse-to-fine manner. The core is first computed at a coarse rectangular sampling using a step size such that the object of interest is represented by a grid roughly five medial atoms in diameter. This coarse sampling does not allow enough steps for errors to compound significantly. The coarse core is then sub-sampled as many times as necessary to produce the desired level of detail. Each round of sub-sampling is performed as follows. Each "cell" of the core grid defined by four adjacent medial atoms is partitioned into four new cells (fig. 4.2). This is accomplished by predicting five new medial atoms. One is predicted in the center of the cell by averaging both the locations and the parameters of the four corner atoms. The other four are predicted by averaging the locations and parameters of adjacent pairs of corner atoms. These five predicted atoms are then corrected using the optimization method described earlier. In cells adjacent to ones that have already been refined, all five new atoms need not

be predicted and corrected. For example, when refining the cell to the right of the one already refined in fig. 4.2, the right-most red atom is reused and only the other 4 are predicted and corrected.

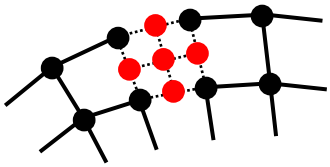

Fig. 4.2. Sub-sampling a slab-like core. Black dots represent previously computed medial atoms at the existing coarse scale, red dots represent newly sampled atoms, solid line segments represent implied connections between computed atoms, and dotted line segments represent implied connections to new atoms.

In sub-sampling the core, atoms are also added outside of the border of the original, coarsely sampled, core. The location of each new atom is predicted by either a) taking half a step out from an existing boundary atom or b) averaging the locations of two adjacent boundary atoms and then taking half a step out (fig. 4.3). The parameter values are predicted as the parameter values of the parent atom in case (a) or the average parameter values of the two parent atoms in case (b). The atom is then corrected and tested to see if it still lies on the core (see Chapter 6). This sub-sampling outside of the boundaries of the original core differentiates this marching scheme from the coarse-to-fine paradigm used in m-reps [Pizer03a]. The two are further differentiated in that the marching scheme presented here is coarse-to-fine only in sampling and not in the scale of interrogation used to locate the medial atoms. It uses the same scale of interrogation at each level of sampling, whereas the m-reps method uses a scale of interrogation proportional to the sampling scale.

With the combination of grid-based marching and the refinement of an initial coarse sampling, the 2D core-following algorithm is well suited for the extraction of slab-like objects from 3D images. Even with these attributes, though, it is significantly less robust than the 1D core-following algorithm in noisy images. To improve performance, geometric penalties are imposed on slab medial atoms.

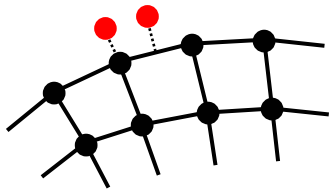

Fig. 4.3. Sub-sampling a slab-like core outside of the current core boundary. Black dots represent previously computed medial atoms at the existing coarse scale, red dots represent newly sampled atoms, solid line segments represent implied connections between computed atoms, and dotted line segments represent connections to new atoms. The locations of the new atoms are predicted by either a) taking a step out from a boundary atom or b) averaging the locations of two adjacent boundary atoms and then taking a step in a direction orthogonal to the line connecting the two boundary atoms and in the tangent plane to the core. The step size is half the step size from the previous level of sampling.

4.4 Penalties on Medial Atoms

Geometric penalties are placed on medial atoms during their optimization to further stabilize slab-like core extraction. These penalties are meant to constrain atoms in ways consistent with the expected geometry of anatomic objects, thereby decreasing the likelihood that an atom's spokes will be drawn away from object boundaries by noise. The penalties are applied in proportion to the size of geometric differences between adjacent medial atoms to deter unrealistic geometries, but no penalties are applied if the differences are below a threshold, chosen as explained below. No penalties are applied in these situations in order to not bias the results when the geometric changes are acceptable.

The geometric penalties are applied during the corrector phase of marching that optimizes medialness. For a medial atom $\underline{m}$ with neighbors $\underline{n}_i$ for $i \in D \subseteq \{N, S, E, W\}$, two terms are added to the medialness equation (eq. 2.1). These terms penalize differences in geometry between adjacent atoms and consequently decrease the likelihood of sudden changes in the object's geometry. The first term penalizes medialness if an atom's radius differs significantly from those of its neighbors but not if the differences are small and consistent with normal width changes in anatomic objects. Large differences correspond to abrupt widening or narrowing of an object, and extreme changes of this type indicate that the object is not being extracted correctly. The second term penalizes medialness if an atom's

spoke orientation differs from those of its neighbors by more than some threshold, specified below. These differences correspond to twisting of the object, and only a reasonable amount of twisting is acceptable.

For each of these two parameters (atom radius and spoke orientation), I choose not to penalize medialness for a small difference in values since gradual widening/narrowing and twisting are normal in anatomic objects. To be precise, let the change in radius $\bar{r} = \left| \underline{m}.r - mean(\underline{n}_i.r) \right|$ for $i \in D$, where $\underline{m}.r$ is the radius of atom $\underline{m}$, and let s be the defined step size of the marching algorithm. If $\bar{r} \le s$, no penalty is applied since such a change in radius corresponds to a widening or narrowing rate of up to 45 degrees, an amount commonly seen in anatomic objects. Beyond that threshold, I apply a penalty of $\left(\bar{r} - s\right) * \frac{M(\underline{m})}{mean(\underline{n}_i.r) - s}$ for $i \in D$, where $M(\underline{m})$ is the medialness value of $\underline{m}$ before applying penalty terms. This is a linear term that results in a penalized medialness value of 0 if the change in radius is equal to the radius of the neighboring atom or atoms. This choice prevents medial atoms from shrinking to a radius of 0.

Similarly, changes in spoke orientations between neighboring atoms are penalized beyond a chosen threshold. Let the change in spoke orientation $\bar{\theta} = mean\left(\angle(\underline{m}.\vec{p}, \underline{n}_i.\vec{p})\right)$ for $i \in D$, where $\angle(\underline{m}.\vec{p}, \underline{n}_i.\vec{p})$ denotes the angle between the spokes of atoms $\underline{m}$ and $\underline{n}_i$. If $\bar{\theta} \le 0.25 * \tan^{-1}\left(\frac{s}{\underline{m}.r}\right)$, no penalty is applied. Beyond that threshold, I apply a penalty of $\left(\bar{\theta} - 0.25 * \tan^{-1}\left(\frac{s}{\underline{m}.r}\right)\right) * \frac{M(\underline{m})}{0.75 * \tan^{-1}(s/\underline{m}.r)}$. The values for these thresholds and penalty weights were chosen empirically to produce desirable results using synthetic and clinical test images, shown in Chapter 7.

During the first (coarse) level of core-following, these penalties are applied to differences in parameter values between the new potential atom and the parent atom from which it was predicted, as that parent atom is the only known neighbor of the new atom.

During subsequent levels of sub-sampling, the penalties are applied to differences between the new atom and up to four known neighbors.

Unconstrained slab atoms are susceptible to the effects of noise, but given accurate parent atoms, the optimization of a new atom is more likely to succeed in the presence of noise when these geometric constraints are applied. Using the coarse-to-fine marching scheme along with the geometric penalties, slab-like core extraction is fairly robust, as shown in subsections 7.2.2 and 7.3.2.1.

Chapter 5. Branch-Handling

5.1 Introduction

This chapter describes methods for locating and traversing branches during the core-based extraction of multi-figure objects from 3D images. This work builds on Chapters 3 and 4, which presented methods for extracting the cores of single-figure tubular and slab-like objects, respectively, directly from image data. The branch-handling techniques presented here work during core-following by constantly probing for features of both the core being extracted and the image itself that indicate a bifurcation. Once a bifurcation is found, the core-following method extracts the new branches.

Sections 5.2 and 5.3 discuss properties of cores at bifurcations of tubes and slabs to motivate both the need for and the workings of the branch-handling techniques. Section 5.4 details a method for detecting object bifurcations that probes for potential bifurcations and then discards false positives. Sections 5.5 and 5.6 then explain how the core-following method resumes along the new branches once a bifurcation is located. The chapter ends with a description of how these methods are combined to extract cores of objects that have both slab-like and tubular figures. Much of the work presented here, particularly in sections 5.2 through 5.4, is inspired by previous work on singularities of cores [Damon98, Miller98, Damon99, Keller99, Miller99, Damon03a, Damon03b] and properties of cores at object bifurcations [Aylward96, Bullitt01, Aylward02].

5.2 Properties of Cores at Tube Branches

The tubular cores defined in this dissertation mathematically do not have branch points; they are curves in space. Hence, as a core approaches a branch in a tubular object the core does not bifurcate to follow the two new branches. Nor does the core follow one of the

two branches, even if one appears to be the main branch. Rather, the core bisects the branches. I will first explain this behavior in 2D, as it is easier to visualize, and then in 3D.

In fig. 5.1, the core-following algorithm was initialized at the bottom of the figure and marched up the "trunk" of the tree structure to extract the core represented by the red curve. During marching, the two medial atom involutes followed the right and left sides of the trunk. As the core passed through the branching region, the left involute continued following the left-most boundary and the right involute continued following the right-most boundary, resulting in a core that increases in radius and bisects the two branches. Similarly, the core-following algorithm was initialized at the outer ends of each of the two branches, resulting in the two blue cores.

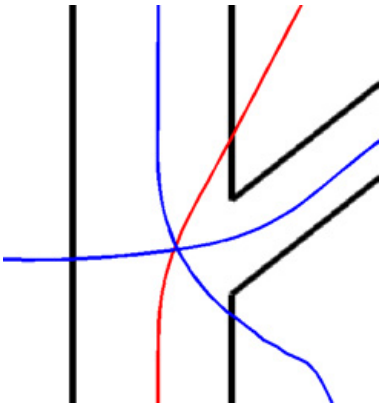

Fig. 5.1. A diagram showing the outline of an ideal branching object in 2D. The three colored curves are the three cores of this object as extracted starting from the outer end of each of the three branches. The object angle θ was allowed to change when computing these results.

These three cores all pass through the object's branching region but they generically do not have a mutual intersection point due to the scale at which they are computed [Damon98]. The three cores do have a mutual intersection point in the limit as the scale used in core computation approaches 0, as well as in certain non-generic cases, such as when one branch continues straight from the trunk and the other branch has a 90 degree branching angle. In the generic case, however, as the scale increases, the three resulting cores diverge from a mutual intersection point (fig. 5.2). Figs. 5.1 and 5.2 were produced using a variable

object angle θ. When θ is fixed at $\pi/2$, as is the case in this dissertation, the core coming from the right branch would stop before the intersection.

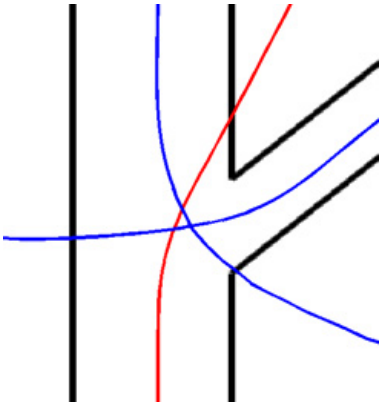

Fig. 5.2. Cores computed from the same branching object as in fig. 5.1 but at a larger scale. The larger the scale used, the farther the three resulting cores are from a mutual intersection point at an object bifurcation.

When computing cores of branching tubular objects in 3D, generically no two cores intersect. Furthermore, while medial axes at 0 scale do share common radius values at their intersection locations, none of the parameters of cores generally agree even at non-generic locations where they do intersect. In addition, while 2D and 3D tubular cores exhibit mathematically similar behavior at object branches, 3D tubular cores exhibit varying behaviors in practice depending on the symmetry of the branches, as discussed next.

5.2.1 Symmetric vs. Asymmetric Branches

When using the implementation presented in this dissertation, computed cores rarely bisect branches of tubular objects in 3D images. To understand why this is the case, consider several cross-sections of a synthetic 3D image of a branching tube, shown in fig. 5.3. In the first cross-section, branching has not begun, so the cross-section is approximately circular and is fit well by a medial atom. In the second cross-section, the two branches have begun to separate. The fit of the medial atom is poorer than in the first cross-section as the top and bottom involutes lie slightly inside the object boundary, while the atoms on the left and right

sides lie outside of the object boundary. In the third cross-section, the branches have grown further apart, and the medial atom fit is worse yet. In the final cross-section, the medial atom is still centered between the two branches, but the poor fit has begun to cause instability, indicated by the slanting of the two branch cross-sections because the medial atom has rotated out of the desired plane. As the core continues past the branching region, the medial response becomes weaker and the medial fit becomes increasingly unstable.

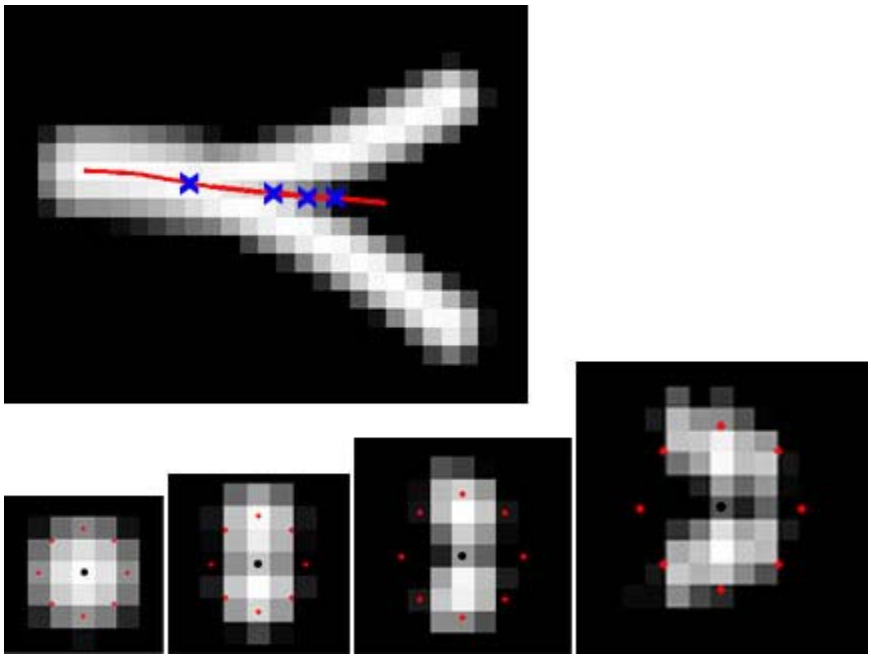

Fig. 5.3. Top: A slice through a nearly symmetric 3D branching tube, with the tube's computed core shown in red. Bottom: Cross-sectional slices of the tube with the core location indicated by a black dot and the 8 involute locations indicated by red dots. The locations of the four cross-sections along the core are indicated in the top image by blue x's, in order from left to right.

In most 3D tubular branches, as in fig. 5.4, the core begins bisecting the two new branches but then jumps to one of the branches. The core-following algorithm of section 3.4 takes a step forward in the core tangent direction to predict a new medial atom, then the medialness of the predicted atom is optimized using conjugate gradient. The optimization process finds a stronger medialness response when the atom is placed along one of the branches rather than when it is centered between the two, causing the atom to jump. This

occurrence is not a kink in the core but a jump from one core to another. Even the leftmost example of fig. 5.4 has a small jump, but the branching in this case is so asymmetric that the jump is not visually evident.

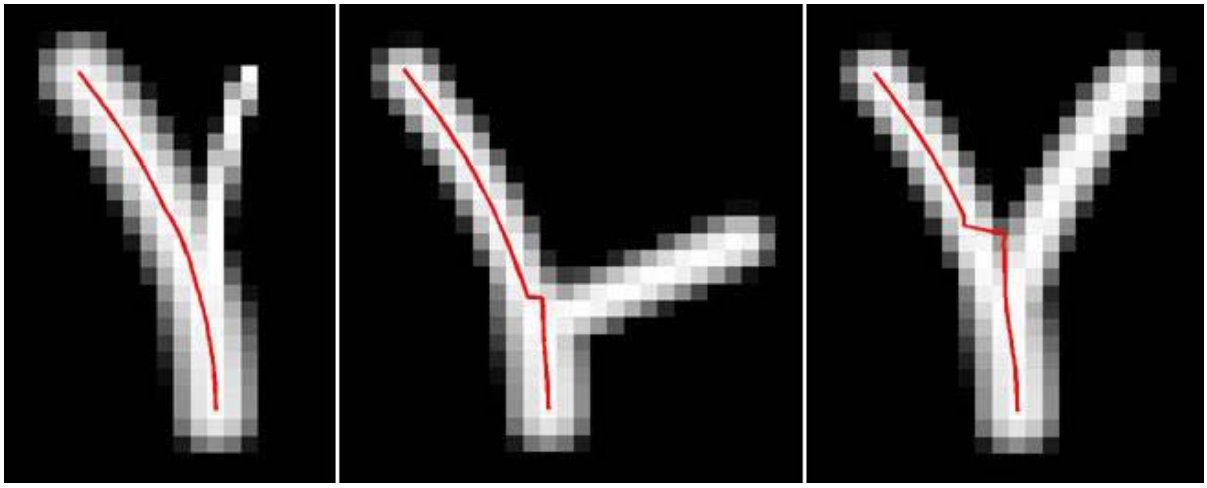

Fig. 5.4. Slices through three different 3D branching tubes, with the tubes' computed cores shown in red. Enough asymmetry exists in each of these cases to cause the core to jump to one of the two branches.

Jumping occurs for several reasons:

- If one branch is wider than the other, the core-following algorithm generally jumps to the wider of the two branches. Before the jump, more medial involutes lie near the boundary of the wide branch than lie near the boundary of the narrow branch, favoring the wide branch. The leftmost image in fig. 5.4 is an example of this situation.

- If the orientation of one branch continues more smoothly from the trunk, the core-following algorithm generally jumps to that branch. At each step of core-marching a new predicted atom is initialized with the same orientation as the atom from which it was predicted, causing the new atom to favor an object with a similar orientation. The middle image in fig. 5.4 is an example of this situation.

- If one branch is brighter than the other, the core-following algorithm generally jumps to the brighter of the two branches. This occurs because medialness increases with object brightness.

The rightmost image in fig. 5.4 is the most symmetric of the three examples; consequently, the core continued bisecting the two branches longer than in either of the other cases. Still, when the medialness response became too weak, the optimization process favored the left branch.

In asymmetric cases, the medialness response of one of the 8 medial atom spokes decreases significantly more than the other spokes as the core passes through the branching region [Aylward02] due to the radial asymmetry created by the branching, as shown in fig. 5.5. In symmetric cases, certain spoke medialness responses also decrease in the branching region, but normally there are at least two such spokes on opposing sides of the computed medial atoms (fig. 5.3). In asymmetric cases, the radius of a core briefly increases while the core bisects the two new branches but then decreases sharply when the core jumps to one of the two branches. In symmetric cases, the radius of a core steadily increases as it passes through a symmetric branch (fig. 5.3). A steady increase in radius also occurs at an aneurysm but does not commonly happen anywhere else along a healthy blood vessel [Aylward96]. The branch-handling method presented in this dissertation exploits these properties of cores at symmetric and asymmetric branches of tubes, as described in subsections 5.4.2 and 5.5.

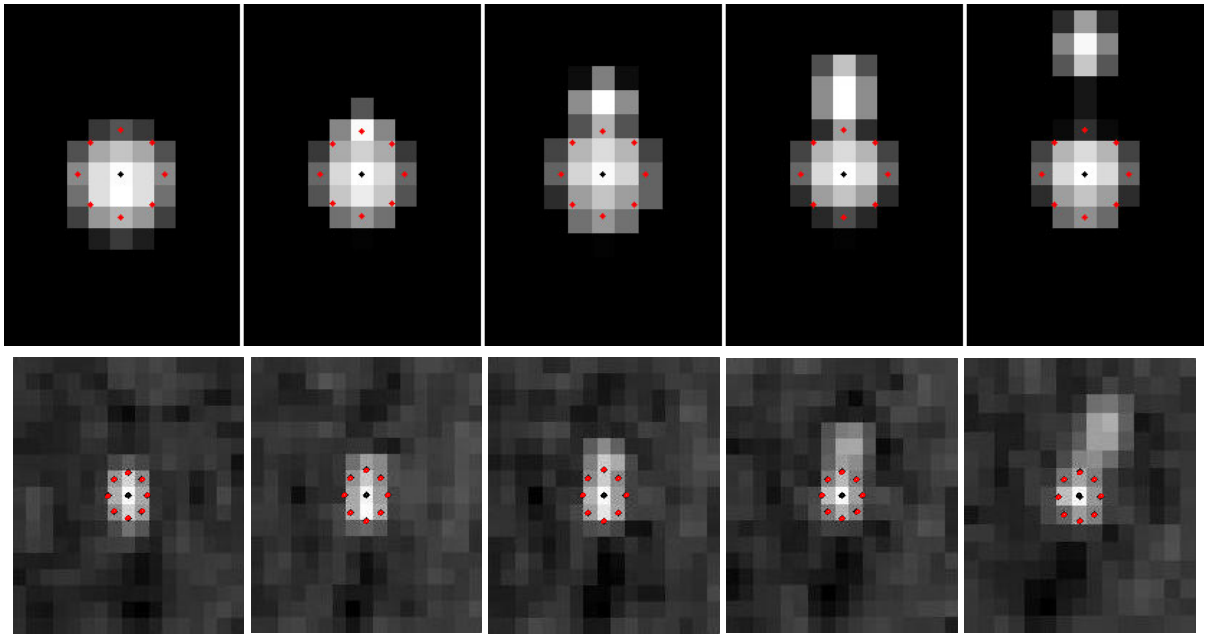

Fig. 5.5. Top: Cross-sectional slices of a synthetic branching tube with the computed core location indicated by a black dot and the 8 involute locations indicated by red dots. Bottom: Similar results showing the core of a branching blood vessel in volume MRA data of the head. In both image sequences the medial atom spoke pointing directly upwards gives a weak medialness response as the core passes through the branching region.

