

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.



Bell & Howell Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600



## **NOTE TO USERS**

**This reproduction is the best copy available**

**UMI**



# Height Ridges of Oriented Medialness

by

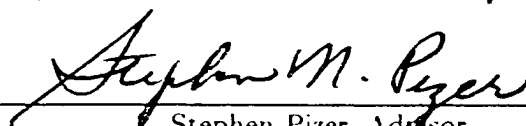
**Jacob David Furst**

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.

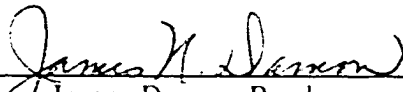
Chapel Hill

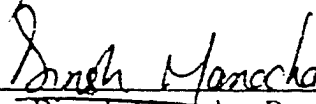
1999

Approved by:

  
Stephen Pizer, Advisor

  
Daniel Fritsch, Reader

  
James Damon, Reader

  
Dinesh Manocha, Reader

  
Donald Stanat, Reader

UMI Number: 9938145

Copyright 1999 by  
Furst, Jacob David

All rights reserved.

---

UMI Microform 9938145  
Copyright 1999, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized  
copying under Title 17, United States Code.

---

**UMI**  
300 North Zeeb Road  
Ann Arbor, MI 48103

Copyright © 1999  
Jacob David Furst  
All rights reserved

## ABSTRACT

**JACOB DAVID FURST: Height Ridges of Oriented Medialness**  
 (Under the direction of Stephen Pizer.)

Shape analysis of objects is an important aspect of medical image processing. Information gained from shape analysis can be used for object segmentation, object-based registration and object visualization. One shape analysis tool is the core, defined to be a height ridge of a medial strength measure made on an image. In this dissertation I present 3D cores, defined here to be optimal scale-orientation height ridges of oriented medial strength measurements. This dissertation covers 1) a medial strength measurement, *Blum-like medialness*, that is robust, efficient, and insensitive to intrafigural interference, 2) a new definition for a ridge, the *optimal parameter height ridge*, and its properties, and 3) an algorithm, *Marching Ridges*, for extracting cores. The medial strength measurement uses Gaussian derivatives, so is insensitive to noise and responds to object boundaries at points rather than on entire spheres, so is faster to calculate and less sensitive to boundaries of other image figures. The Marching Ridges algorithm uses the grid structure of the image domain to identify ridge points as zero-crossings of first derivatives and to track ridges through the image domain. I include results of this algorithm on medical images of cerebral vasculature, a skull, kidneys, and brain ventricles.



# Acknowledgments

Thanks to Steve Pizer, for his guidance, mentoring, support, encouragement, and sage advice. This dissertation is in large part a product of his love for science.

Thanks to my committee: Dan Fritsch, Jim Damon, Dinesh Manocha and Don Stanat, for carefully reading all my words and making many useful and meaningful suggestions for improvement. Further thanks to Don Stanat for taking a naive and somewhat ignorant young man into his world and introducing him gently and wisely to the world of graduate computer science and then at the end, coming out of retirement to serve on my committee.

Thanks to Jan Prins and Russell Taylor, who helped support me when medical image processing funds were low. Thanks also to the alumni of the Computer Science Department, for their generous gift of the Department Alumni Fellowship.

Thanks to Jason Miller, Rob Katz, and Rob Keller for their weekly participation in our interdepartmental, graduate student meetings. This work benefited greatly from the informal conversations we had.

Thanks to Andrew Thall for providing wonderful support in the completion of this dissertation while I was away in Chicago. Thanks also to Graham Gash, Carolyn Din, and Nicole Williams for the administrative and staff support over the entire course of this work. Their patience and assistance greatly eased some rough times.

Thanks to Trish Richardson, without whom I would not have come so far. Her unstinting support, her suggestions, and her love sustained me and gave me balance.

Thanks to my brothers, Syd Miller and Matt Furst, for the use of their couches, their unbiased perspectives and their emotional and intellectual support.

Thanks to my mother, whose progress towards her own doctorate provided a prod to mine, and for her commiseration and words of support.

Thanks to my father, whose example and sage wisdom kept me going from beginning to end.

Finally, thanks to the grants that actually paid my support: NIH grant CA47982, NIH grant RR02170, and ONR grant N00014-92-0182.

# Contents

<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Background . . . . .	2
1.3 Medialness . . . . .	3
1.4 Ridges . . . . .	7
1.5 Locus-finding Algorithms . . . . .	9
1.6 Thesis . . . . .	10
1.7 Contributions . . . . .	10
1.8 Organization . . . . .	11
<b>2 Background</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Medical Image Analysis . . . . .	12
2.2.1 Weighting Functions and Convolution . . . . .	12
2.2.2 Apertured Measurement Functions and Scale Space . . . . .	14
2.2.3 Generalized Product Spaces . . . . .	15
2.2.4 Invariance to Similarity Transformations . . . . .	15
2.3 Blum Symmetric Axis . . . . .	16
2.4 Medialness . . . . .	18
2.4.1 Isotropic Medialness Weighting Functions . . . . .	19
2.4.2 Oriented Medialness Weighting Functions . . . . .	21
2.4.3 Other Medialness Weighting Functions . . . . .	23
2.5 Ridges . . . . .	25
2.5.1 Maximum Convexity Height Ridges . . . . .	27
2.5.2 Height Ridges of Intensity . . . . .	28
2.5.3 Height Ridges of Boundariness . . . . .	29
2.5.4 Height Ridges of Medialness . . . . .	31
2.5.5 Height Ridges of Curvature . . . . .	32
2.5.6 Subdimensional Maximum Property . . . . .	35

2.6	Locus-