5.3 Properties of Cores at Slab Branches

In this dissertation I classify slab branches in three categories that display different properties: crest branches, abutting objects, and face branches. A crest branch occurs when a child figure's core is connected to end atoms of the parent figure's core. A finger is an example of such a crest branch, as its core is connected to an end atom of the palm's core. The core of a crest branch generally continues smoothly from the core of the parent figure at least in position and radius if not in orientation. Since the core of a crest branch continues smoothly from the core of its parent figure, the core-following algorithm computes the two as part of the same core, so this type of branch will not be discussed further in this section.

Abutting objects occur when two separate objects appear to be a parent and a child figure of the same objects. Such a branch might occur in a CT image when the kidney and liver have indistinguishable intensities, causing them to appear as figures of a single object

with no defined border. In this type of branching the child figure is a separate object, not computed by the core-following algorithm.

A face branch occurs when the core of a child figure is connected to the core of the main figure anywhere but at an end atom. Such a branch might appear as a finger protruding from the palm of a hand or a nose protruding from a face. The core of a face branch does not continue from the core of the parent figure. Consequently, what I will call the principal core of the sub-figure must be seeded and then followed separately from the parent figure. To automate this process, the sub-figure must be detected during core-following of the parent figure. In order to understand the problem of face branch detection I first describe the structure of the cores involved.

The behavior of core-following at face branches in 3D bears similarities to the behavior of cores at branches of tubular objects in 2D, presented at the beginning of section 5.2, since the medial atoms used in all three situations have two spokes. In particular, as a slab-like core passes through a branching region in 3D, the two medial atom spokes follow the outer boundaries of the branching object and the core bisects the two branches. Since the spokes follow the object's outer boundaries, the core's radius increases through the branching region. While medialness responses may decrease due to the deviation of opposing object boundaries from parallel, neither individual response is expected to drop significantly more than the other, unlike the case of a tubular object in 3D, where a single spoke's medialness response drops significantly in the vicinity of a branch.

Slab-like cores also differ from 3D tubular cores in that a branching slab is usually represented by a parent and one child (fig. 2.3), whereas a branching tube is represented by 3 cores – a parent and two children. The parent figure in a branching slab-like object usually encircles the base of the child figure allowing for the parent core to continue marching even after the branching region has been encountered.

Finally, slab-like cores differ from 3D tubular cores in the relative locations of cores of individual figures at branches. Cores of slab-like figures intersect along a curve in Euclidean space, whereas cores of tubular figures generically do not intersect each other in

Euclidean space. Cores of both slab-like and tubular figures generically do not intersect in parameter space.

5.3.1 Transitions at Slab Branches

Section 5.3 described the behavior of slab-like cores at object branches. This behavior, in which the core curves so as to bisect the main figure and the sub-figure, can be seen when following the core of the main figure shown in fig. 5.6 along the path indicated by the blue arrow. When following the core along the path indicated by the red arrow, however, the sub-figure is not encountered, and the core continues following the main figure. This raises the question: What happens between these two situations, for example, when following the core along the path indicated by the purple arrow?

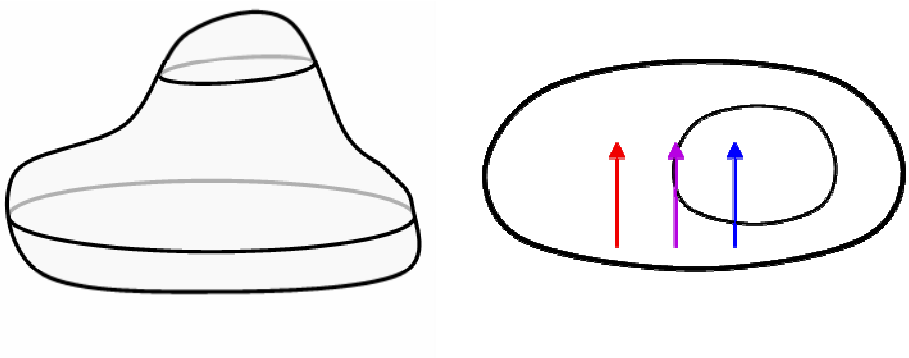

Fig. 5.6. Left: A representation of a branching slab-like object. Right: A projection view of the same object from above, with the sub-figure shown as a darker blob. When the slab-like core of the main figure is followed along the path of the red arrow it is unaffected by the sub-figure. When the core is followed along the path of the blue arrow it bisects the main figure and the sub-figure. But what happens to the core as it is followed along the path of the purple arrow?

To understand the answer to this question, consider the Smoothed Local Symmetries (SLS), introduced in section 2.2.1, which was originally developed as a 2D variant of Blum's MAT. The relationship between the SLS and the MAT parallels the relationship between the cores of this dissertation, which use medial atoms with a fixed object angle, and the cores of others, which use medial atoms with a variable object angle. The SLS is a better analog to

the cores described here, since the "spokes" of a SLS location are collinear (fig. 2.4). On the other hand, the MAT "spokes" are perpendicular to the object boundaries at the locations of tangency and are consequently non-collinear.

Dr. James Damon considered the extension of the SLS to 3D to gain a better understanding of the behavior of cores in the vicinity of branches. In the diagrams in fig. 5.7, consider following the SLS along the path indicated by the leftmost red line segment; since no sub-figure is encountered in this situation, the SLS is relatively flat. When following the SLS along paths indicated by parallel line segments further to the right, the sub-figure begins to have an effect, causing the SLS of the main figure to bow in the direction of the sub-figure. Along one of these parallel paths, somewhere near the edge of the sub-figure, a discontinuity appears along the SLS of the main figure. At the location of this discontinuity, a transition occurs in which the SLS separates into two sheets, and the sheets begin to push through each other. Moving further into the region of influence of the sub-figure along the parallel paths, the two separate SLS sheets continue to lengthen until they each reach a maximum in length somewhere near the middle of the sub-figure and then shorten again. Another skeletal sheet, corresponding to the medial surface of the sub-figure, exists somewhere above the curves on the right side of fig. 5.7.

According to the predictions of Damon, this behavior of the SLS is an approximation of the behavior of slab-like cores at branching locations. Since cores, unlike the SLS, are computed at scale, their behavior should be less consistent than that of the SLS. I attempted to experimentally determine the precise locations of discontinuities similar to the one shown in fig. 5.7 by applying the slab-like core-following algorithm to images of branching slabs. These attempts were unsuccessful due to the instability of cores and their reliance on the chosen marching direction. Less precise experiments seemed to support the prediction that slab-like cores behave in a similar manner to the 3D SLS, as core-following produced shapes qualitatively like those in the right side of fig. 5.7. For example, when passing near a branch, cores bowed in the direction of the sub-figure, regardless of the direction of core-following. On the other hand, when approaching the central region of a slab branch from opposite directions, two separate cores were produced, each bisecting the main figure and the sub-

figure. These mathematical and experimental results are incomplete and require further study. Fortunately, the method for branch-detection, presented next, terminates the main figure's core and seeds the sub-figure's principal core before any discontinuities are encountered.

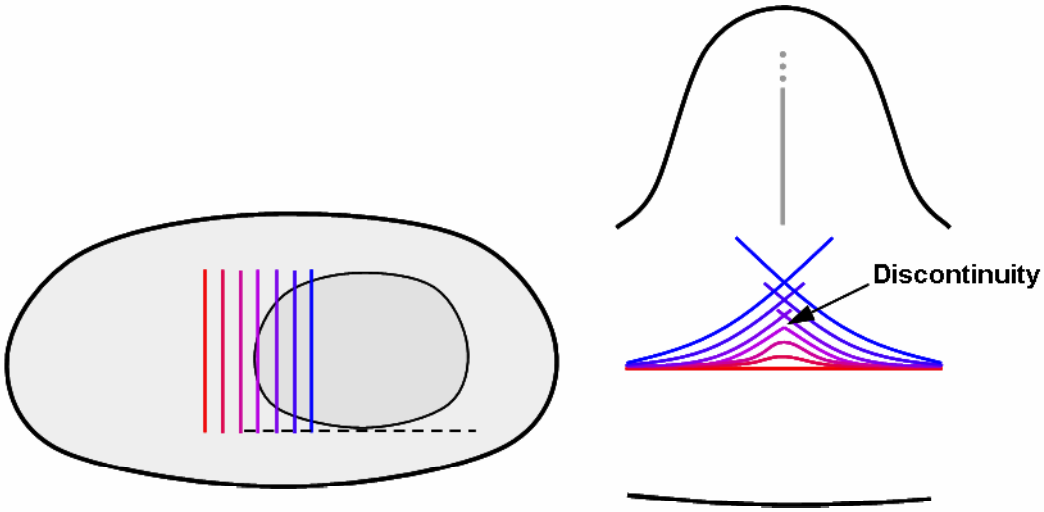

Fig. 5.7. Left: A projection of the slab-like object of fig. 5.6 from above. The colored line segments indicate 7 particular cuts of the SLS shown at right, along with part of the profile of the object (black) and the skeleton of the sub-figure (gray). The cut shown in red is far enough from the sub-figure to be unaffected by it. As the cuts come nearer to the sub-figure they bow, and eventually a discontinuity forms. As the cuts enter the region of influence of the sub-figure, two separate sheets form and lengthen (blue). Similar behavior occurs along cuts that approach the sub-figure from other directions, such as the one indicated by the dotted line segment in the left side of this fig.

5.4 Branch-Detection

Since neither tubular cores nor slab-like cores bifurcate, core-following alone cannot extract branching objects from images. An explicit branch-handling method is required to traverse branching regions and continue core-following at the new branches. The branch-handling method presented in this dissertation consists of two parts. First, the method detects when the core being followed reaches a branching region. Second, the method locates the new branches and resumes marching along the core of each new branch.

Branch-detection can be further divided into two components, described in the next two subsections. Initially, a cornerness operator is applied to find potential branching locations. Next, a set of heuristics is used to reject false positives.

5.4.1 The Cornerness Operator

Most locations on surfaces of anatomic objects have principal curvatures that are positive or near 0; such locations are either convex or slightly hyperbolic. Near bifurcations of both tubular and slab-like objects, at least one of the two principal curvatures typically becomes significantly negative due to a more pronounced hyperbolic region or even a concavity. Accordingly, branch-detection techniques can utilize surface curvature as an indication of branching. In the method presented here, both hyperbolic and concave regions are detected using a cornerness operator that responds to curved image edges (fig. 5.8).

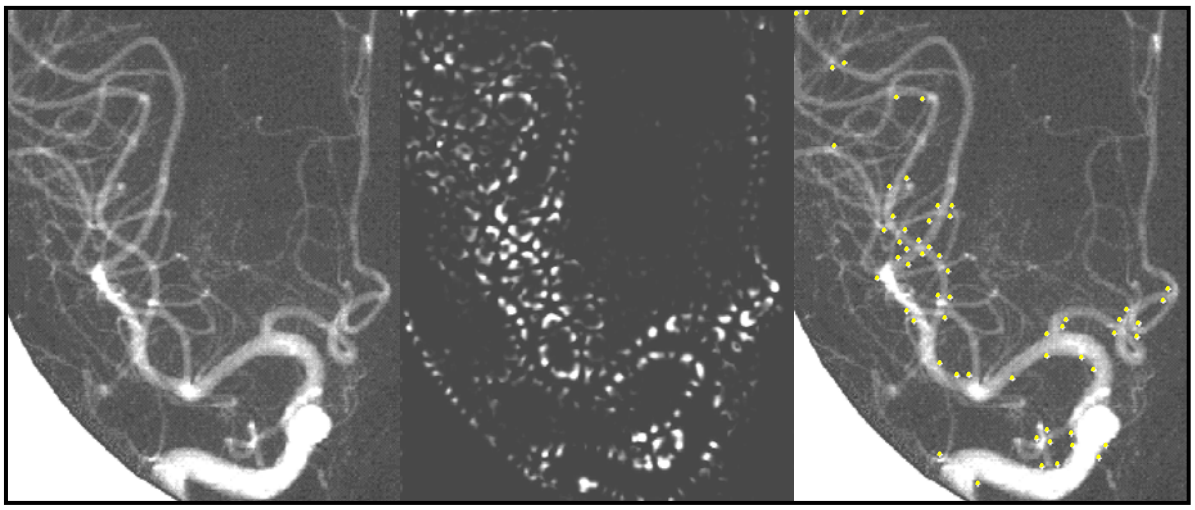

Fig. 5.8. Left: A 2D digital subtraction angiogram (DSA) projection image of the head. Center: The cornerness operator $I_{\vec{u}\vec{u}} I_{\vec{v}}^2$ applied to the DSA image at a scale of $\sigma = 2$ pixels. Right: Local maxima of cornerness above an arbitrarily chosen threshold, displayed on the original DSA image. This example is shown in 2D to facilitate visualization; similar results are achieved in 3D.

The 3D cornerness operator used in this dissertation is given by equation 2.5, $C = I_{\vec{u}\vec{u}} I_{\vec{v}}^2$. This operator computes the squared first derivative of the image in the gradient

direction, $\vec{v}$, thus responding strongly to image edges, and multiplies it by the second derivative of the image in an orthogonal direction $\vec{u}$ (fig. 5.8). The vector $\vec{u}$ is defined as the eigenvector corresponding to the largest eigenvalue of the Hessian matrix in the plane normal to $\vec{v}$. Since the second derivative of $\vec{u}$ gives the curvature of level curves in an image [Lindeberg98], this operator favors curved image edges.

This cornerness operator is applied locally to the image, during each step of core-following, at the same scale used to compute medialness (see section 6.2). Its response is then probed at each of the current medial atom's involute locations. Since the cornerness operator gives positive responses at hyperbolic and concave corners of branches, the response at one or more of the involute locations generally shows a spike as the core passes through a branching region. The cornerness response at each involute is tracked individually during core-following, and if an atom has at least one involute that shows a significant spike in cornerness, that atom is flagged as being at a potential branching location. The choice of what is considered a significant spike is heuristic; in this dissertation I flag an atom if it meets three conditions:

1. At least one of the atom's involutes is at a local maximum in its cornerness response compared with the same involute of neighboring atoms.

2a. For a tubular object, the cornerness response at that involute is at least double the most recent local minimum of cornerness of the same involute along the core.

2b. For a slab-like object, the cornerness response at that involute is at least double the nearest local minimum of cornerness of the same involute along the series of medial atom neighbors in the direction away from the branch, chosen as the direction from which the current medial atom was predicted (see section 4.3 and fig. 4.1).

3. The sum of the cornerness responses over all involutes of the atom is a local maximum.

The first condition is intuitive, as discussed in the previous paragraph. The second condition is used to eliminate insignificant local maxima caused by noise. The choice to require an increase by a factor of 2 is heuristic; a relatively small value was chosen to avoid missing desired branches. This choice errs on the side of allowing false positives, but those are dealt with later, as explained in the next subsection. Using only these first two conditions, a medial atom would be flagged when any of its involutes returned a positive response and the corresponding involute on each neighboring atom returned a negative response, regardless of how small in magnitude the positive response is. The third condition requires that the local maximum of the chosen involute be significant enough to counter the predominantly uncorrelated cornerness responses of the other involutes.

This method effectively detects branches of both tubes and slabs, as shown in Chapter 7, but it incorrectly identifies branches in certain situations, most often at locations where an object bends sharply. The method also finds false positive branches in cases of voxel jaggedness due to noise or aliasing, particularly at narrow locations in an object where a small scale of interrogation is used. Such false positives are identified and discarded, as described next.

5.4.2 Rejecting False Positives

False positive branches are identified via a set of heuristics motivated in sections 5.2 and 5.3 that differ depending on whether the object being extracted is slab-like or tubular. When a tubular branch is examined and determined to be a true positive, the heuristics are further used to classify it as either a symmetric branch or an asymmetric branch. When an object is composed of a combination of slab-like and tubular figures, the parent figure at a bifurcation dictates which heuristics are used. This situation is discussed further in section 5.7, which combines slabs and tubes. The measures and multiplication factors presented next for identification of false positive branches were determined by statistical pattern analysis on a variety of objects in both synthetic and clinical images. Furthermore, none of these measures were changed during the validation presented in Chapter 7, regardless of the attributes of the images and/or objects being analyzed.

When extracting branching slab-like objects, the only value used in identifying false positives is the core's radius. As described earlier, the radius of a slab-like core generally increases as the core passes through a branching region. Hence, when a potential branch is detected, the radius of the current medial atom, call it r_0, is noted. Core-following then continues forward for a distance $3 * r_0$. If the radius of the core does not surpass a value of $1.75 * r_0$ anywhere within that distance, the potential branch is regarded as a false positive.

When extracting branching tubular objects, three heuristics are considered. First, the relative medialness responses of the individual atom spokes are examined. Let dM_m be the difference in values between the strongest and the weakest medialness response among the individual spokes of a medial atom $\underline{m}$. Let dM_0 be that same difference in values for the atom at which a potential branch was detected. Let r_0 be the radius of the same atom. If dM_0 is less than 1.5 times the average of dM_m over all atoms located between the current atom (exclusive) and a distance $10 * r_0$ backwards along the core, the potential branch is regarded as a false positive. This measure tests to see if there exists a weakly responding spoke on the potential branch medial atom relative to previous atoms. Another option would have been to simply consider the value dM_0 without comparing it to the values at previous atoms, but dM_0 is also high whenever the object of interest deviates from tubularity, regardless of whether there is a branch.

A second heuristic checks for an increase in the core's radius in a similar way as described for slab-like cores two paragraphs ago. If it finds a significant increase in radius, the current atom is accepted as being at a branching location. Moreover, that branching is categorized as a symmetric one, since only symmetric tubular branches display a significant increase in core radius, as described in subsection 5.2.1.

A third heuristic tests for an "up-down-up" pattern in cornerness along the core direction to recognize an asymmetric branch. The first "up," or spike, in cornerness is already identified in the branch-detection step (condition 3 of subsection 5.4.1). As the core continues marching in an asymmetric branching situation, cornerness decreases and often

becomes negative (fig. 5.9). Finally, as the core exits the branching region along one of the two branches, cornerness once again achieves a local maximum due to the angle between the two branches. This "up-down-up" pattern is tested as follows. Let C_0 and r_0 be the cornerness and the radius, respectively, of the atom at the potential branching location (location 1 in fig. 5.9). The method probes forward a distance $3 * r_0$ along the core for a local minimum in cornerness that is at most $0.25 * C_0$ (location 2 in fig. 5.9); call this minimum cornerness value $C_{\min}$. If such a minimum is found, the method probes forward a distance $3 * r_0$ from the location of the minimum, searching for a local maximum in cornerness that is at least $3 * C_{\min}$ (location 3 in fig. 5.9). If this second maximum is found, the medial atom at location 1 is accepted as an asymmetric branching location; otherwise it is rejected as a false positive. These values were determined using statistical pattern analysis and were not changed during validation.

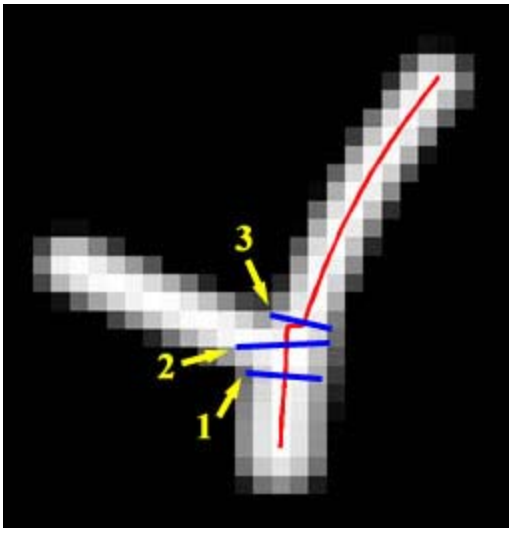

Fig. 5.9. A slice through a 3D branching tube, with the tube's computed core shown in red. The blue lines represent the planes of three different medial atoms. At location 1 there is a spike in cornerness due to the angle created between the trunk and the left branch. At location 2 there is a dip in cornerness since the involute lies somewhat inside the object. At location 3 there is another spike due to the angle between the two branches.

The choice of whether to extract tubular cores or slab-like cores is one that must be made explicitly by the user. When extracting tubular cores, however, the categorization of whether a branch is symmetric or asymmetric is made automatically, as described, for use when continuing following the cores of the two new branches. This continuation of core-following, the next step after branch-detection, is discussed in the next two sections.

5.5 Continuation of Core-Following for Tubes

Once a tubular object bifurcation is detected, the branch-handling method terminates the current core at the medial atom where the bifurcation was detected, $\underline{m}_0$, and attempts to locate the cores of the two new branches. Each of these two cores is then extracted using the marching method of section 3.4. The technique for locating the cores is different for symmetric and asymmetric bifurcations.

During experimentation I have applied this method to several thousand branches of blood vessels and not found a single symmetric branch; symmetric tubular bifurcations are extremely unlikely in clinical images. Blood vessels are unlikely to be symmetric about a bifurcation, and even when they are nearly symmetric, image noise can cause a core to favor one branch over the other. On the other hand, trials have shown that image noise has less influence on cores when the tubular objects being tracked are tens of pixels rather than a few pixels in diameter. Consequently, symmetric branching behavior is possible when extracting objects wider than blood vessels. For the sake of completeness, I present the method for continued following of cores of symmetric branches.

When a symmetric bifurcation is detected, the starting locations of the two new cores are estimated. First, I define the branching region within which $\underline{m}_0$ resides to start where the trunk of the bifurcation begins widening and to end as soon as the two resulting branches no longer overlap. I have found that this branching region is unstable for branch prediction, so the method must first exit the branching region. To do this, a step of size r_0 is taken forward along the current core from $\underline{x}_0$, where r_0 and $\underline{x}_0$ are the radius and location, respectively, of $\underline{m}_0$. At the new location, the method described in section 3.4 computes a new medial atom

$\underline{m}_1$ by predicting and then optimizing atom parameters. A step of size r_0 provides a location near the end of the branching region that has given good results in qualitative experiments using synthetic images similar to the one shown in fig. 5.10. No quantitative testing has been performed to determine the effectiveness of this step size. After taking a step, the tips of two opposing spokes of the new atom $\underline{m}_1$ will usually lie along the edges of the object, and these two spokes will have higher responses in medialness than the rest of the spokes (right side of fig. 5.10). Since these two spokes typically pass through the two new branches and end at the outer edges of the branches, the starting locations of the two new cores are estimated at a small distance d in toward $\underline{x}_1$ from the two specified spoke tips. The particular choice of d is described in the next paragraph. Atom $\underline{m}_1$ is used only for the purpose of estimating the new cores and is not saved as part of the current core.

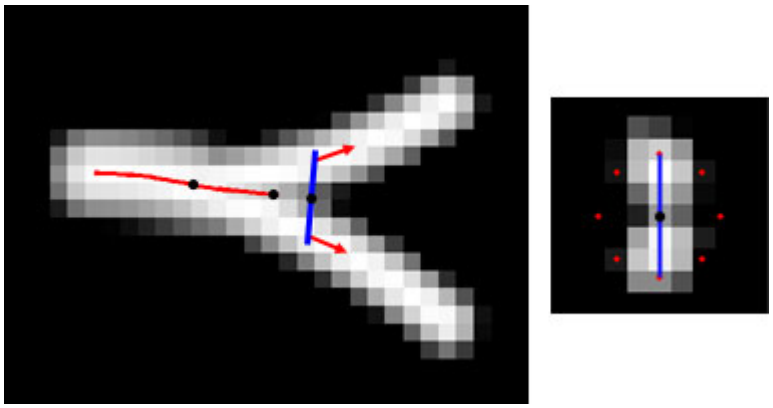

Fig. 5.10. Left: A slice through a 3D branching tube, with the tube's computed core shown in red. The black dot at the right end of the core is the location of the last medial atom along the core, $\underline{m}_0$. The black dot in the middle of the core is the location of the medial atom $\underline{m}_{-1}$ used to estimate the radii of the new cores. The final black dot and the attached blue line represent the location and spokes, respectively, of another medial atom, $\underline{m}_1$, from which the locations and orientations of the new cores are estimated. The two red arrows indicate the desired starting locations and marching directions of the new cores. Right: A cross-sectional slice of the tube at the location of $\underline{m}_1$, with the core location indicated by a black dot and the 8 involute locations indicated by red dots. The blue line represents the two spokes of $\underline{m}_1$ with the strongest medialness response.

Using these two starting locations for the two new cores, the method must estimate the starting radii and orientations. The starting radii of both cores are estimated as $\frac{1}{\sqrt{2}} * r_{-1}$, where r_{-1} is the radius of medial atom $\underline{m}_{-1}$ located a distance $3 * r_0$ backwards along the core from $\underline{m}_0$. The method uses r_{-1} rather than r_0 as a measure of the radius of the tube before branching since the tube is already dividing at the location of $\underline{m}_0$. The factor of $\frac{1}{\sqrt{2}}$ is chosen so the cross-sectional areas of the tubes sum to the cross-sectional area of the tube from which they branched, a property of normal blood vessels. I use this new radius to set the distance from the spoke tips, $d = \frac{1}{\sqrt{2}} * r_{-1}$.

The starting azimuth and altitude of each of the two new cores are estimated as the azimuth and altitude of the vector from $\underline{x}_0$ to the estimated starting location of the new core. These estimates approximate the direction in which each of the two branches extends from the original tube. The spin angle φ of each of the new cores is initialized as the spin angle of the atom $\underline{m}_0$. As described in section 3.4, the initial value of φ is not critical.

When a tubular branch is asymmetric, the core-following algorithm jumps from the core of the trunk to the core of one of the two branches, as described in subsection 5.2.1. Hence, only one of the two branches remains to be located, as follows. One of the spokes of the medial atom $\underline{m}_0$ at which the branch was detected will have a particularly weak response because its involute is located within the new branch rather than along an object edge. For example, in both rows of fig. 5.5 a branch was detected in the second image from the left, and in both cases, the top spoke responds weakly. Since that involute is located within the new branch, its location is used as the estimated starting location for the new core.

The initial radius of the new core is estimated as $\frac{1}{2} * r_{-1}$. The smaller factor than that used in the symmetric case reflects the assumption that when the branching is asymmetric, the branch not followed by the core is the smaller of the two branches. The particular choice

of $\frac{1}{2}$ as a multiplication factor is heuristic, but it is effective, as described in the next paragraph. The azimuth, altitude, and spin angle defining the initial frame of the new core are taken from $\underline{m}_0$ as in the symmetric case. The kink in the core that continued from the trunk to one of the branches is not fixed but could be as a subject of future work.

The estimated medial atom or atoms are then refined using the corrector portion of the method presented in section 3.4, whether an asymmetric branching is detected and one new medial atom is estimated or a symmetric branching is detected and two new medial atoms are estimated. In other words, the estimated locations and parameters of the new atoms are corrected using the combined Euclidean and parameter space optimization process. This optimization is robust enough that the estimates described above are sufficient. As subsections 7.2.1 and 7.3.1.2 show, this method for the continuation of core-following is more effective than the branch-detection method upon which it relies.

Once the new medial atoms are corrected, core-following resumes at each of the two new branches. When false positive branches are not rejected by the method discussed in subsection 5.4.2, they are taken care of here. If the new core being tracked ends within 5 steps of marching, as determined by the core-termination method described in Chapter 6, it is rejected as a false positive. A minimum of 5 steps are needed by the core-termination method, and experiments have shown that the core-following method terminates within 5 steps if a true branch does not exist (see subsection 7.3.1.3).

5.6 Continuation of Core-Following for Slabs

When extracting branching slab-like objects using cores, the branch-detection method presented in section 5.4 finds multiple branch atoms on the parent figure's core (fig. 5.11). This differs from the tubular case, in which a single atom on the parent figure's core is identified as a branch atom. Another difference between branching tubular objects and branching slab-like objects is that a tubular object has two child figures at a branching, whereas a slab-like object typically only has one child figure (fig. 5.11). As discussed in section 5.3, this child figure can be categorized as a crest branch, an abutting object, or a face

branch. When the child figure is a crest branch, core-following continues on to the child figure from the branch figure, so branch-handling is not required. When the child figure is either an abutting object or a face branch, however, the branch-detection method identifies a set of medial atoms on the parent figure's core as branch atoms, and, as mentioned in section 5.3, finds atoms seeding the child figure's principal core.

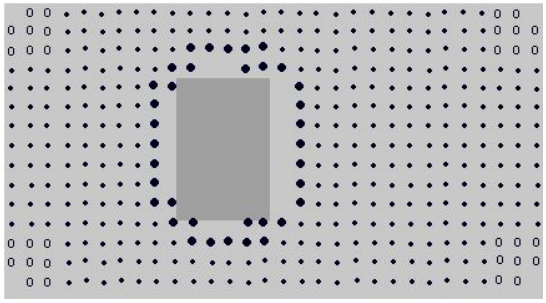

Fig. 5.11. One slice of a synthetic image of a rectangular slab (*light gray*). The dark gray area represents a rectangular child figure that protrudes from the parent figure toward the viewer. The dots show the computed medial atoms of the parent figure and their associated cornerness values. Open dots represent negative cornerness values, small dots represent both positive and negative cornerness values near zero, and large dots represent high cornerness values (above a chosen threshold). All medial atoms indicated by large dots were flagged as branch points, leaving a hole in the parent figure's core. The lack of image noise in this example resulted in such a regular grid of medial atoms.

When medial atoms along a slab-like core are identified as branch atoms according to the branch-detection method, they are flagged. No new atoms are predicted, but core-following continues from other atoms until extraction of the parent figure's core terminates. Then one of the flagged atoms, call it $\underline{m}_0$, is chosen, and a child figure atom is predicted from $\underline{m}_0$'s geometry as follows. This process will be repeated for each of the flagged atoms as described below, and the order in which flagged atoms are chosen is not specified.

First, the method determines how many neighboring atoms $\underline{m}_0$ has in the parent figure's core, where an atom's neighbors are defined as in section 4.3 (see fig. 4.1). Many of the atoms at branching locations will have 3 neighbors (see fig. 5.11), in which case the location of the child figure is predicted in the remaining compass direction. Let $\vec{f}$ represent

this forward direction in which the child is predicted. If $\underline{m}_0$ has 4 neighbors, it is disregarded as a branch atom, since at least one of its neighbors is likely to point to the same child figure. If $\underline{m}_0$ has 2 neighbors, the direction in which the child figure is predicted, $\vec{f}$, is found by averaging the remaining 2 compass directions; if it has 1 neighbor, $\vec{f}$ is chosen as the middle of the 3 remaining compass directions. Due to the computation of atoms using the specified marching method, an atom cannot have 0 neighbors.

Next, a step is taken from $\underline{m}_0$ in the direction $\vec{f}$, and a new atom $\underline{m}_1$ is computed. The parameters of an atom $\underline{m}_{-1}$ located one step backwards from $\underline{m}_0$ are applied at the location of $\underline{m}_1$, thereby defining two new spokes (fig. 5.12). One of these two spokes is chosen based on the direction of curvature of the core from $\underline{m}_{-1}$ to $\underline{m}_1$, which is likely caused by the child figure of interest, and the starting location of the child figure's core is estimated as the tip of this chosen spoke.

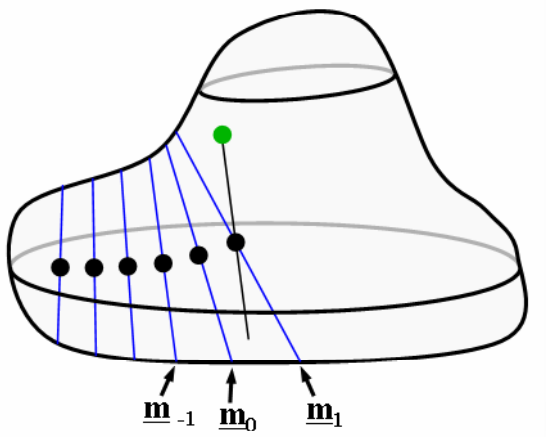

Fig. 5.12. A representation of a branching slab-like object. Several atoms of the parent figure's core are displayed as black dots with blue spokes. $\underline{m}_0$ is the atom at which a potential branch was detected. The pair of black spokes shown in the fig. are computed by applying the parameters of atom $\underline{m}_{-1}$ at the location of atom $\underline{m}_1$. The green dot shows the estimated starting location of the child figure's core.

The initial forward direction of the new core, $\vec{b}$, is estimated as the direction of the spoke chosen in the previous paragraph. This estimate is based on the assumption that the child figure branches in a direction roughly normal to the surface of the parent figure. This estimate is sufficient, as discussed below.

The direction of the spokes within the plane normal to $\vec{b}$ and the radius of the new estimated atom are unclear as there is no defined relationship between the parent figure and the child figure. Several possible geometries are explored. In particular, two orthogonal spoke directions are considered – one in the North-South direction of atom $\underline{m}_{-1}$ and one in its East-West direction (see section 4.3). Four different radii are also considered at equal increments from 1 voxel to the radius of atom $\underline{m}_{-1}$. The choice to use these four radii was arrived at through experimentation, as one of the four was always enough to find the desired child figure using the test images of subsection 7.3.2.2. The two spoke orientations and the four radii are combined in all eight possible permutations. Each permutation, along with the location and the $\vec{b}$ vector defined above, is used as an initialization to determine a potential medial atom via the optimization process defined in section 4.3. Of these eight resulting atoms, the one with the strongest medialness response is chosen as the correct starting atom for the new core.

The core-following method then extracts the core of the child figure starting from the chosen medial atom. If the core-following method terminates before finding five medial atoms (see Chapter 6), the child figure is discarded as a false positive. Once core-following of the child figure is complete, another flagged branch atom from the parent figure is arbitrarily chosen and a new child figure atom is predicted. If this predicted atom falls within the volume of an existing figure, as defined by the figure's core, the atom is discarded. If the predicted atom does not fall within the volume of an existing figure but core-following from this predicted atom leads into an existing figure, the new core is discarded. This is necessary to avoid duplicate figures since multiple branch atoms point to a single figure. This process continues until all flagged branch atoms on the parent figure have been considered.

The estimated starting locations and parameters of cores of slab-like sub-figures are not as accurate as when extracting tubular objects. In many cases, the core predicted from a given branch atom terminates within 5 steps of marching and is discarded due to a lack of image information. In particular, 63% of the predicted sub-figure cores in the experiment of subsection 7.3.2.2.1 were discarded. Fortunately, slab-like cores have multiple branch atoms, and the child figure's core is often found from at least one of the branch atoms. As subsections 7.2.3 and 7.3.2.2 show, this method effectively locates most abutting object branches and face branches. No automated method has been developed to differentiate between an abutting object branch and a face branch. Rather, an option is available whereby the user can label computed branches as abutting objects once core extraction is complete. Any branches labeled as such are not considered part of the desired object.

5.7 Combining Slabs and Tubes

An object may be composed of a combination of slab-like figures and tubular figures. For example, a kidney can be represented as a slab, but the ureter, renal artery, and renal vein attached to the kidney as tubes. Such a combination of figures can be considered as a single branching object and extracted using cores. Although a general method could be developed to seek either tubular or slab-like branches and automatically determine which type of branch is found, the method presented here requires the user to indicate beforehand which type of branch is being sought. When extracting a slab-like main figure with tubular branches, for example, the main figure is extracted as a slab and branch-handling is executed using the process defined for slabs. When core-following resumes at a predicted branch, however, the tubular core-following algorithm is employed. Results of this method are presented in subsection 7.2.3.

Given this method for branch-handling, the only component still needed for the extraction of general branching objects using cores is core-termination. A technique for determining when to stop core-following is presented in the next chapter.

Chapter 6. Core-Termination

6.1 Introduction

This chapter presents a method for automatically terminating the core-following algorithms described in Chapters 3 and 4, which would otherwise continue following the core endlessly. The core-following algorithm should terminate under either of two conditions: 1) the object being extracted comes to an explicit end, where the end of the object is capped; or 2) the object being extracted comes to an implicit end due to a decreased signal-to-noise ratio (fig. 6.1). The second condition can occur, for example, when following a blood vessel that gradually narrows to the point that it is indiscernible in the image. In either of these two cases, a method is required to terminate the core-following algorithm at the appropriate location. The core-following algorithm should also terminate when an image edge is encountered, but that situation is trivial.

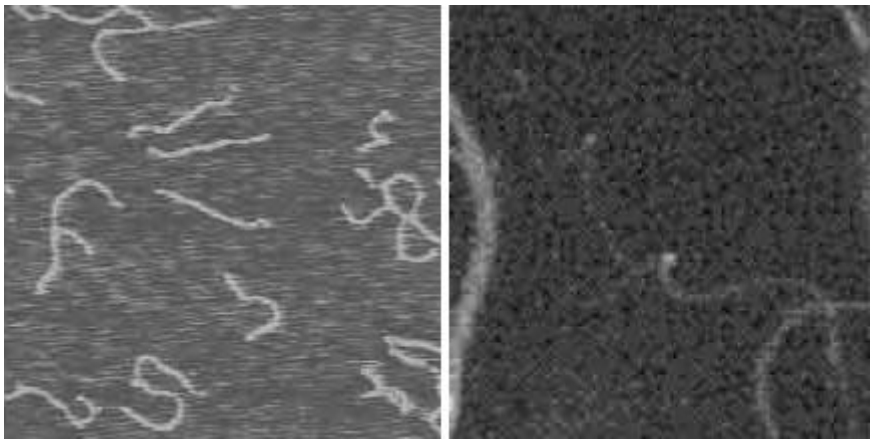

Fig. 6.1. Left: An atomic force microscope image of strands of DNA, examples of tubular objects with explicit (capped) ends. Right: Part of a 2D digital subtraction angiogram of the head showing a blood vessel that narrows as it spirals toward the upper left corner of the image. This blood vessel is an example of a tubular object with an implicit end.

Section 6.2 discusses detection and localization, two terms that refer to measures of success in core extraction. The ideas presented in this section developed from initial discussions with Dr. Mads Nielsen of the IT University of Copenhagen, Denmark. Section 6.3 presents properties of medial atoms that are randomly positioned and oriented within 3D images. Finally, section 6.4 integrates elements from the previous two sections into a method that automatically terminates cores of objects with both explicit and implicit ends.

6.2 Detection and Localization

A main objective of most segmentation techniques, including the one presented in this dissertation, is to locate objects as robustly and as accurately as possible. This objective becomes more difficult to achieve when an object narrows. When utilizing a scale-space segmentation method, the choice of scale plays a role in determining the success of the method on narrow objects. In this paper I choose scale, in voxels, as $\sigma = \max(0.25 * r, 1)$, where r is the radius of a medial atom and σ is the standard deviation of the Gaussian derivatives used to fit the medial atom to an image. The motivation for this choice is explained in this chapter.

Two scale-space concepts affect the extraction of narrow objects using cores – detection and localization. As the width of an object in a 3D image approaches the inner scale of the image, the signal-to-noise ratio of that object decreases. As the signal-to-noise ratio of an object decreases, the ability to track its core decreases. Specifically, the detection and localization of the object's core deteriorate. These two terms are defined using the example of a narrowing blood vessel in a 3D MRA.

Detection, as applied to the extraction of cores, is the percentage of attempts on which an object's core is found. As a blood vessel narrows it becomes dimmer, causing the medialness values computed along its core to decrease. When medialness values become sufficiently small, medial atoms can potentially respond more strongly to randomly aligned image noise than to an object. As a result, once a blood vessel becomes too narrow, the core that was following the blood vessel may begin marching in a seemingly random manner, responding to noise at each step. When the core no longer follows the object of interest,

detection of that object's core has failed. Detection is improved by increasing the scale of interrogation [Witkin83]. For core-following, this is done by increasing the standard deviation of the Gaussian derivatives used to compute medialness. This increases detection rates since Gaussians act as low-pass filters. In other words, when a Gaussian is used as an image filter, it blurs the image to which it is applied. Gaussian blurring removes high frequency information from the image, decreasing the influence of noise relative to the influence of structured image information, such as a blood vessel, on core extraction. The use of larger scale results in more blurring, so increasing the scale of interrogation decreases the effects of image noise, thereby improving detection rates.

Localization, as applied to the extraction of cores, is proportional to the distance between the computed core of an object and the medial axis of the unblurred object; the lower it is the better. In other words, localization is good when there is a minimal difference between the core at the computed scale and the core at zero scale in a noise-free image. Localization is improved by decreasing the scale of interrogation [Witkin83]. This is true because image edges, as detected by Gaussian derivatives, move in space depending on the scale of the Gaussians. For example, curved edges straighten under Gaussian blurring. Similarly, bright, narrow objects widen under Gaussian blurring. The greater the scale, the more pronounced these effects are. Thus, localization is optimal at 0 scale and deteriorates as scale increases.

Detection and localization are in conflict, as the former improves at larger scales while the latter improves at smaller scales. Traditionally the scale of interrogation used in computing cores has been width-proportional – as the radius of a core decreases the standard deviation of the Gaussians is decreased proportionally [Pizer98]. As mentioned in subsection 2.5.2, cores have often been extracted using medialness kernels composed of Gaussian derivatives with a standard deviation of 0.25 times the core's radius. This results in localization proportional to the width of the object of interest. The motivation for this is as follows. When analyzing an object with a radius of 50 voxels, for example, an error of two voxels in the location of the core may be insignificant relative to the advantages of a good

detection rate. When analyzing an object with a radius of one voxel, on the other hand, an error of two voxels results in a core that does not even lie within the object.

When using a width-proportional scale of interrogation to compute the core of a narrowing blood vessel, however, both the narrowing of the object and the corresponding decrease in scale affect detection negatively. Certain medical applications require the segmentation of blood vessels less than one voxel in diameter [Bullitt01]. The width-proportional medialness kernels described above results in the use of Gaussian derivatives with a standard deviation of less than one eighth of a voxel. Due to the discretization of both the image and the applied kernels, such small kernels are ineffective.

For most purposes, the use of width-proportional scale is preferred, as it provides width-proportional localization and effective detection. However, the standard deviations of the Gaussians used in core extraction in this dissertation are not allowed to decrease below 1 voxel, as the desired blurring properties of Gaussians deteriorate rapidly below this size.

Another option for the selection of scale, which could be used directly for core-termination, is as follows. At each step of the core-following algorithm, choose the smallest scale at which the object of interest can be detected, thereby minimizing the localization error. Using this method, as an object narrows, the scale of interrogation will increase to keep the detection rate constant. This results in increasing localization errors as the object narrows. When potential localization errors become greater than the radius of the object, the core-following algorithm should terminate, since the detected core is no longer guaranteed to lie inside the object [Nielsen02]. This method has not been implemented due to the difficulty of determining the smallest scale at which an object can be correctly detected, but it remains a potential area of future work.

The method used in this dissertation to determine when to terminate the core-following algorithm, presented in section 6.4, relies on knowledge of the distribution of medialness values of random medial atoms, discussed next.

6.3 Distribution of Random Atom Medialness Values

The core-termination method exploits the expected distribution of medialness values of atoms randomly positioned and oriented within a 3D image. Following is background for section 6.4.

In the core-termination method, a large number of medial atoms are randomly placed in a 3D image. Each of these medial atoms is randomly positioned in x, y, and z. Each has a random orientation, where the orientation is determined by randomly and independently choosing values for three variables: the azimuth of the atom's forward direction, $\vec{b}$, is chosen in the range $(-\pi, \pi]$; the altitude of $\vec{b}$ is chosen in the range $[-\pi/2, \pi/2]$; and the spin angle φ about $\vec{b}$ is chosen in the range $(-\pi, \pi]$. All atoms are assigned the same radius, as explained in section 6.4. Radius is not chosen randomly because doing so would produce results across scale space, and for core-termination we desire statistics produced at a given, local, scale.

Experiments have shown that the distribution of medialness values of randomly positioned and oriented atoms in medical images is unimodal and roughly symmetric about 0 (fig. 6.2). This is expected since the majority of random atoms are not aligned with an object; for medialness values computed by weighting an image by unaligned kernels composed of Gaussian first derivatives, values near 0 are expected to occur more frequently than values larger in magnitude. Furthermore, since a Gaussian first derivative is symmetric about the origin, any given medialness value M computed from a random atom is as likely to occur as $-M$. Consequently, the distribution of medialness values is expected to be symmetric about 0. The method for core-termination, defined next, computes the medialness values of the randomly placed atoms and then utilizes the expected properties of their distribution.

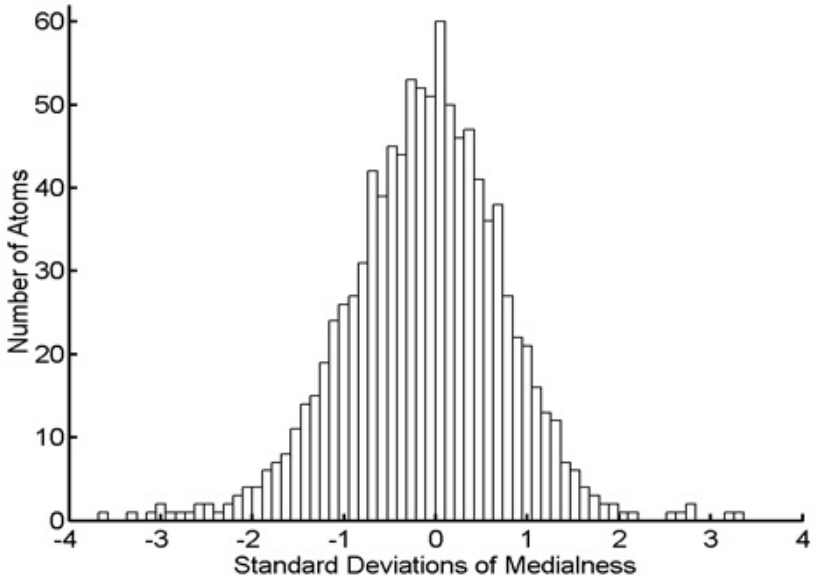

Fig. 6.2. The distribution of a set of 1000 medialness values randomly sampled from a region in an MRA, where the region is chosen as described in section 6.4. This histogram is partitioned into 75 buckets.

6.4 The Method for Core-Termination

The method used for deciding when to terminate the core-following algorithm is based on the confidence with which the desired core is detected. This confidence is determined by the strength of the medialness value of the core relative to local statistics on its expected medialness value. The following technique is implemented during each step of the core-following algorithm, regardless of whether the object being extracted is tubular or slab-like.

After the core-following algorithm predicts and corrects a new atom $\underline{m}_0$, it's medialness value, M_0, is recorded. Next, random atoms are placed in a spherical region of the image centered at $\underline{x}_0$ and having a radius $5 * r_0$, where $\underline{x}_0$ and r_0 are the location and radius, respectively, of $\underline{m}_0$. A radius of $5 * r_0$ provides a region large enough to make the likelihood that a random atom will be aligned with the object being tracked negligible yet small enough so that the image attributes are fairly homogeneous throughout the region. A

set of 60 randomly positioned and oriented atoms, all of radius r_0, are placed within this region, as described in section 6.3. The choice to use a single radius value for all the atoms was motivated in the previous section.

The medialness value of each of these atoms is computed, and the mean μ_M and standard deviation σ_M of this set of values are determined. Since these 60 medialness values are not optimized over Euclidean or parameter space, their computation is cheap relative to the medial atom optimization performed during each step of core-following, which normally requires several hundred medialness computations. As explained in section 6.3, μ_M is expected to be near 0. Assuming that the distribution of 60 samples is Gaussian, the computed mean will be within one quarter of a standard deviation of the expected mean roughly 97.5% of the time. When this computed mean is not within one quarter of a standard deviation of 0, the image is resampled using twice as many medial atoms in an attempt to obtain more reliable statistics. This process of resampling using twice the number of atoms is repeated until statistics with the desired mean are achieved. Resampling is rarely required in practice.

Using these statistics, the medialness value M_0 of the current atom $\underline{m}_0$ is compared to the standard deviation σ_M of the randomly sampled atoms. Assuming that M_0 is computed at a core location, its value should be positive in sign and significant in magnitude relative to the random atoms. Using this reasoning, $\underline{m}_0$ is flagged as potentially being off of the desired core if $M_0 \leq 3 * \sigma_M$. I made the choice to use 3 standard deviations as the cutoff empirically and did not change this value during validation.

When extracting tubular objects, if at least four out of five consecutive medial atoms are flagged due to weak medialness, the core-following algorithm terminates and those five atoms are discarded. Cores can have weak responses at particular locations either because of a break in the object being tracked or because of noise. In either case the core-following algorithm should traverse weak locations rather than terminating when a single atom responds weakly. The ability to continue core-following despite up to three weak atoms

within a short distance produces desirable results, as shown in Chapter 7. Requiring four out of five weak atoms before terminating makes the method more robust. The method allows for the possibility that one out of five atoms gives a strong response due to noise even when the core is no longer following a desired object.

When extracting slab-like objects, the method does not require multiple weak responses – it immediately discards a single atom flagged due to weak medialness. Because slab-like cores march in four directions rather than along a curve, a small hole in an object does not terminate the core-following algorithm, as core-following can continue around that hole. Similarly, if a medial atom responds weakly and is discarded because of noise rather than because of an actual break, that hole will be filled in later as core-following wraps around the hole and predicts another medial atom coming in from a different direction. When the "predicted atom" queue described in section 4.3 is empty, the slab-like core-following algorithm terminates.

For tubes and slabs, this method effectively terminates the core-following algorithm both in situations where there is an explicit, or capped, end, and in situations where there is an implicit end in which the object tapers to the point that it is indiscernible. Results and analyses of the core-termination method as applied to both types of ends are presented in Chapter 7.

Chapter 7. Evaluation

7.1 Introduction

This chapter presents evaluations and analyses of the methods described in the previous chapters for computing cores of branching objects. Given the potential complexity and variability of anatomic objects, a system for evaluating the methods must be carefully designed; such a system is detailed here.

Section 7.2 demonstrates the methods' performance in clinical images by providing visualizations of anatomic objects extracted from 3D images using both 1D and 2D cores. These visualizations only provide a demonstration, not a quantification, of cores' capabilities. Section 7.3 presents a more rigorous evaluation of their accuracy and effectiveness. It describes a detailed evaluation of the behavior of cores as applied to both branching tubes and branching slabs, and it presents an analysis of the effects of varying object geometry on the behavior of cores to improve the understanding of their capabilities. The analysis in section 7.3 is performed using synthetic images.

Unless otherwise noted, the results presented in subsection 7.2.1 were computed with the pre-processing step of Westin *et al.* [Westin01], described in section 3.2, that uses an anisotropic adaptive filtering technique to reduce noise and enhance tubular structures. The results presented throughout the rest of this chapter were computed without use of this pre-processing step. Also, unless otherwise noted, all parameters for the core-following, branch-handling, and core-termination algorithms were set at the values described in Chapters 3-6, including fixing the medial atom object angle θ at $\pi/2$.

7.2 Demonstration of Results in Clinical Images

This section begins by demonstrating the success of tubular cores and then separately demonstrating the success of slab-like cores for computing the geometry of anatomic objects in medical images. The success of tubular cores is exhibited by their application to cerebral vasculature; 3D visualizations of vessel trees automatically extracted from head images are shown. Similarly, the success of slab-like cores is exhibited via visualizations of kidneys extracted from abdominal images. After tubular and slab-like cores are each examined, they are integrated to extract kidneys and adjoining vasculature from abdominal images.

7.2.1 Blood Vessel Trees as Branching Tubes

To demonstrate the success of tubular core-following with branch-handling and core-termination, the methods described in this dissertation were used to extract cerebral vasculature from medical images of the head. Healthy cerebral arteries are roughly tubular in shape and have, except at particular locations such as the Circle of Willis, a tree-like topology, making them good candidates on which to demonstrate the 1D core methods. Vasculature can be used to test the limits of 1D core-following and core-termination since vessels gradually narrow to widths significantly less than the inner scale of current medical images. In addition, cerebral vasculature is complex in structure and variable from person to person, requiring an effective branch-handling algorithm for a correct segmentation. The demonstrations provided in this subsection are meant to show that the 1D cores described in this dissertation, along with branch-handling and end-detection, provide an effective means for extracting the branching geometry of tubular structures in 3D medical images.

The methods for tubular core-following with branch-handling and core-termination were applied to the vasculature of three clinical magnetic resonance angiogram (MRA) data sets of the head. These data sets were acquired using a Siemens 3T system and were provided courtesy of Dr. Elizabeth Bullitt, UNC-Chapel Hill. Fig. 7.1 shows such an MRA and the corresponding 3D cores model automatically extracted from the MRA. This image was originally $512 \times 512 \times 91$ voxels with a resolution of $0.42 \times 0.42 \times 1.25$ mm. Before core-following, the image was cropped to speed up processing and was linearly interpolated in the axial direction to produce isotropic 0.42 mm voxels.

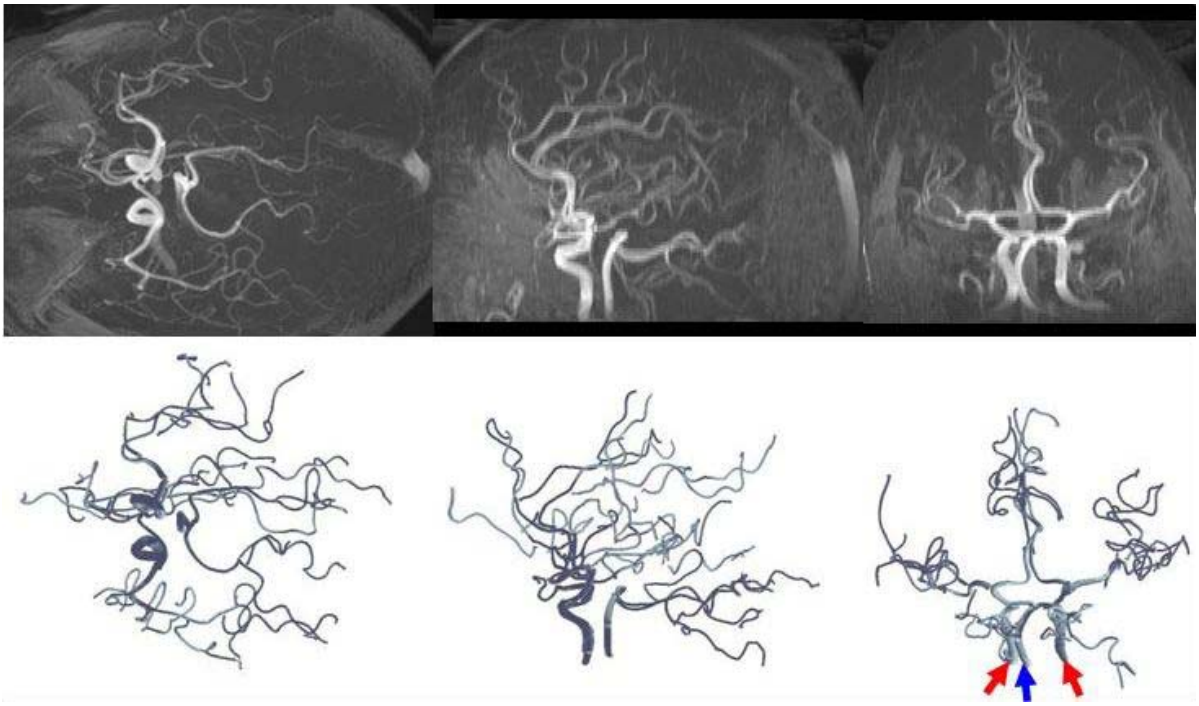

Fig. 7.1. Top: Maximum intensity projection images of a 389 × 300 × 273 voxel portion of a resampled MRA of the head. Bottom: Corresponding 3D cores model shown from slightly rotated axial (*left*), sagittal (*middle*), and coronal (*right*) viewpoints. Arrows indicate right and left carotid arteries (*red*) and basilar artery (*blue*).

In the example in fig. 7.1, core-following was manually initialized in three locations – the left and right carotid arteries and the basilar artery. These three arteries are clearly visible at the bottom of the coronal view of the model as well as at the bottom of the coronal view of the MRA. From these three initializations, the methods for core-following with branch-handling and core-termination were employed without further user interaction to achieve the results shown. Similar results were achieved when the algorithms were applied to the other two MRA's examined. In the three MRA's the branch-handling algorithm located 42, 46, and 47 branch points resulting in 85, 93, and 95 blood vessel segments, respectively. These qualitative results demonstrate that the described algorithms are capable of extracting the branching geometry of fairly narrow and complex tubular structures in 3D medical images. A more thorough analysis of the algorithms' capabilities is provided later in this chapter.

The algorithms were also applied to the same three images without the use of the pre-processing step described in section 3.2. These experiments were performed to determine the relative importance of the pre-processing step to the method as a whole. The experiments show that the pre-processing step barely affects the core-following and core-termination algorithms while the success of the branch-handling algorithm deteriorates somewhat when pre-processing is omitted. Of the 282 vessel segments found when pre-processing was employed, 259 (92%) were found when the pre-processing step was omitted. In addition, 2 false positive branches were found when pre-processing was not used. Of the 259 vessel segments found in both cases, the core-following algorithm lost track of only one vessel segment when pre-processing was not used. In the other 258 vessel segments, the core-following algorithm produced similar results regardless of whether or not pre-processing was used, as shown in fig. 7.2. The core-termination method also produced consistent results, as the location of termination of each of the 258 true positive vessel segments differed by at most two voxels between the case with pre-processing and the case without pre-processing.

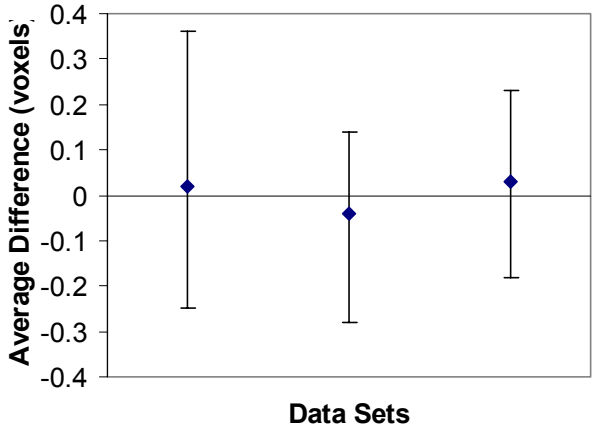

Fig. 7.2. To quantify the effects of pre-processing on the core-following algorithm, cores of blood vessels were computed in MRA data both with and without the use of pre-processing. For each medial atom computed without pre-processing, the shortest distance was found to a "corresponding atom" linearly interpolated between atoms computed with pre-processing. For each of three MRA data sets, the mean and standard deviations of these distances were computed, and the results are shown here. The maximum distance found between any pair of corresponding atoms in the three data sets was 1.22 voxels.

Running time for core computation of each of the MRA's tested was roughly four hours on a 2 GHz PC in the MATLAB environment. Tests show that the same computation would take approximately 25 minutes when implemented in C++.

7.2.2 Kidneys as Slabs

In this subsection I demonstrate the success of slab-like core methods for extracting kidneys from 3D images. The algorithms for core-following with branch-handling and core-termination were run on five clinical CT images of the abdomen provided courtesy of Dr. Sue Weeks and Dr. Stephen Aylward, UNC-Chapel Hill. The images each consist of 256 × 256 × 128 isotropic voxels with resolutions ranging from 0.90 to 1.25 mm/voxel. An iodine contrast agent was used during the acquisition of each image. These demonstrations are meant to show that the 2D cores described in this dissertation provide an effective means for extracting the geometry of slab-like structures in relatively low noise 3D medical images. The core-termination algorithm was employed during the extraction of the kidneys' cores. The branch-handling algorithm was also employed, even though kidneys do not branch, in order to see how the method would respond to the presence of a liver in each image. The liver sometimes abuts the right kidney, resulting in a weak image edge and causing the liver to appear as a branch of the kidney.

For each kidney in each of the five images, core-following with branch-handling and end-detection was performed ten times for a total of 100 test runs. Each test run was initialized at a different location within the given kidney. Of these 100 test runs, the abutting liver was detected as a branch 12 times, all 12 of which occurred while extracting the right kidney. In another 7 test runs the core clearly lost track of the kidney, as determined by inspection, 4 times while extracting the right kidney and 3 times while extracting the left kidney. Five of these 7 failures occurred near the ends, or crests, of the kidneys, where core tracking is known to be less stable (see sections 4.2 and 7.3.2.1). The other 2 failures occurred at locations of high surface curvature, where the medial atom optimization algorithm could not correctly adjust to the bending of the kidney in the presence of image noise. The remaining 81 test runs gave results similar to those shown in fig. 7.3.

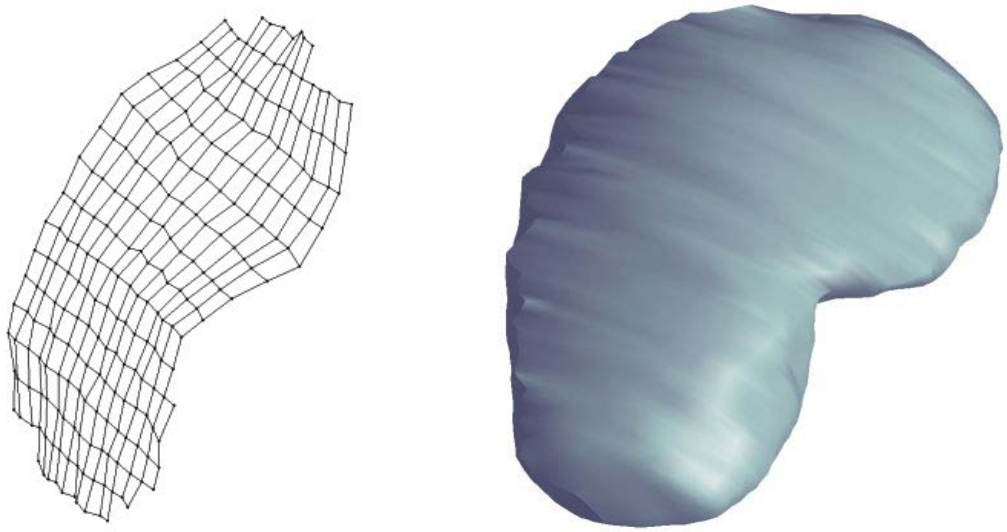

Fig. 7.3. Left: 3D view of the 2D slab-like core of a kidney, computed from 3D CT data. Medial atoms are shown as dots connected by line segments. Right: A visualization of the 3D kidney surface implied by the computed core.

These examples demonstrate the initial performance of 2D cores for the extraction of non-branching slab-like objects. The slab-like core-following method appears to be effective but not as robust as the tubular core-following method, as expected due to the methodological differences described in Chapters 3 and 4. Branch-handling during 2D core-following is discussed next, and a more rigorous analysis of 2D core-following is presented in subsection 7.3.2.1.

7.2.3 Kidneys and Adjoining Vasculature

Cores can be used to extract objects that are composed of both slab-like and tubular figures. In this subsection, the extraction of kidneys and adjoining renal arteries, renal veins, and ureters from 3D images demonstrates this capability as well as the effectiveness of the branch-handling method for slab-like cores. The same five CT data sets used in the previous subsection were used here, and once again 100 test runs were performed. In this case, I set an option in the program so that the methods would expect a slab-like main figure and tubular sub-figures.

Fig. 7.4 shows an example of a test run in which the kidney was extracted and the renal artery was automatically located as a branch, as described in section 5.7. The kidney and artery are separate figures with implied connections between medial atoms. Of the 100 test runs, the slab-like core-following algorithm lost track of the kidney's core 8 times. Six of these 8 failures occurred near the ends of the kidneys and the other 2 failures occurred at locations of high surface curvature, similar to the results of the previous section. In the remaining 92 test runs, the renal artery was located as a branch 84 times, the renal vein was located 79 times, and the ureter was located 34 times. Trials using synthetic images showed that the low success rate in finding the ureter was due to geometric causes, as the ureter often parallels the wall of the kidney.

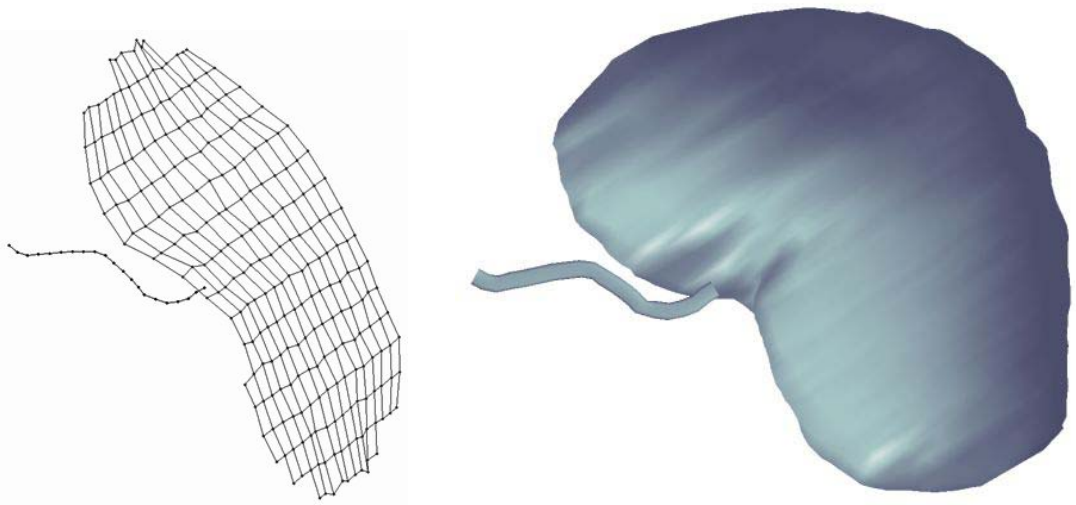

Fig. 7.4. Left: 3D view of the 2D slab-like core of a kidney along with the 1D tubular core of the attached renal artery, computed from 3D CT data as described in section 5.7. Right: A visualization of the 3D kidney and artery surfaces implied by the computed cores, viewed from a slightly different angle.

These trials demonstrate that the branch-handling method and the method for integrating slab-like and tubular cores are fairly effective but not extremely robust. The next section provides a detailed analysis of the methods presented in this dissertation, both slab-like and tubular.

7.3 Analysis of the Effects of Object Geometry on Cores in Synthetic Images

In this section I present a quantitative analysis of the effects of various object geometries on the performance of object extraction via cores. This analysis is meant to characterize the situations in which the core methods described in this dissertation can be effectively used. Given a particular type of object in a 3D image, this analysis can be used to predict the performance of these cores on the object. Analyses are performed separately for branching tubular objects and for branching slab-like objects and are presented in subsections 7.3.1 and 7.3.2, respectively.

In both subsections, noise is added to the synthetic images at three different levels: Gaussian noise with standard deviations per voxel of 40%, 20%, and 10% of the range of intensity values present in each image. The first two noise levels, which I will call "ultrasound" and "MR," resemble the noise present in ultrasound data and in 1.5T MR data, respectively [Aylward02]. The lowest noise level, which I will call "half MR," is meant to show the capabilities of cores as applied to images of the quality that can be expected in the coming years. These noise models are rough approximations. The ultrasound noise model is particularly rough, as the speckle pattern of ultrasound data is correlated (see [Czerwinski98] for more detail on the properties of ultrasound speckle). Nevertheless, experiments using synthetic images with these Gaussian noise models provide an indication of how well core algorithms will perform in medical images.

7.3.1 Branching Tubes

In the following subsections I describe the effects of object geometry on the tubular core methods of this dissertation. I do so by individually examining each phase of the process for computing 1D cores in 3D images, namely, core-following, branch-handling, and core-termination.

7.3.1.1 Tubular Core-Following

This subsection presents an analysis of the tubular core-following algorithm without accounting for branching and ending. The tests described next are designed to show the

extent to which the tubular core-following algorithm is robust and accurate when applied to tubular objects in medical images, even when the noise level in the images is high by today's standards. As an initial example of cores' impressive resistance to image noise, fig. 7.5 shows one slice of a three-dimensional image of a curved binary tube that has a radius of three voxels. Gaussian noise was added to the image with a magnitude per voxel of approximately five times what would be expected in MR data. Even in such a noisy image the core of the entire tube was extracted in 43 out of 50 test runs.

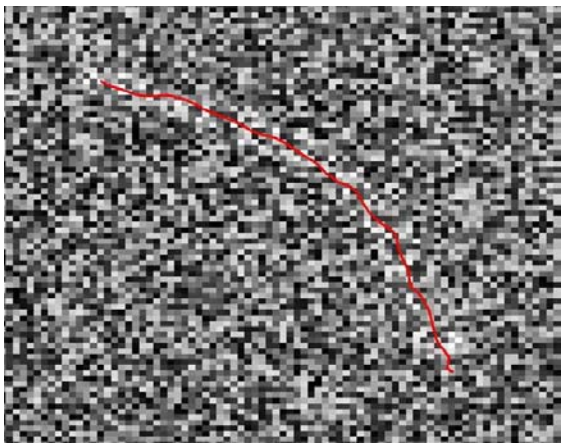

Fig. 7.5. One slice of a noisy synthetic 3D image of a tube of radius three pixels whose axis lies entirely in the plane displayed. The noise level is five times MR. The red curve is the computed core of the tube.

To better quantify the performance of the tubular core-following algorithm on objects such as blood vessels, two experiments were conducted. One experiment tested the extent to which the core-following algorithm can handle narrow tubes, specifically, tubes with a diameter on the order of one voxel. The other experiment tested the effects of tube curvature on the core-following algorithm. For both experiments, synthetic images were created to simulate blood vessels in MRA data. This was done by creating images containing tubular objects with parabolic profiles and then adding Gaussian noise to the images, according to the suggestions of Dr. Stephen Aylward, UNC-Chapel Hill, and [Hoogeveen98].

7.3.1.1.1 Narrow Tubes

To test the performance of the core-following algorithm on narrow tubes, I created a set of 80 images each containing a single tubular object. The tube shapes were based on 20 different curves. Each curve varied in both curvature and torsion along its length, with both the radius of curvature and the radius of torsion not decreasing below 10 voxels. None of these tubes were sharply bent since that situation is examined in the experiment discussed in subsection 7.3.1.1.2. For each of the 20 curves, four tube images were created, with each of the four tubes having the given curve as its centerline and having a diameter of 0.5 voxels, 1 voxel, 1.5 voxels, or 2 voxels. Three different levels of noise were added to each of the 80 images, as described at the beginning of section 7.3, to create a total of 240 images. The core-following algorithm was applied twice to each of the 240 tubes, initialized once at each end of each tube, resulting in 480 trial runs. Since this experiment was only meant to test the core-following algorithm, the core-termination algorithm was not employed. Rather, a rough termination location was manually specified for each tube, and core-following was stopped once the core crossed a plane through the specified location and orthogonal to the core. This way, the computed core never terminated early.

Fig. 7.6 shows the results of this experiment. Each curve in the two charts represents a different noise level, and each data point represents 40 trial runs at a given noise level and tube diameter. The data points in the chart on the left show the percentage of tubes in which every medial atom of the computed core was located within the volume of the tube. The data points in the chart on the right show the percentage of tubes in which every medial atom was located within one voxel of the tube's true centerline.

The results show that the core-following algorithm can be expected to reliably follow moderately tortuous tubes of diameter 2 voxels, as all 120 trial runs on such tubes were successful, regardless of the noise level. Moreover, since the core-following algorithm becomes more robust with increasing tube diameters, tubes of diameter greater than 2 voxels will be reliably followed as well. These results also show that in images with a level of noise typical of MR data, tubes as narrow as 1 voxel in diameter are reliably extracted by the core-following algorithm. The correct extraction of narrower tubes cannot be relied upon, although even tubes as narrow as half a voxel in diameter are correctly followed the majority

of the time. In images with a level of noise typical of ultrasound data, results of core-following begin deteriorating when tubes become less than 2 voxels in diameter, although only one of the 40 trials using tubes with a diameter of 1.5 voxels failed. Core-following in ultrasound data becomes quite poor with tubes of less than 1 voxel in diameter.

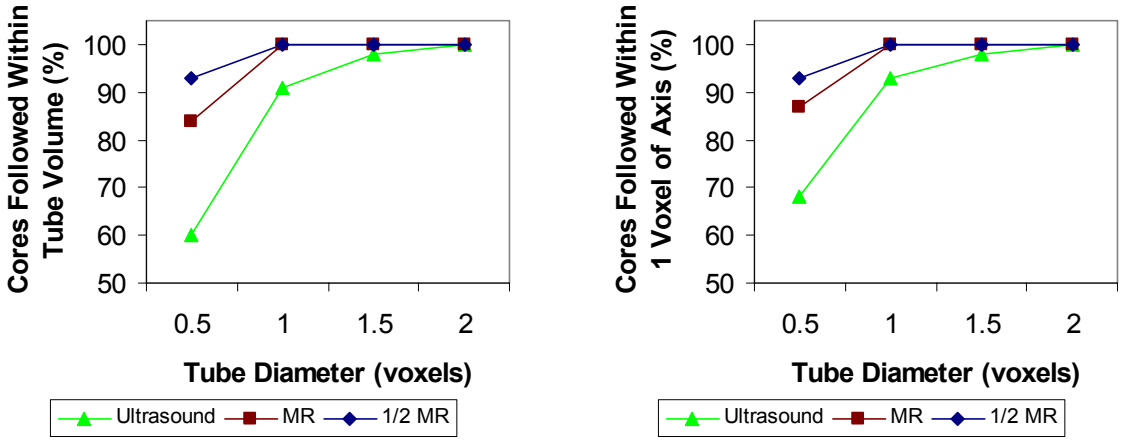

Fig. 7.6. The percentage of tubes correctly extracted from synthetic image data by the core-following algorithm. Each data point represents 40 trial runs at a given tube diameter and a given noise level, indicated by the three curves. Ultrasound, MR, and half MR correspond to Gaussian noise with standard deviations per voxel of 40%, 20%, and 10% of the image intensity values. The chart on the left shows the percentage of computed cores that remained within the volume of the tube, while the chart on the right shows the percentage of computed cores that remained within one voxel of the true centerline of the tube. The results show that the success of the core-following algorithm is related to tube width and is inversely related to image noise.

7.3.1.1.2 Bent Tubes

To test the performance of the core-following algorithm on bent tubes, I created 5 images, each containing a tube bent in the plane, *i.e.*, having no torsion, and having a diameter of 2 voxels. This diameter was chosen so as to not have a significant effect on the core-following algorithm's failure rate, as the intention is to test the effects of tube curvature, not tube diameter. Each tube formed a "U" shape. Specifically, each of the 5 tubes was straight for a distance of 20 voxels from one end, then curved in such a way that its centerline formed a semicircle, and finally continued straight for another 20 voxels. The curved

portions of the 5 tubes were given radii of curvature of 1, 2, 4, 8, and ∞ voxels (fig. 7.7). In the case of the tube with a radius of curvature of 1 voxel, the two straight parts of the "U" touch each other. At the other end of the spectrum, the tube with an infinite radius of curvature is straight. Like in the previous experiment, three different levels of noise were added to each of the images, resulting in 15 images.

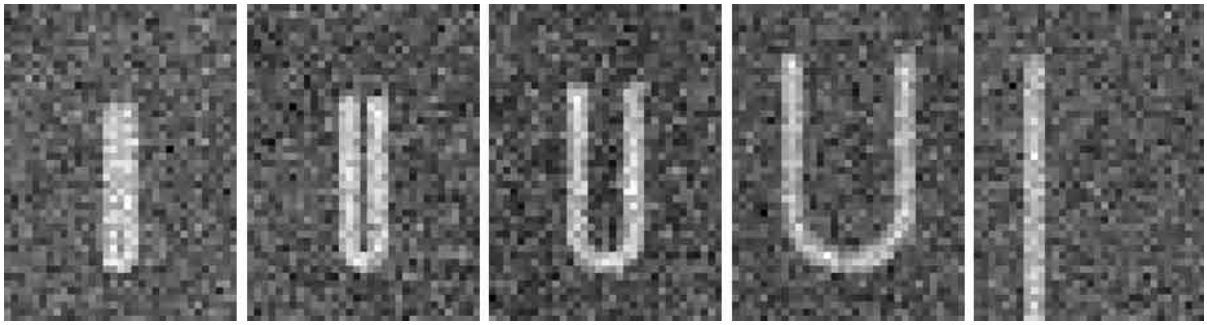

Fig. 7.7. Slices through synthetic 3D images of five bent tubes whose curved portions have radii of curvature of 1, 2, 4, 8, and ∞ voxels, from left to right. Noise typical of MR data was added to each image.

As described in section 3.4, the initial step size used in the tubular core-following algorithm is 1, but the step size is automatically decreased at tortuous locations. Experience has shown that when the core-following algorithm fails to follow a sharply bent tube, it often succeeds using a smaller step size. Thus, the experiment presented here is not meant to test what object curvature causes the core-following algorithm to fail; rather, it is meant to test what step sizes are required to successfully follow objects of varying curvatures.

For each of the 15 tube images created for this experiment, the step size for the core-following algorithm was set to the default value of 1 voxel, and the core-following algorithm was initialized 10 times, each time at a different location along a straight portion of the tube. In these experiments, the step size was not allowed to decrease automatically during core-following. If all 10 trials for a given image successfully followed the core through the bent portion of the tube, that set of trials was labeled a success. If any of the 10 trials failed, the step size was halved and a new 10 trials were conducted on the same image. This process was repeated until a step size was found at which all 10 trials correctly followed the core of

the tube. Fig. 7.8 shows the largest step size at which each of the 15 tubes was successfully extracted 10 out of 10 times.

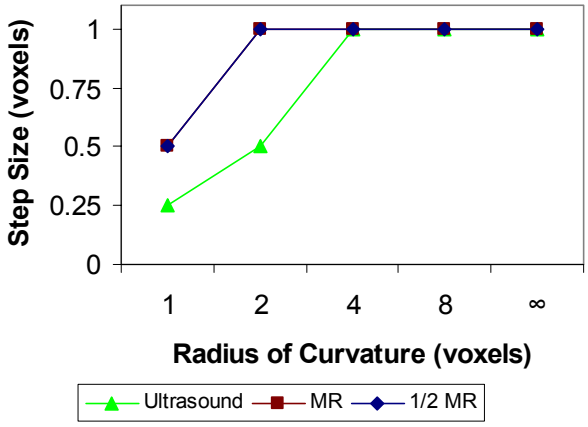

Fig. 7.8. The largest step size at which the core-following algorithm reliably followed the core of a curved tube. Each data point represents that a tube at the given noise level, indicated by the three curves, and with the given radius of curvature was correctly extracted 10 out of 10 times. The curves for MR and half MR are identical.

These results reflect the following reasoning. When following the core of a curved tube using the predictor-corrector marching algorithm described in section 3.4, the predictor portion of the algorithm estimates a new location by approximating the core as being locally linear. The larger the step size and the more curved the core, the worse the resulting estimate. Given a poor estimate, the corrector portion of the algorithm will be unlikely to find the correct core location. Not accounting for noise, the success of the algorithm in following the core of a curved tube is related to how well the core is locally approximated by a straight line. Since the step size can be decreased arbitrarily, there is hypothetically no limit to the curvature of tubes whose cores can be correctly followed. Trials were conducted using a step size of 0.01 voxels, tubes of diameter 2 voxels, and the three noise levels described earlier in an attempt to follow the cores of tubes having extreme amounts of both curvature and torsion. This included tubes with kinks, *i.e.*, tubes with a centerline having discontinuous tangent directions. Using such a small step size, every tube was correctly

extracted. Running time is inversely proportional to step size, and typical running times are given in subsection 7.2.1.

The experiments reported in these last two subsections show that the tubular core-following algorithm is robust in the presence of image noise and capable of extracting tubular objects of varying shapes and sizes. The only situation in which tubes cannot be reliably extracted from images with the level of noise typical in clinical data is when the diameters of the tubes are less than the inner scale of the image data. While the tube extraction method presented in [Aylward02] shows similar results, to my knowledge there is currently no method that can extract objects significantly narrower than a voxel without the aid of *a priori* models of the objects.

7.3.1.2 Tubular Branch-Handling

This subsection presents an analysis of the tubular branch-handling method described in Chapter 5. The analysis is meant to show the capabilities of the branch-handling method on branches of varying geometries, assuming the core-following algorithm has reached a branching region correctly.

Two geometric variations of tubular branches are analyzed here, branching angle and branch diameter. To analyze both, I created images with structures meant to simulate branching blood vessels in MRA data. Each branching blood vessel image was created by first computing a centerline with a single bifurcation. A tubular structure with a parabolic cross-sectional profile was then fit to each centerline.

7.3.1.2.1 Branching Angle

To test the effect of branching angle on the branch-handling method, tubes were created with four different branching angles. The smallest branching angle, which I refer to as an angle of 0 degrees, indicates that the two child branches have centerlines that share a tangent direction at the point where they branch and which then diverge. In graphics terminology, joining the centerline of the parent with the centerline of either child branch results in a G^1 continuous curve. The other three branching angles tested were 22.5 degrees,

45 degrees, and 90 degrees – their centerlines are G^0 continuous but not G^1 continuous (fig. 7.9).

Fig. 7.9. Slices through synthetic 3D images of four branching tubes with branching angles of 0, 22.5, 45, and 90 degrees, from left to right. Noise typical of MR data was added to each image.

7.3.1.2.2 Branch Size

To test the effects of various branch sizes on the branch-handling method, tubes were created with three different branch sizes. In all three cases, the two child branches were created in such a way that their cross-sectional areas at the location of branching sum to the cross-sectional area of the parent. This was done to mimic the shape of healthy blood vessels, which have this property for reasons of blood flow consistency. In the three branch sizes tested, the smaller of the two child branches had a cross-sectional area 0.10, 0.25, and 0.50 times the cross-sectional area of the parent. Put another way, the three respective ratios of the cross-sectional area of the smaller child branch to that of the larger child branch were 1:3, 1:√3, and 1:1 (fig. 7.10).

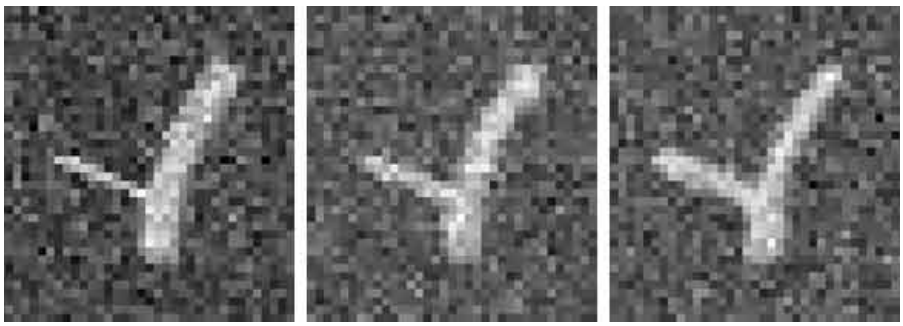

Fig. 7.10. Slices through synthetic 3D images of three branching tubes, where the smaller branch has a cross-sectional area 0.10, 0.25, and 0.50 times the cross-sectional area of the parent, from left to right. Noise typical of MR data was added to each image.

7.3.1.2.3 Combining Branching Angle and Branch Size

All possible combinations of the four choices of branching angles, three choices of branch sizes, and three levels of image noise described at the beginning of section 7.3 were used to create 36 images, each having a single branching tube. Tubular core-following with branch-handling was performed 10 times on each image, each time initialized at a different location along the trunk of the branching tube. A trial was considered a success if both branches were found and then followed for more than 5 steps; more detail is given in the next paragraph. Figs. 7.11 and 7.12 show the results of the 360 trials. The left and right sides of fig. 7.11 show the success of the branch-handling method versus branching angle and branch size, respectively, for the three different levels of image noise. Fig. 7.12 shows the combined effect of branching angle and branch size on the success of the branch-handling method for the highest of the three noise levels only, providing a breakdown of the 24 trials in which the branch-handling method failed under high noise. Of the 120 trial runs performed on images with the middle noise level, the branch-handling method failed on only three runs, all three of which had a small branch size (0.1 times the cross-sectional area of the parent). The branch-handling method succeeded in all 120 trial runs using images with the lowest noise level.

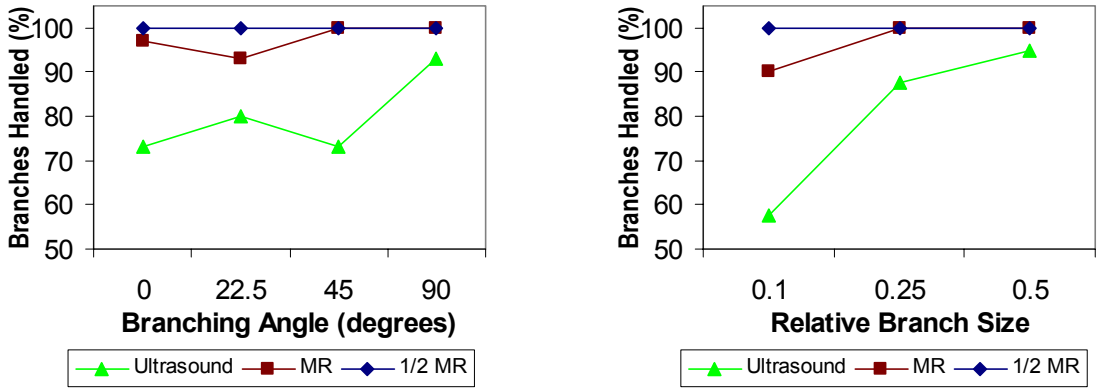

Fig. 7.11. The success of the branch-handling code on synthetic images of branching tubes with various branching angles and branch sizes. The chart on the left shows the percentage of branches correctly handled for each branching angle, over all branch sizes. Conversely, the chart on the right shows the percentage of branches correctly handled for each branch size, over all branching angles.

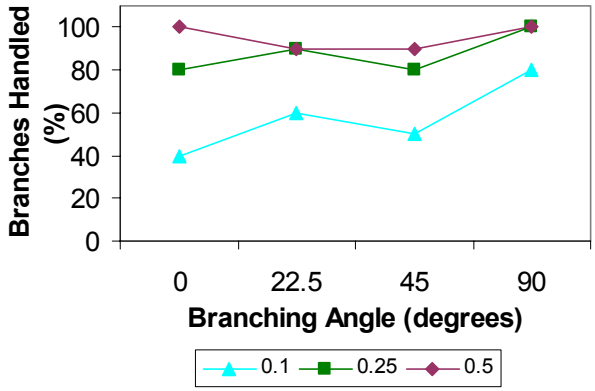

Fig. 7.12. The success of the branch-detection code on synthetic images with a noise level typical of ultrasound data. Unlike the charts in fig. 7.11, this chart shows the results of only the noisiest set of test images, and the three curves indicate different relative branch sizes. Each data point shows the percentage of test runs in which the branch was correctly detected for a given branch angle and a given branch size.

Since the branch-handling algorithm consists of two steps, branch-detection and continuation of core-following (see Chapter 5), it is useful to know which of the two steps was at fault in the cases where the branch-handling method failed. Of the 27 trial runs in

which the branch-handling method failed, branch-detection failed 26 times – in each of those trials core-following continued from the trunk of the tube through to one of the two child branches without detecting a branching location. Of the remaining 334 trials in which a branching location was detected, continuation of core-following failed in only one case. This case was the one with the largest branching angle, the narrowest child branch, and the highest level of noise. After the branching location was correctly detected in this case, both new cores followed the larger of the two child branches. This problem could potentially be solved by adding a test in the code that makes sure the same child branch is not followed twice. Still, the second step of branch-handling was shown to be effective, failing in only 1 out of 334 trials. The first step of branch-handling, however, did not produce such dependable results.

The results show that the branch-detection method is reliable in low noise images regardless of the branching angle or the sizes of the child branches. In images with a noise level typical of MR, the branch-detection method appears to be reliable as long as one child branch is not much wider than the other child branch, regardless of the branching angle. In noisy images such as ultrasound, the branch-detection method worked the majority of the time but did not appear robust, failing in nearly 20% of the trial runs. The failure rate was related to the difference in widths between the two child branches, while branching angle did not have a statistically significant ($p < 0.05$) effect on branch-detection. These results indicate that the tubular branch-handling method can be used reliably in medical images that are relatively noise-free by today's standards. In noisier images, the branch-handling method is unlikely to find false positives branches, but a significant number of true branches are likely to be missed.

7.3.1.3 Tubular Core-Termination

This subsection presents an analysis of the tubular core-termination method described in Chapter 6. As explained in that chapter, the core-following algorithm should terminate when there is a significant decrease in the image signal-to-noise ratio. The difficulty is in following the desired tubular object as long as possible but stopping when there is no basis in the image information for following the tube further. The analysis in this subsection is meant

to show the extent to which the core-termination algorithm accomplishes this for tubular objects such as blood vessels.

7.3.1.3.1 Tapered Ends

The core-termination algorithm was tested on a set of 20 synthetic images of tubes of varying but moderate curvatures and torsions that narrowed to a width of zero at different rates (fig. 7.13). As in subsection 7.3.1.1.1, both the radius of curvature and the radius of torsion of these tubes had a minimum of 10 voxels. The narrowing rates of the tubes, *i.e.*, half of the angle spanned by the tapered end of each tube, varied from 2 to 10 degrees. The fastest narrowing tube went from a diameter of 6 voxels to a diameter of 0 voxels over a distance of roughly 17 voxels, while the slowest narrowing tube went from a diameter of 3 voxels to a diameter of 0 voxels over a distance of roughly 43 voxels. Three different levels of image noise were added to each of these 20 images, as in subsections 7.3.1.1 and 7.3.1.2. These simulated images of narrowing blood vessels differ from clinical images of blood vessels in that real vessels narrow to a diameter much smaller than a voxel but never end, whereas these simulated vessels end at a specific location. It is thus possible to determine concretely whether the core-termination method works using the simulated vessels – if core-following continues past the location where a simulated tube has narrowed to a width of zero, core-termination has failed. The closer core-following comes to this tapered end, the better.

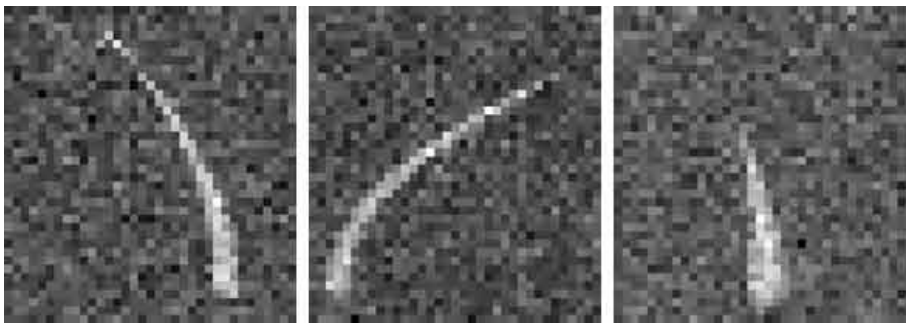

Fig. 7.13. Slices through synthetic 3D images of three different tubes that narrow to a width of zero. The tubes in these three examples have no torsion, but similar images were created of tubes with moderate amounts of torsion. Noise typical of MR data was added to each image.

Fig. 7.14 shows the results of 5 trial runs conducted on each of the 60 images, where each of the 5 trials on a particular image was initialized at a different location. Each pie chart shows what percentage of trial runs stopped the indicated number of voxels before the end of the tube being followed, one pie chart per noise level. No trial runs continued past the end of the tube. These charts show that in the lowest noise cases, the core-following algorithm terminated within 1 voxel of the true tube end in 96 of the 100 trials and either 2 or 3 voxels early in the remaining 4 trials. In the middle noise cases, the core-following algorithm terminated within 1 voxel of the tube end in 90 trials and at most 4 voxels early in the other 10 trials. In the highest noise cases, the core-following algorithm terminated within 1 voxel of the tube end in only 75 of the 100 trials. Still, of the 300 trial runs, core-following never continued past the true end of a tube, nor did it terminate more than 5 voxels early. These results show that while the core-termination algorithm may stop core-following before desired, particularly in noisy images, it reliably terminates core-following at reasonable locations.

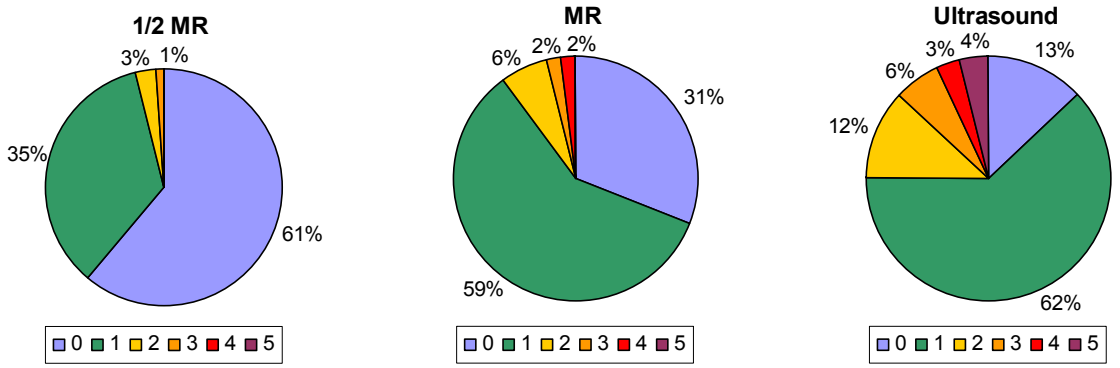

Fig. 7.14. The locations at which the core-termination algorithm terminated the following of tapered tubes. The three pie charts, from left to right, show results in low noise, medium noise, and high noise images having Gaussian noise with standard deviations per voxel of 10%, 20%, and 40% of the image intensity values. The legend values refer to how many voxels before the actual end of the tube core-following terminated, where zero means that core-following stopped exactly at the tapered end of the tube. Each pie chart shows the percentage of trial runs that stopped a given number of voxels early.

7.3.2 Branching Slabs

This subsection parallels subsection 7.3.1 in presenting an analysis of the effects of object geometry on slab-like cores, rather than on tubular cores, by individually examining each of the three phases for computing 2D cores of slabs in 3D images.

7.3.2.1 Slab-Like Core-Following

The analysis of the slab-like core-following algorithm is somewhat different from that of the tubular core-following algorithm due to differences in expected shapes of slab-like and tubular anatomic objects. Tubular objects in the human body, such as blood vessels, bronchi, and nerves, tend to be small in diameter, and they generally narrow or widen gradually. In contrast, slab-like objects can be wide but can have high rates of narrowing as they often have rounded or capped ends rather than tapered ends. As a slab-like core nears the edge of a kidney, for example, the narrowing rate of the kidney with respect to distance along the core approaches infinity. This behavior near rounded object ends differs from the behavior of cores computed using medial atoms with a variable object angle. Consider a slab-like object whose end is semi-circular in cross-section (fig. 7.15). In this cross-section, the variable object angle core ends at the center of curvature of the object's ends. The fixed object angle core used in this dissertation continues to the object boundary. When the object angle is fixed, the core-following algorithm is less stable in this outermost part of the core, as explained in section 4.2. Consequently, there is less of a need in the slab-like case to test the limits of how narrow of an object can be handled; rather, there is a need to test how well the core-following algorithm performs when applied to a rapidly narrowing object. Like tubular objects, however, slab-like objects can bend. With this motivation, I analyze the effects of narrowing rates and object bending on the performance of the slab-like core-following algorithm.

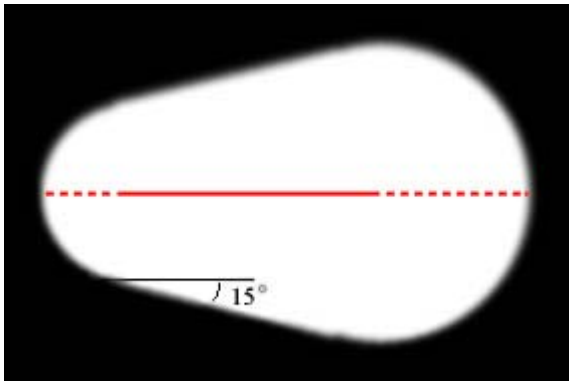

Fig. 7.15. A diagram showing a slice through a slab-like object that narrows/widens at a rate of 15 degrees. The solid red line represents a slice through the core of the object computed using medial atoms with a variable object angle. The solid and dotted red lines together represent the core computed using a fixed object angle. In this slice, the two flat boundaries of the object each form an angle of 15 degrees with line segments parallel to the core.

7.3.2.1.1 Narrowing Slabs

To better quantify the effects of object narrowing on slab-like cores, a set of 20 synthetic images was created of single-figure objects with opposing walls that narrow at differing rates – 4 images each of objects with opposing walls that form angles of 0, 15, 30, 45, and 60 degrees with the object's core (fig. 7.15). Each object was 100 voxels at its widest and narrowed to 20 voxels, with spherical capped ends as shown in fig. 7.15. Each object with a narrowing rate of 0 degrees was 50 voxels wide along its whole length of 200 voxels and had spherical capped ends. Three different levels of Gaussian noise were added to each image, as described at the beginning of section 7.3, resulting in 60 images. Slab-like core-following was performed 10 times on each of the 60 images, initialized at a different location within the object each time but always near the wide end of the object so as to advance in the direction of narrowing. Subsequently, core-following was performed another 10 times on each image, initialized near the narrow end of each object so as to advance in the direction of widening.

As described in section 4.2, the medialness response of a medial atom with two collinear spokes is proportional to the cosine of the angle of narrowing or widening. As a result, core-following is expected to be affected more by image noise as opposing object

boundaries deviate from parallel. This behavior is expected to be similar regardless of whether the core is advancing in the direction of narrowing or in the direction of widening.

Fig. 7.16 shows the results of the experiment conducted to test these predictions. As expected, the success rate of core-following was inversely related both to image noise and to the rate of narrowing or widening. For example, when following objects with parallel boundaries, 237 of the 240 test runs extracted the desired core without deviating more than three voxels from the known truth, a success rate of nearly 99%. The number of successful test runs decreased to 225 out of 240 (94%) at a narrowing/widening angle of 30 degrees, where the narrowing/widening angle is defined as the half angle shown in Fig. 7.15. The success rate then dropped dramatically to 169 out of 240 successful test runs (70%) at a narrowing/widening angle of 60 degrees. A separate analysis of the 600 trial runs starting from the narrow end of the object of interest and the 600 trial runs starting from its wide end showed no statistically significant difference ($p < 0.05$) in a chi-squared test between the narrowing/widening rates of the two sets of data.

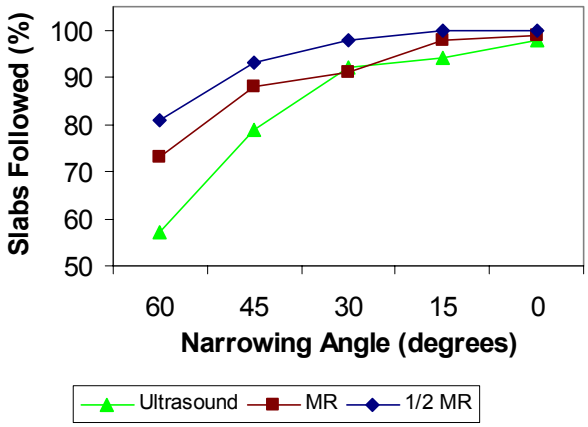

Fig. 7.16. The success of the core-following method on synthetic images of narrowing slab-like objects. Each data point shows the percentage of test runs in which the object was correctly extracted for a given noise level, indicated by the three curves, and a given narrowing angle. The results show that the performance of core-following is inversely related to both image noise and narrowing angle.

These results show that when opposing object walls are parallel or nearly parallel, the slab-like core-following algorithm performs well but not quite as robustly in the presence of image noise as does the tubular core-following algorithm. For example, at narrowing/widening angles of 15 degrees or less, core-following succeeded in all 160 trial runs on images with the lowest noise level. When the same objects were extracted from images with the highest noise level, core-following failed 7 times. The results also indicate that as the narrowing/widening angle increases significantly, as is common near slab-like object ends, the success rate of core-following decreases dramatically. The implications of this effect and possible improvements to the performance of the slab-like core-following algorithm are discussed in Chapter 8.

7.3.2.1.2 Tortuous Slabs

The second experiment used to analyze the slab-like core-following algorithm was designed to test the stability of cores in tortuous objects. Results of this experiment were expected to show that the behavior of core-following in tortuous slab-like objects is similar to that of core-following in tortuous tubular objects, presented in subsection 7.3.1.1.2. That is, any amount of curvature or torsion in an object can be handled by decreasing the step size of the core-following algorithm.

To test this hypothesis, I created a set of 8 images of helical ribbons having varying amounts of curvature and torsion and added three different levels of noise to each image. Each helical ribbon was given thickness by convolution with a ball of radius 3 voxels, and each image was blurred using a Gaussian of standard deviation 1 voxel before noise was added. Ten trial runs of core-following were performed on each of the 24 resulting images, with each trial run initialized at a different location within the object. Fig. 7.17 shows an example of a slab-like core extracted from one of these images. The image used in this example had the lowest level of noise and an object with fairly low curvature and torsion relative to the set of objects examined.

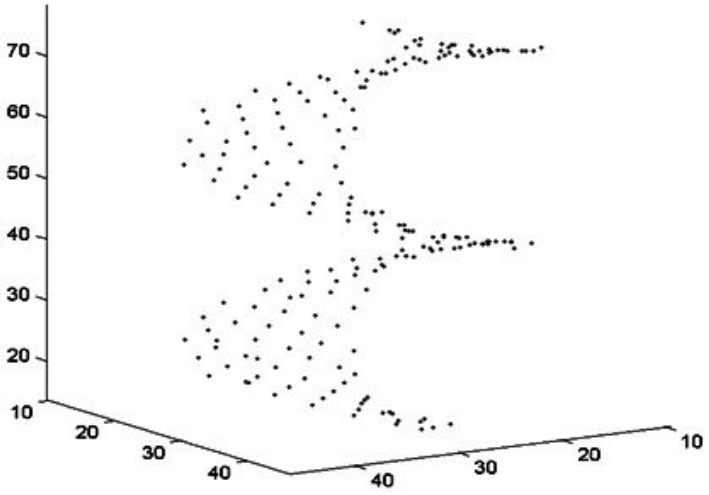

Fig. 7.17. A 3D view of the medial atoms computed from a synthetic image of a helical ribbon. The axis units are image voxels.

As expected, when the slab-like core-following algorithm was applied to objects with high curvature or high torsion, success rates improved by decreasing the step size of core-following. For example, when following the core of the most tortuous object in the low noise image using a step size of 1 voxel, only 3 of the 10 trial runs successfully followed the core, whereas all 10 trial runs successfully followed the core using a step size of 0.125 voxels. Unfortunately, a chart such as the one in fig. 7.8 could not be produced for slab-like core-following since some of the objects in noisier images were not extracted successfully regardless of step size. Specifically, 10 trials were conducted on each of the 24 images using a step size of 0.01 voxels. All 80 trials on low noise images successfully followed the core, 78 of the 80 medium noise trials successfully followed the core, and 73 of the 80 high noise trials successfully followed the core. These results verify the findings in the previous experiment on slab-like core-following that, although the slab-like core-following algorithm performs well in relatively noise-free images, it is not as robust as the tubular core-following algorithm in noisier images.

In an attempt to confirm the conjecture that cores of tortuous slab-like objects can be followed as well as those of objects with less curvature and torsion given a small enough step size, the results of the 240 trials conducted using a step size of 0.01 voxels were examined.

Of the 9 trials in which the core-following algorithm failed, 3 failures occurred when extracting the most tortuous object, 2 occurred when extracting the second most tortuous object, and 4 occurred when extracting the other 6 objects combined. Although these 9 failures provide too small of a sample size to draw definite conclusions, their distribution suggests that the behavior of the core-following algorithm differs between tubular objects and slab-like objects in that arbitrary tortuosity cannot be overcome by decreasing the step size.

After further experimentation, I believe the reason for this difference to be as follows. Although the tortuosity of an object does not directly lead to an increased difficulty in core-following when using a small enough step size, it can cause parts of the object to interfere with each other in the sense that a medial atom being fit to the object may be influenced by image intensities elsewhere along the object. For example, the edges located at opposite ends of a flat slab have little or no effect on each other, but if the slab is curved enough, the same edges may cause problems for the core-following algorithm (fig. 7.18), particularly in noisy images. This effect is not evident when following the cores of tubular objects. An explanation for this difference is that medial atoms used for computing slab-like cores have only two spokes and so are influenced by image intensities at nearby edges. On the other hand, when computing tubular cores, even if one or two spokes of a medial atom are influenced by a nearby edge, they tend to be "outvoted" by the other spokes.

Fig. 7.18. Diagrams showing slices through two different slab-like objects. When dealing with tortuous objects, such as the one on the left, object edges may interfere during core computation if they are close to each other and if the image is noisy. This is not a problem with straighter objects, such as the one on the right.

Although core-following of tortuous slab-like objects can be improved by decreasing the step size, such objects still cause difficulties for the core-following algorithm. Further tests are needed.

7.3.2.2 Slab-Like Branch-Handling

To analyze the performance of the slab-like branch-handling method, branches were divided into three categories: crest branches, face branches, and abutting objects. A detailed description of these types of branches is provided in section 5.3. Branch-handling is not used when extracting an object with a crest branch from a 3D image, since core-following initialized on the main figure continues through to the core of the sub-figure, as explained in section 5.6. This leaves face branches and abutting objects, and the branch-handling method was analyzed separately for each of these two types of branches.

7.3.2.2.1 Face Branches

To test the performance of the slab-like branch-handling method on face branches, a set of 32 images was created of objects with a main figure and one sub-figure. Eight images each were created with branching angles of 0, 22.5, 45, and 90 degrees. A branching angle of 0 degrees indicates that the two figures have medial sheets that share a tangent plane at the point where they branch and that then diverge. This is analogous to the case of tubes with a branching angle of 0 degrees, described in subsection 7.3.1.2.1. Three levels of noise were added to each image, and slab-like core-following with branch-handling was run ten times on each of the 96 resulting images, starting from a different location within the main figure each time.

As shown by the curve labeled "1/2 MR" in fig. 7.19, the branch-handling method succeeded at least 75 out of 80 times in each case having a low image noise, seemingly unrelated to branching angle. As noisier images were analyzed, the performance of the branch-handling method deteriorated. Unlike the low noise situation, in images with a level of noise typical of ultrasound, branching angle did have an effect on the success of the branch-handling method. The branch-handling method performed poorly in noisy images as the branching angle decreased.

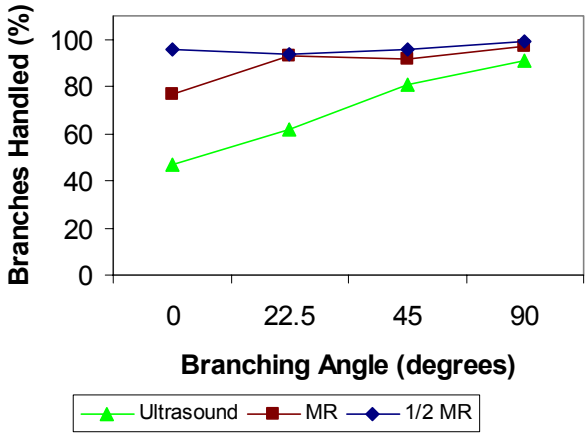

Fig. 7.19. The success of the branch-handling code on synthetic images of branching slab-like objects. Each data point shows the percentage of test runs in which the branch was correctly handled for a given image noise level, indicated by the three curves, and a given branching angle. The results show that the performance of the branch-handling method is related to branching angle and is inversely related to image noise.

This relationship between branching angle and the success of the branch-handling method is different from what was observed with tubular objects. In the case of slab-like objects, branching angle likely affects the performance of branch-handling due to the method for deciding which branches of slabs are true positives and which are false positives, described in subsection 5.4.2. The only heuristic used to make this decision is the radius of the core being followed, as a potential branch is rejected unless there is a significant increase in radius. For objects with a small branching angle, the increase in radius as the core passes through the branching region is relatively slow, so these branches are more likely to be rejected as false positives. The method for detecting false positive tubular branches uses a combination of several heuristics, making it more robust and less likely to reject true branches as false positives.

As was done in subsection 7.3.1.2.3 during the analysis of the tubular branch-handling method, the failures seen for slab-like branch-handling can be separated into failures in the branch-detection step and failures in the continuation of core-following step. The results displayed in fig. 7.19 include 138 failures out of 960 trial runs, and all 138 failures occurred in the branch-detection step. The step for continuation of core-following of

slab-like objects is quite robust, since a sub-figure is predicted from multiple locations on the parent figure, and only one of these predictions needs to find the sub-figure for this step to succeed (see section 5.6). Furthermore, if the algorithm predicts a figure that has already been extracted (as was the case in the failure example given in subsection 7.3.1.2.3 for branching tubes), the duplication is identified and core-following does not continue along that figure. This analysis shows that only the branch-detection step of slab-like branch-handling needs improvement in order to give reliable results in noisy images. Possible future improvements are discussed in Chapter 8.

7.3.2.2.2 Abutting Objects

The third possible slab-like branch type occurs in the case of two separate but abutting objects. There is currently no method for identifying abutting objects as such automatically. Rather, the expectation is that the branch-handling method either ignores an abutting object or recognizes it as a sub-figure rather than part of the main figure, in which case the user can manually indicate that the sub-figure be disregarded. Accordingly, the identification of an abutting object as a sub-figure is considered a success, as is the situation in which branch-handling misses the abutting object altogether. The only case considered a failure is when core-following continues from one abutting object to another, mistaking the two to be a single figure.

To test the success of the slab-like branch-handling method on abutting objects, 8 images were created, each having two abutting objects of equal size. The 8 images differed in the contact between the two objects, ranging from a single point of contact to contact along 20% of each object's surface. Three levels of noise were added to each image, and ten test runs were performed on each of the 24 resulting images, with the core-following algorithm initialized at a different location within one of the two objects each time. In 198 of the 240 test runs (83%) the branch-handling method identified the other object as a sub-figure. In another 41 of the test runs (17%) the branch-handling method missed the other object altogether. Not surprisingly, all 41 of these test runs occurred in cases where the two objects had a minimal area of contact. The core-following algorithm incorrectly extracted both objects as one figure in only one test run out of 240. This occurred in the image with

the largest area of contact and the highest level of noise. These results show that the slab-like core method is well suited to handle medical images with abutting objects.

7.3.2.3 Slab-Like Core-Termination

The core-termination algorithm described in Chapter 6 is applied in the same manner regardless of whether the object being considered is tubular or slab-like. Hence, the results shown in subsection 7.3.1.3.1 for tubular core-termination can be applied to slab-like core-termination as well. Moreover, slab-like core-termination generally provides less challenge than does tubular core-termination, as the most difficult situation for the core-termination algorithm is the one where an object tapers gradually until it is no longer apparent in the image information. This situation occurs most often with tubular objects, while slab-like objects typically have capped ends.

7.3.2.3.1 Capped Ends

When approaching capped ends, the cores of this dissertation ideally terminate at the object boundary. This differs from the behavior of cores with a variable object angle, which terminate before reaching the boundary. To determine the performance of the slab-like core-termination algorithm on objects with capped ends, the algorithm was tested on the 60 slab-like objects used subsection 7.3.2.1.1, as those test cases provided a variety of sizes and shapes of capped object ends. The code for core-following and core-termination was run only once on each of these 60 images. The core-following algorithm failed in 7 of the 60 test runs, losing track of the desired object. Those 7 runs were not considered in the results, as this experiment was intended to test the success of the core-termination algorithm given a correctly followed core. The other 53 cores, which appeared correct by brief inspection, had nearly 32 end atoms on average, where an end atom is an atom at which the core-termination algorithm decided to stop core-following. This provided a total of 1687 end atoms at which to evaluate the core-termination algorithm. All 1687 end atoms were located between the actual object boundary and a distance of 4 voxels toward the inside of the object from its boundary. The core-termination algorithm tended to stop core-following before the boundary because of the weak medialness responses given near the rapidly narrowing object ends. The end atoms located 3 or 4 voxels in from the boundary all occurred at locations of relatively

wide, rounded object ends. Although slab-like core-termination often stops before reaching the object boundary, these results show that the algorithm reliably terminates core-following near object boundaries without overshooting the boundary, even in noisy images.

Chapter 8. Conclusions and Discussion

8.1 Discussion

The earlier chapters of this dissertation presented methods for the extraction of branching object geometry, and the extraction of the objects themselves, from 3D grayscale images using a variant of optimum parameter cores. They presented methods for the computation of both 1D cores of branching tubular objects and 2D cores of branching slab-like objects. The methods compute both types of cores automatically, from one seed point per branching structure, and an indication as to whether the figures of interest are tubular or slab-like. The individual methods for core computation are core-following, branch-handling, and core-termination.

Core-following for tubular objects, as described in Chapter 3, is robust and capable of following cores of tubular objects of varying widths, curvatures, and torsions in noisy images. Cores of tubular objects more than two voxels in diameter are computed consistently and reliably in synthetic images meant to simulate images acquired by ultrasound, a relatively noisy modality. Cores are computed consistently even in images with significantly more noise than ultrasound images. In initial experiments, cores of tubes six voxels in diameter were computed correctly in 43 out of 50 images with a level of noise five times that of MR (subsection 7.3.1.1), providing confidence in the performance of the method in medical images. Cores of tubes less than two voxels in diameter are computed reliably in all but the noisiest medical images. For example, cores of tubes one voxel in diameter were computed correctly in 40 out of 40 images with a level of noise typical of MR. Cores of tubes less than one voxel in diameter can be followed correctly in many cases, but the core-following algorithm does not perform consistently on such narrow objects (subsection 7.3.1.1.1). Regardless of tube diameter, curvature and torsion are not limiting factors in core-following. Although core-following sometimes loses track of an object at

sharp bends when using the default step size of one voxel, it does not lose track if the step size is decreased appropriately (subsection 7.3.1.1.2). Consequently, cores of tubes of arbitrary curvature and torsion can be followed. These results for the core-following of tubes are on par with the results produced by the method of [Aylward02], a proven method for the traversal of narrow tubes in medical images. The robustness shown by the core-following algorithm presented in this dissertation is primarily due to the use of Gaussian derivatives as edge detectors at scale and the combination of constraints imposed on the structure of each medial atom employed during core-following.

Core-following for slab-like objects, as described in Chapter 4, is effective but not as robust as core-following for tubular objects in noisy images. The main difference between the two algorithms is that the medial atoms used for slab-like core-following have two collinear spokes, whereas the medial atoms used tubular core-following have a set of concentric spokes. The slab-like atoms are more susceptible to image noise, image artifacts, and boundary perturbations because of their smaller along-boundary extent and decreased constraints. These effects were discussed in more detail in section 3.3, and a potential solution is presented in section 8.3. Nevertheless, slab-like core-following consistently computes cores of slabs when the opposing object boundaries being followed by the two spokes are nearly parallel. Slab-like core-following becomes poor when an object narrows or widens rapidly, particularly when opposing object boundaries deviate roughly 45 degrees or more from parallel (subsection 7.3.2.1.1). Although this does not significantly affect the extraction of most anatomic objects, it is of concern as cores near object ends, since anatomic slab-like objects tend to narrow rapidly at their ends. This issue is discussed in section 8.3. Slab-like core-following also becomes poor in situations where an object's curvature and torsion are high. However, results appear to indicate that the level of curvature and torsion common in medical images does not have a negative impact on slab-like core-following when using a small enough step size (subsection 7.3.2.1.2). If 2D cores are to be used to analyze slab-like objects with high curvature or torsion, further research should be done on this matter.

Branch-handling for tubular objects, as described in Chapter 5, is fairly robust, allowing for the automated extraction of tree structures. In images that are noise-free by today's standards, branch-handling reliably locates branches of various sizes and branching angles and subsequently follows the cores of the new branches. In images with a level of noise typical of MR, branch-handling sometimes misses tubes with a diameter of one to two voxels. This occurs mainly in cases with small branching angles, parent tubes that are significantly wider than the missed children, and low signal-to-noise ratios at the bifurcations. Fortunately, no false positive branches have been encountered in experiments using images with this noise level, a desirable outcome considering that false positive branches are potentially more damaging in medical applications than are false negatives. In images with a level of noise typical of ultrasound, errors are common. While false positives are unlikely even in noisy images, the rate of false negative branches in experiments was nearly 20% (subsection 7.3.1.2.3). These failures occurred because the branch-detection method described in section 5.4 looks both for a corner in the image information and for a significant break in the tube wall caused by a bifurcation. In the case of a narrow child tube and a small branching angle these two indicators can be mistaken for image noise. Potential solutions are given in section 8.3.

Branch-handling for slab-like objects, as described in Chapter 5, is robust except in cases with small branching angles and high image noise. Recall that I consider three different types of branches, defined in section 5.3. The first type of branch, a crest branch, can be located and extracted from 3D images without the need for branch-handling, so I will not discuss it further. The other two types of branches, face branches and abutting objects, are not distinguished by the branch-handling method. The difference between the two is evident in the outcomes of core computations. When branch-handling is applied to a face branch, it is considered a success if it finds the sub-figure and a failure if it does not. When applied to 640 images with a level of noise typical of MR or less, branch-handling correctly found the face branch 596 times, a success rate of 93%. When the same experiment was conducted using 320 images with a level of noise typical of ultrasound, branch-handling correctly found the branch 226 times, a success rate of only 71% (subsection 7.3.2.2.1). Moreover, the success rate in this noisiest set of images was directly related to branching

angle (fig. 7.19). Accordingly, if the images being analyzed are fairly noise-free or if the branching angles are known to be near 90 degrees, branch-handling will likely find most branches. If the images are noisy and the branching angles are not known to be large, the method is unreliable. In either case, however, false positives are unlikely. For abutting objects, branch-handling is considered a success whether or not it finds the sub-figure and a failure only if it mistakes both objects to be a single figure. In experiments, branch-handling succeeded in 239 out of 240 images of abutting objects (subsection 7.3.2.2.2). This success is due to the sharp angle generally created between the boundaries of the abutting objects, easily identifiable by the cornerness operator (subsection 5.4.1).

Core-termination for tubular objects, as described in Chapter 6, works in tandem with core-following. Since some tubular objects, such as blood vessels, gradually narrow but never end, the aim is to follow a tube as long as possible without losing track. Core-termination reliably stops core-following before the signal-to-noise ratio becomes too low for core-following to work. The noisier the image is, the earlier the core terminates, but core-termination does not stop core-following when there is a small break in the object being followed, nor does it stop core-following at a location where an object is still clearly visible (subsection 7.3.1.3.1). For a more precise quantification, see fig. 7.14.

Core-termination is also reliable when applied to slab-like objects, which often have capped ends. Although the same core-termination algorithm is used for tubes and slabs, its behavior is different when object ends are explicit rather than implicit. In the case of explicit ends, there is a correct stopping location. But since slab medial atoms have spokes that are by definition collinear, the medialness responses rapidly deteriorate as a core nears a capped end (see section 4.2). As a result, core-termination generally causes core-following to stop short of the actual object end. While core-termination does not cause core-following to stop more than a few voxels early (subsection 7.3.2.3.1), section 8.3 discusses ways to further improve the accuracy of core-termination for slabs.

The methods for tubular core-following, tubular branch-handling, and tubular core-termination can be combined to extract the geometry of branching tubular structures from 3D medical images, as demonstrated in section 7.2. Similarly, the corresponding methods for

computing slab-like cores can be combined to extract the geometry of branching slab-like structures from 3D medical images. Finally, the tubular core methods and the slab-like core methods can be combined to compute the geometry of structures having both tubular and slab-like figures, as shown in subsection 7.2.3.

Overall, the methods presented in this dissertation are accurate and show impressive resistance to image noise, particularly when applied to tubular objects. The major strengths of these methods are their broad applicability and their ability to extract complex structures from 3D images without user interaction and without the need for prior information. A weakness of the methods is their lack of reliability in analyzing slab-like objects in noisy images. In addition, the methods are computationally expensive, limiting their potential use for certain applications that require fast results.

8.2 Review of Thesis and Contributions

This section restates the five scientific contributions claimed in section 1.4 and identifies how each has been accomplished by the work presented in this dissertation. It then reiterates the dissertation's thesis statement and defends the claims made in that statement.

CONTRIBUTION 1. *A robust implementation of 1D cores in 3D*

Chapter 3 presented a method for following 1D cores of tubular objects in 3D images. The method marches along a core using a predictor-corrector algorithm, at each step optimizing the fit of a filter composed of eight concentric Gaussian derivatives to the image data. In addition to optimizing over location and radius, as does the method of [Aylward02], the method presented in this dissertation is novel in that it optimizes over core orientation, although the effects of that difference have not been carefully studied. This method is highly resistant to image noise due in part to the concentricity of the Gaussian derivatives and the choice to constrain the medial atom object angle θ to $\pi/2$ radians (see section 3.3). It is capable of effectively following the cores of sharply bent tubes due to the adaptive step sizes used during marching, as shown in subsection 7.3.1.1.2. It makes use of techniques developed in other tube extraction methods but is novel in the way it is combined with

branch-handling and core-termination techniques, as described in the following contributions, in order to extract branching tubular structures from 3D images.

CONTRIBUTION 2. *A system for handling branches using 1D cores in 3D*

Chapter 5 described techniques for handling object branches during core-following. Specifically, section 5.2 contributes to knowledge by providing a detailed discussion of the properties and behaviors of 1D cores at branches of tubular objects. It explained the mathematical reason that cores of tubular objects necessarily bisect a bifurcation rather than following either branch. It then described the geometric attributes of branching tubular objects that, in practice, regularly cause core-following to jump from the core of the parent tube to the core of one of the child tubes.

Sections 5.4 and 5.5 presented a system for handling tubular object branches based on the properties discussed in section 5.2. While this system uses some techniques developed by others, it includes several original sub-contributions. Most significantly, the branch-handling method presented in this dissertation is based on the use of an affine-invariant corner detector applied to 3D images in a manner based on core geometry. The corner detector, which responds to curved image edges, is probed at locations defined by the medial atoms used during core-following (subsection 5.4.1). A key choice in applying the corner detector effectively was to compute it at the same scale as medialness (eq. 2.1). The resulting values provide dependable indications as to whether particular core locations correspond to object bifurcations.

A second sub-contribution associated with the system for tubular branch-handling is a technique for predicting the locations and attributes of the two child branches once an object bifurcation has been detected. This technique makes use of the geometry of the parent tube's computed core to predict the locations at which to start following the cores of the children. It also predicts the starting radii and orientations of both children's cores.

Finally, Chapter 5 contributes a means for extracting tubular tree structures from 3D images by presenting a system that combines the first two sub-contributions with contributions of others. This system can extract a tubular tree structure in one pass starting from a single seed point and requiring no user interaction or post-processing. As shown in

subsection 7.3.1.2, it is unlikely to follow false positive branches, and it only tends to miss true branches in cases of narrow branches and low signal-to-noise.

CONTRIBUTION 3. *A method for detecting ends of cores*

Chapter 6 presented a method for determining when to terminate the core-following algorithm. The method is based on a novel statistical analysis of the distribution of random atom medialness values and the resulting expected medialness values of random atoms versus atoms located along a core. This method has proven resistant to image noise and reliable for terminating core-following, whether the core being followed is tubular or slab-like and whether the object being extracted has an explicit, capped end or an implicit, tapered end.

Core-termination is also robust in the sense that it can traverse small breaks in an object without terminating core-following prematurely. This ability to traverse small object breaks is due to the choice to require a particular number of weak medial atoms before terminating core-following (see section 6.4).

CONTRIBUTION 4. *An implementation of 2D cores with branch-handling in 3D*

The methods from contributions 1, 2, and 3 were modified and further developed in this dissertation to allow for the extraction of slab-like branching objects from 3D images using 2D cores. Furthermore, the 1D and 2D core methods were integrated to allow for the extraction of objects that have both tubular and slab-like figures, as described in section 5.7 and demonstrated in subsection 7.2.3. This is a significant contribution, as it provides a system for segmenting general branching structures, not just tubular structures, using an image-based medial method that does not depend on a prior model.

Chapter 4 presented the first component of slab-like core computation, core-following. While the medial atoms described in section 4.2 are similar to medial atoms previously used by others, the way they are used to follow a 2D core by means of a coarse-to-fine grid-based marching algorithm is new, as explained in section 4.3. Section 4.4 then presented a means by which to apply penalties to slab medial atoms, a significant addition

meant to constrain the atoms and improve the resistance of the 2D core-following algorithm to image noise. These penalties were designed to limit the possible geometries of the resulting core in a manner consistent with expected geometries of anatomic objects.

Chapter 5 presented the method for slab-like branch-handling. Section 5.3 detailed the properties and behaviors of 2D cores at branches of slab-like objects, and sections 5.4 and 5.6 used this knowledge to describe a method for locating an object bifurcation and then re-seeding core-following along the sub-figure. Since a slab-like main figure gives no indication of the radius and orientation of the sub-figure, the range of radii and orientations explored during re-seeding makes this portion of the method possible (see section 5.6). The method was designed to be broadly applicable, able to handle a variety of branching situations.

Chapter 6 provided the last component of slab-like core computation, namely, core-termination. Although this component was nearly unchanged from the version applied to tubular cores, the behavioral differences between tubular and slab-like core-termination were described in that chapter. Given core-termination, a complete and automated system for computing 2D cores of branching slabs is realized. As shown in subsection 7.3.2, this system is effective in fairly low noise images but is not as robust as the corresponding system for computing 1D cores of branching tubes. It is particularly prone to error at locations where a figure widens or narrows rapidly and at bifurcations with a small branching angle and a small sub-figure.

CONTRIBUTION 5. *An analysis of the effects of object geometry on the performance of cores*

Chapter 7 presented a significant contribution to knowledge in the form of evaluations and analyses that provide a better understanding of the behaviors, capabilities, and limits of both the tubular and the slab-like core methods presented in this dissertation. These measurements were summarized with each contribution above.

Section 7.2 demonstrated the capabilities of these methods to extract branching structures from 3D medical images. It displayed visualizations of implied object surfaces created from cores computed automatically in grayscale images. It also gave brief descriptions of the causes for failure in cases where the core was not correctly computed.

Section 7.3 provided a thorough analysis of the effects of object geometry on core computation. This was accomplished by examining an assortment of geometric properties of both branching tubes and branching slabs. Each geometric property was varied, and the resulting behaviors of the core methods were explored. The section gave descriptions of each experiment, summarizing the effects of each geometric property on the core methods to aid in the reader's understanding of the key results. This analysis was performed on synthetic images so that results could be compared to known truth.

I now return to the thesis statement of this dissertation, which is as follows.

Optimum parameter cores with branch-handling and end-detection provide an automated means for computing the branching geometry of tubular structures in 3D medical images and for computing the branching geometry of general structures in relatively low noise 3D medical images.

The reviews of contributions 1, 2, and 3 describe the methods necessary to compute the branching geometry of tubular structures in 3D medical images. The work referenced in contribution 5, and specifically subsections 7.2.1 and 7.3.1 of this dissertation, detailed a thorough evaluation of these methods, supporting the first part of the thesis statement that, "Optimum parameter cores with branch-handling and end-detection provide an automated means for computing the branching geometry of tubular structures in 3D medical images."

The review of contribution 4 describes the extension of tubular core methods to slab-like core methods. The successes and failures of these slab-like core methods were scrutinized in subsections 7.2.2 and 7.2.3 for clinical images and in section 7.3.2 for simulated images with controlled parameters. Results support the second part of the thesis statement that, "Optimum parameter cores with branch-handling and end-detection provide

an automated means for computing the branching geometry of general structures in relatively low noise 3D medical images."

8.3 Conclusions and Future Work

The methods presented in this dissertation are intended to be general, allowing for the analysis of a wide variety of object shapes and structures. They are designed to handle arbitrarily complex tree structures, both tubular and slab-like, as well as structures where tubular child figures branch from slab-like parent figures. There are two particular object geometries, though, which the current methods cannot handle.

First, the core-following algorithm is not suited to compute the core of any tubular object that loops back on itself, such as a torus. Given such an object, the core-following algorithm will continue in circles *ad infinitum*, repeatedly computing the core of the same object. The solution to this problem is relatively straightforward. The tubular core-following algorithm is precise in the sense that the locations, radii, and orientations of a core computed during one pass around a torus, for example, will be identical to the corresponding values computed during another pass around the same object in the limit as the step size goes to zero. Accordingly, the algorithm can test whether a similar core already exists during each step of core-following. If so, the core-following algorithm should terminate. Likewise, the methods presented in this dissertation will fail given a structure in which a branch loops back to merge with a previously extracted figure, but a logic for detecting such cycles can be developed. The proposed solution is already adopted for core-following of slab-like objects, as explained in sections 5.6 and 6.4.

Second, the methods have not been developed to handle situations where a slab-like child figure branches from a tubular parent figure. This type of situation is uncommon in the human anatomy but is a potentially interesting modification to the current methods. Experience indicates that the existing methods for branch-detection, described in section 5.4, would effectively detect such branches. On the other hand, the methods for continuation of core-following, described in sections 5.5 and 5.6, are likely insufficient. Further experiments

are needed to determine the abilities of the current methods to handle such a branch and the modifications that would improve results.

The most important area of future work would be improving the consistency and the accuracy of slab-like core computation in noisy images. Although the current slab-like core methods are capable of effectively computing the cores of branching slab-like structures in relatively low noise medical images, results are unreliable in noisy images. The upcoming paragraphs suggest nontrivial improvements, potentially requiring significant amounts of further research.

The slab-like core-following algorithm is reliable at locations where objects show little narrowing or widening, as illustrated in subsection 7.3.2.1, but it becomes unstable at locations of significant narrowing or widening. When dealing with anatomic objects, this problem is of particular concern near object ends. The cores presented in this dissertation struggle with narrowing or widening objects primarily because medial atom object angles are constrained to $\pi/2$, not allowing the atom's spokes to remain normal to the object boundaries. Unfortunately, if this constraint is removed, slab-like core-following becomes even less stable due to the susceptibility of individual spokes to image noise, as detailed in section 4.2. A possible solution is to compute cores using a constrained object angle but to stop core-following at locations consistent with the ends of cores computed using unconstrained object angles. Cores with variable object angles stop well before reaching the end of the object, as described in section 2.7, significantly reducing the need for core computation at unstable locations. This modification would require much work for two reasons: 1) such a combination of core methods in which one variant is used for core-following and another is used for core-termination would be nontrivial, likely introducing mathematical difficulties due to the differing behaviors of the cores; 2) a replacement would have to be developed for the effective core-termination method of Chapter 6.

Another possible solution for improving the stability of slab-like core-following is to use slab medial atoms that are more like tube medial atoms in that they have a set of more than 2 constrained spokes. Slab atoms of more than 2 spokes would likely have 2 "pads" of

spokes designed to respond to opposing object boundaries. Each pad would be composed of multiple spokes preferably spread over a small region of the appropriate object boundary. The simplest form of a 2-pad medial atom would impose such limiting assumptions as parallel object boundaries and object boundaries with no curvature. To minimize the undesirable effects of these assumptions, a 2-pad medial atom could estimate the first- and second-order geometric properties of the object boundaries using knowledge from the individual spoke medialness responses. It could then adjust the shape of each pad of spokes accordingly, creating a better fit of the medial atom to the object boundaries. Significant research and development would be required to implement this in a way adequately robust against noise.

The slab-like branch-handling method is unreliable in noisy images, particularly in situations where the child figure is significantly narrower than the parent figure and the branching angle is small, as shown in subsection 7.3.2.2.1. This is a problem since it limits the possible applications of the core methods described here. A potential approach to reducing the effects of this problem is the application of an anisotropic filter to the image as a pre-processing step. A similar approach, using the filter described in section 3.2, significantly improved the success of the tubular branch-handling method. Although that filter was specifically designed for tubular objects, a comparable tactic could be used to improve the signal-to-noise ratio of images having slab-like objects. Another potential approach to improving the reliability of the slab-like branch-handling method is to develop a more comprehensive system for rejecting false positive slab branches (subsection 5.4.2). The current system only uses metrics on the core's radius, whereas the system for rejecting false positive tube branches combines a variety of metrics, making it more resistant to image noise. Developing such a system would require further research, since the choice of metrics and the manner in which to combine them are not obvious. Insight into this problem could also be gained by conducting additional research on the transitions of slab-like cores at branching locations. This topic was discussed in section 5.3.1, but it is still not understood well. A more detailed mathematical definition of the 3D Smoothed Local Symmetries formulation and a better understanding of its relationship to the slab-like cores presented in this dissertation could provide clues for improving the branch-handling method.

A possible use of the slab-like branch-handling method in particular, as suggested by Dr. Stephen Pizer, UNC-Chapel Hill, is to combine it with the m-reps method [Pizer03a] to find unmodeled sub-figures, or branches, attached to modeled figures. The m-reps method segments an image by deforming a prior model to objects in the image. The use of a prior model gives m-reps a significant advantage over cores in terms of stability and resistance to noise, but it also imposes the requirement for a prior model. This requirement precludes the use of m-reps on branches that cannot be modeled due to variability or unpredictability. The idea for future work is to take advantage of the stability of m-reps on figures that can be modeled, while at the same time applying the core branch-handling methods at locations defined by the m-reps in order to identify and follow unmodeled branches. This integration may be relatively straightforward, as both cores and m-reps are based on medial atoms. Further research is required, however, to determine the similarities and differences between the two methods and to gain a better understanding of the changes needed for such an integration to succeed.

A final possibility for future work is to update the implementation to improve the code's running time. Currently, the extraction of a blood vessel tree from head MRA data takes on the order of hours, and the extraction of a single kidney from CT data can take tens of minutes. Code optimizations have already been implemented, as the extraction of a single blood vessel segment took hours at one point, but much work remains to be done. One obvious modification is convert the code from MATLAB, it's current environment, to C++. I have already converted a key algorithmic portion of the code to C++ and have used that to estimate that a full conversion would improve running time roughly tenfold. However, a full conversion to C++ would be a major undertaking. Further code optimizations would also be a worthwhile endeavor. Since this code has been used only for research purposes, running time has not been a major concern. Given these described improvements in functionality, reliability, and running time, however, the cores presented in this dissertation could provide valuable data for a variety of clinical applications.

Bibliography

[Aylward96] Aylward, SR, SM Pizer, E Bullitt, D Eberly (1996). Intensity ridge and widths for tubular object segmentation and description. *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, **56**: 131-138.

[Aylward02] Aylward, SR, E Bullitt (2002). Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging*, **21**: 61-75.

[Blom92] Blom, J (1992). *Topological and Geometrical Aspects of Image Structure*. Ph.D. Thesis, Utrecht University.

[Blum67] Blum, H (1967). A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, W Whaten-Dunn, ed., MIT Press: 362-380.

[Blum78] Blum, H, RN Nagel (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, **10**: 167-180.

[Bouix00] Bouix, S, K Siddiqi (2000). Divergence-based medial surfaces. *European Conference on Computer Vision*, **1**: 603-618.

[Brady84] Brady, M, H Asada (1984). Smoothed Local Symmetries and Their Implementation. *The International Journal of Robotics Research*, **3(3)**: 36-61.

[Bruce96] Bruce, JW, PJ Giblin, F Tari (1996). Ridges, crests and sub-parabolic lines of evolving surfaces. *International Journal of Computer Vision*, **18**: 195-210.

[Bullitt01] Bullitt, E, SR Aylward, K Smith, S Mukherji, M Jiroutek, K Muller (2001). Symbolic description of intracerebral vessels segmented from magnetic resonance angiograms and evaluation by comparison with X-ray angiograms. *Medical Image Analysis*, **5**: 157-169.

[Burbeck96] Burbeck, CA, SM Pizer, BS Morse, D Ariely, GS Zauberman, J Rolland (1996). Linking object boundaries at scale: a common mechanism for size and shape judgements. *Vision Research*, **36**: 361-372.

[Canny86] Canny, J (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8(6)**: 679-698.

[Caselles97] Caselles, V, R Kimmel, G Sapiro (1997). Geodesic active contours. *International Journal of Computer Vision*, **22(1)**: 61-79.

[Chen95] Chen, H, J Hale (1995). An algorithm for MR angiography image enhancement. *Magnetic Resonance in Medicine*, **33(4)**: 534-540.

[Cohen92] Cohen, I, L Cohen, N Ayache (1992). Using deformable surfaces to segment 3-D images and infer differential structures. *CVGIP: Image Understanding*, **56(2)**: 242-263.

[Cootes95] Cootes, TF, D Cooper, CJ Taylor, J Graham (1995). Active shape models – their training and application. *Computer Vision and Image Understanding*, **61(1)**: 38-59.

[Cootes98] Cootes, TF, GJ Edwards, CJ Taylor (1998). Active appearance models. *Proceedings of the European Conference on Computer Vision*, **2**: 484-498.

[Cootes01] Cootes, TF, CJ Taylor (2001). Statistical models of appearance for medical image analysis and computer vision. *Proceedings of SPIE Medical Imaging*.

[Czerwinski98] Czerwinski, RN, DL Jones, WD O'Brien (1998). Line and Boundary Detection in Speckle Images. *IEEE Transactions on Image Processing*, **7(12)**: 1700-1714.

[Damon98] Damon, JN (1998). Generic structure of two-dimensional images under Gaussian blurring. *SIAM Journal on Applied Mathematics*, **59(1)**: 97-138.

[Damon99] Damon, JN (1999). Properties of ridges and cores for two-dimensional images. *Journal of Mathematical Imaging and Vision*, **10**: 163-174.

[Damon03a] Damon, JN (2003). On the smoothness and geometry of boundaries associated to skeletal structures I: sufficient conditions for smoothness. *Annales de l'Institut Fourier*, **53**: 1941-1985.

[Damon03b] Damon, JN (2003). On the smoothness and geometry of boundaries associated to skeletal structures II: geometry in the Blum case. To appear in *Compositio Mathematica*.

[Damon04] Personal communications with J Damon, University of North Carolina at Chapel Hill. August 6, 2004.

[de Bruijne03] de Bruijne, M (2003). *Model-Based Segmentation of Vascular Images*. Ph.D. Thesis, Utrecht University.

[Deschamps01] Deschamps, T, LD Cohen (2001). Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, **5**: 281-299.

[Deschamps02] Deschamps, T, LD Cohen (2002). Fast extraction of tubular and tree 3D surfaces with front propagation methods. *Proceedings of ICPR02*: 731-734.

[Du95] Du, YP, DL Parker, WL Davis. Vessel enhancement filtering in three-dimensional MR angiography. *Journal of Magnetic Resonance Imaging*, **5(3)**: 353-359.

[Duncan00] Duncan, JS, N Ayache (2000). Medical image analysis: progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22(1)**: 85-106.

[Eberly94] Eberly, D, RB Gardner, BS Morse, SM Pizer, C Scharlach (1994). Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, **4**: 351-371.

[Eberly96] Eberly, D (1996). Ridges in image and data analysis. *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, Netherlands.

[Frangi98] Frangi, AF, WJ Niessen, KL Vincken, MA Viergever (1998). Multiscale vessel enhancement filtering. *Medical Image Computing and Computer Assisted Intervention*, WM Wells, A Colchester, S Delp, eds., Lecture Notes in Computer Science, **1496**: 130-137.

[Frangi99] Frangi, AF, WJ Niessen, RM Hoogeveen, T van Walsum, MA Viergever (1999). Model-based quantitation of 3D magnetic resonance angiographic images. *IEEE Transactions on Medical Imaging*, **18**: 946-956.

[Fridman03] Fridman, Y, SM Pizer, S Aylward, E Bullitt (2003). Segmenting 3D branching tubular structures using cores. *Medical Image Computing and Computer Assisted Intervention*, RE Ellis, TM Peters, eds., Lecture Notes in Computer Science, **2879**: 570-577.

[Fridman04] Fridman, Y, SM Pizer (2004). *Extracting the geometry of branching slabs via cores*. Technical Report TR04-16, University of North Carolina at Chapel Hill, Computer Science Department.

[Fritsch93] Fritsch, DS (1993). *Registration of Radiotherapy Images Using Multiscale Medial Descriptions of Image Structure*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Fritsch95] Fritsch, DS, D Eberly, SM Pizer, MJ McAuliffe (1995). Stimulated cores and their applications in medical imaging. *Information Processing in Medical Imaging*, Y

Bizais, C Barillot, R DiPaola, eds., Kluwer Series in Computational Imaging and Vision: 365-368.

[Fritsch97] Fritsch, DS, SM Pizer, L Yu, V Johnson, E Chaney (1997). Localization and segmentation of medical image objects using deformable shape loci. *Information Processing in Medical Imaging*, Lecture Notes in Computer Science, **1230**: 127-140.

[Furst98] Furst, JD, SM Pizer (1998). Marching optimal-parameter ridges: an algorithm to extract shape loci in 3D images. *Medical Image Computing and Computer Assisted Intervention*, WM Wells, A Colchester, S Delp, eds., Lecture Notes in Computer Science, **1496**: 780-787.

[Furst99] Furst, JD (1999). *Height Ridges of Oriented Medialness*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Gerig92] Gerig, G, O Kübler, R Kikinis, FA Jolesz (1992). Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, **11**: 221-232.

[Giblin00] Giblin, PJ, BB Kimia (2000). A formal classification of 3D medial axis points and their local geometry. *Computer Vision and Pattern Recognition*, **11**: 566-573.

[Giblin03] Giblin, PJ, BB Kimia (2003). On the local form and transitions of symmetry sets, medial axes, and shocks. *International Journal of Computer Vision*, **54(1/2/3)**: 143-157.

[Golub89] Golub, GH, D O'Leary (1989). Some history of the conjugate gradient and Lanczos methods. *SIAM Review*, **31**: 50-102.

[Golub96] Golub, GH, CF Van Loan (1996). *Matrix Computations, 3rd ed*. Johns Hopkins University Press, Baltimore, MD.

[Hoogeveen98] Hoogeveen, RM, CJG Bakker, MA Viergever (1998). Limits to the accuracy of vessel diameter measurements in MR angiography. *Journal of Magnetic Resonance Imaging*, **8**: 1228-1235.

[Iijima62] Iijima, T (1962). Basic theory on the normalization of a pattern. *Bulletin of Electrical Laboratory*, **26**: 368-388. In Japanese.

[Kass88] Kass, M, A Witkin, D Terzopoulos (1988). Snakes: Active contour models. *International Journal of Computer Vision*, **1(4)**: 321-331.

[Katz03] Katz, RA, SM Pizer (2003). Untangling the Blum Medial Axis Transform. *International Journal of Computer Vision Special UNC-MIDAG issue*, **55(2/3)**: 139-153.

[Kelemen99] Kelemen, A, G Székely, G Gerig (1999). Elastic model-based segmentation of 3-D neuroradiological data sets. *IEEE Transactions on Medical Imaging*, **18(10)**: 828-839.

[Keller99] Keller, R (1999). *Generic Transitions of Relative Critical Sets in Parameterized Families with Applications to Image Analysis*. Ph.D. Thesis, of North Carolina at Chapel Hill.

[Kitchen82] Kitchen, L, A Rosenfeld (1982). Gray-level corner detection. *Pattern Recognition Letters*, **1**: 95-102.

[Knutsson89] Knutsson, H (1989). *Filtering and Reconstruction in Image Processing*. Ph.D. Thesis, Linköping University.

[Knutsson92] Knutsson, H, L Haglund, H Barman, GH Granlund (1992). A framework for anisotropic adaptive filtering and analysis of image sequences and volumes. *Proceedings of ICASSP*, **3**: 469-472.

[Koenderink84] Koenderink, JJ (1984). The structure of images. *Biological Cybernetics*, **50**: 363-370.

[Koenderink90] Koendering, JJ (1990). *Solid Shape*. The MIT Press, Cambridge, MA.

[Koller95] Koller, TM, G Gerig, G Székely, D Dettwiler (1995). Multiscale detection of curvilinear structures in 2-D and 3-D image data. *International Conference on Computer Vision*, E Grimson, S Shafer, A Blake, K Sugihara, eds.: 864-869.

[Leyton87] Leyton, M (1987). *Symmetry-curvature duality*. Computer Vision, Graphics, and Image Processing, **38**: 327-341.

[Lindeberg94] Lindeberg, T (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, Netherlands.

[Lindeberg98] Lindeberg, T (1998). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, **30(2)**: 117-156.

[Lorensen87] Lorensen, WE, HE Cline (1987). Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, **21(4)**: 163-169.

[Lorenz97] Lorenz, C, IC Carlsen, TM Buzug, C Fassnacht, J Weese (1997). A multiscale line filter with automatic scale selection based on the Hessian matrix for medical image segmentation. In BM ter Haar Romeny, LMJ Florack, JJ Koenderink, MA Viergever,

eds., *Scale-Space Theory in Computer Vision*, Lecture Notes in Computer Science. Springer-Verlag, **1252**: 152-163.

[Lorigo99] Lorigo, LM, O Faugeras, WEL Grimson, R Keriven, R Kikinis, CF Westin (1999). Co-dimension 2 geodesic active contours for MRA segmentation. *Information Processing in Medical Imaging*, A Kuba, M Sámal, A Todd-Pokropek, eds., Lecture Notes in Computer Science, **1613**: 126-139.

[Lorigo01] Lorigo, LM, OD Faugeras, WEL Grimson, R Keriven, R Kikinis, A Nabavi, CF Westin (2001). CURVES: Curve evolution for vessel segmentation. *Medical Image Analysis*, **5(3)**: 195-206.

[Malladi95] Malladi, R, JA Sethian, BC Vemuri (1995). Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17(2)**: 158-175.

[Marr80] Marr, D, E Hildreth (1980). Theory of edge detection. *Proceedings of the Royal Society of London*, **B207**: 187-217.

[Mather83] Mather, J (1983). Distance from a manifold in Euclidean space. *Proceedings of Symposium of Pure Mathematics*, **40**: 199-216.

[McInerney96] McInerney, T, D Terzopoulos (1996). Deformable models in medical image analysis: a survey. *Medical Image Analysis*, **1(2)**: 91-108.

[Metz85] Metz, CE (1985). Evaluation of image quality by ROC analysis: Concepts, techniques, and future possibilities. *Japanese Journal of Radiological Technology*, **4**: 990-1002.

[Metz86] Metz, CE (1986). ROC methodology in radiological imaging. *Investigative Radiology*, **21**: 720-733.

[Miller98] Miller, JE (1998). *Relative Critical Sets in R^n and Application to Image Analysis*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Miller99] Miller, JE, JD Furst (1999). The maximal scale ridge: Incorporating scale into the ridge definition. *Scale Space Theories in Computer Vision: Proceedings of Second International Conference*, M Nielsen, P Johansen, O Fogh Olsen, J Weickert, eds., Lecture Notes in Computer Science, **1682**: 93-104.

[Morse93] Morse, BS, SM Pizer, A Liu (1993). Multiscale medial analysis of medical images. *Information Processing in Medical Imaging*, HH Barrett and AF Gmitro, eds., Lecture Notes in Computer Science, **687**: 112-131.

[Morse94] Morse, BS (1994). *Computation of Object Cores From Greyscale Images*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Morse98] Morse, BS, SM Pizer, DT Puff, C Gu (1998). Zoom-invariant vision of figural shape: effects on cores of image disturbances. *Computer Vision and Image Understanding*, **69**: 72-86.

[Nackman82] Nackman, LR (1982). *Three-Dimensional Shape Description Using the Symmetric Axis Transform*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Näf96] Näf, M, O Kübler, R Kikinis, ME Shenton, G Székely (1996). Characterization and recognition of 3D organ shape in medical image analysis using skeletonization. *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*: 139-150.

[Nielsen02] Personal communications with M Nielsen, IT University of Copenhagen, Denmark. March 28, 2002.

[Ogniewicz90] Ogniewicz, RL, M Ilg (1990). Skeletons with Euclidean Metric and Correct Topology and their Application in Object Recognition and Document Analysis. *Proceedings of the 4th International Symposium on Spatial Data Handling*, **1**: 15-24.

[Ogniewicz92] Ogniewicz, RL, M Ilg (1992). Voronoi skeletons: theory and applications. *Computer Vision and Pattern Recognition*: 63-69.

[Orkisz97] Orkisz, MM, C Bresson, IE Magnin, O Champin, PC Douek. Improved vessel visualization in MR angiography by nonlinear anisotropic filtering. *Magnetic Resonance in Medicine*, **37(6)**: 914-919.

[Osher88] Osher, S, JA Sethian (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79(1)**: 12-49.

[Perona90] Perona, P, J Malik (1990). Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**: 629-639.

[Piegl96] Piegl, L, W Tiller (1996). *The NURBS Book, 2nd ed*. Monographs in Visual Communication. Springer-Verlag, New York, NY.

[Pizer98] Pizer, SM, D Eberly, BS Morse, DS Fritsch (1998). Zoom-invariant vision of figural shape: The mathematics of cores. *Computer Vision and Image Understanding*, **69**: 55-71.

[Pizer03a] Pizer, SM, PT Fletcher, S Joshi, A Thall, Z Chen, Y Fridman, D Fritsch, G Gash, J Glotzer, M Jiroutek, K Muller, G Tracton, P Yushkevich, EL Chaney (2003). Deformable m-reps for 3D medical image segmentation. *International Journal of Computer Vision Special UNC-MIDAG issue*, **55(2/3)**: 85-106.

[Pizer03b] Pizer, SM, K Siddiqi, G Székely, JN Damon, SW Zucker (2003). Multiscale medial loci and their properties. *International Journal of Computer Vision Special UNC-MIDAG issue*, **55(2/3)**: 155-179.

[Pohle01] Pohle, R, KD Toennies (2001). A new approach for model-based adaptive region growing in medical image analysis. In W Skarbek, ed., *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science. Springer-Verlag, **2124**: 238-246.

[Polak71] Polak, E. *Computational Methods in Optimization*. Academic Press, New York, NY.

[Poli96] Poli, R, G Valli (1996). An algorithm for real-time vessel enhancement and detection. *Computer Methods and Programs in Biomedicine*, **52(1)**: 1-22.

[Puff95] Puff, D (1995). *Human vs. Model Performance for Medical Image Estimation Tasks*. Ph.D. Thesis, University of North Carolina at Chapel Hill.

[Quek01] Quek, FKH, C Kirbas (2001). Vessel extraction in medical images by wave-propagation and traceback. *IEEE Transactions on Medical Imaging*, **20(2)**: 117-131.

[Sato98] Sato, Y, S Nakajima, N Shiraga, H Atsumi, S Yoshida, T Koller, G Gerig, R Kikinis (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, **2(2)**: 143-168.

[Schmitt89] Schmitt, M (1989). Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques. *Geometry and Robotics*, JD Boissonnat and JP Laumons, eds., Lecture Notes in Computer Science, **291**: 225-246.

[Shattuck01] Shattuck, DW, SR Sandor-Leahy, KA Schaper, DA Rottenberg, RM Leahy (2001). Magnetic resonance image tissue classification using a partial volume model. *Neuroimage*, **13(5)**: 856-876.

[Siddiqi99] Siddiqi, K, S Bouix, AR Tannenbaum, SW Zucker (1999). The hamilton-jacobi skeleton. *International Conference on Computer Vision*: 828-834.

[Siddiqi02] Siddiqi, K, S Bouix, A Tannenbaum, SW Zucker (2002). Hamilton-jacobi skeletons. *International Journal of Computer Vision*, **48(3)**: 215-231.

[Staib96] Staib, LH, JS Duncan (1996). Model-based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, **15(5)**: 720-731.

[Székely96] Székely, G, A Kelemen, C Brechbühler, G Gerig (1996). Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models. *Medical Image Analysis*, **1(1)**: 19-34.

[ter Haar Romeny02] ter Haar Romeny, BM (2002). *Front-End Vision and Multi-scale Image Analysis*. Kluwer Academic Publishers, Dordrecht, Netherlands.

[Terzopoulos88] Terzopoulos, D, K Fleischer (1988). Deformable models. *The Visual Computer*, **4(6)**: 306-331.

[Thirion96] Thirion, J, A Gourdon (1996). The 3D marching lines algorithm. *Graphical Models and Image* Processing, **58(6)**: 503-509.

[van Bemmel02] van Bemmel, CM, LJ Spreeuwers, MA Viergever, WJ Niessen (2002). Level-set based carotid artery segmentation for stenosis grading. *Medical Image Computing and Computer Assisted Intervention*, T Dohi, R Kikinis, eds., Lecture Notes in Computer Science, **2489**: 36-43.

[Vasilevskiy02] Vasilevskiy, A, K Siddiqi (2002). Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**: 1565-1578.

[Wells96] Wells, WM, WEL Grimson, R Kikinis, FA Jolesz (1996). Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, **15(4)**: 429-443.

[Westin00] Westin, CF, LM Lorigo, OD Faugeras, WEL Grimson, S Dawson, A Norbash, R Kikinis (2000). Segmentation by adaptive geodesic active contours. *Medical Image Computing and Computer Assisted Intervention*, SL Delp, AM DiGioia, B Jaramaz, eds., Lecture Notes in Computer Science, **1935**: 266-275.

[Westin01] Westin, CF, L Wigstrom, T Loock, L Sjoqvist, R Kikinis, H Knutsson (2001). Three-dimensional adaptive filtering in magnetic resonance angiography. *Journal of Magnetic Resonance Imaging*, **14**: 63-71.

[Wilson97] Wilson, DL, JA Noble (1997). Segmentation of cerebral vessels and aneurysms from MR angiography data. *Information Processing in Medical Imaging*. J Duncan, G Gindi, eds., Springer-Verlag: 423-428.

[Wilson99] Wilson, DL, JA Noble (1999). An adaptive segmentation algorithm for time-of-flight MRA data. *IEEE Transactions on Medical Imaging*, **18(10)**: 938-945.

[Witkin83] Witkin, AP (1983). Scale-space filtering. *Proceedings of the Eight International Joint Conference on Artificial Intelligence*: 1019-1022.

[Yomdin81] Yomdin, Y (1981). On the local structure of the generic central set. *Compositio Mathematica*, **43**: 225-238.

[Yushkevich02] Yushkevich, P, PT Fletcher, S Joshi, A Thall, SM Pizer (2002). Continuous medial representations for geometric object modeling in 2D and 3D. *Image and Vision Computing*, **21(1)**: 17-28.

[Zhang01] Zhang, Y, M Brady, S Smith (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. *IEEE Transactions on Medical Imaging*, **20(1)**: 45-57.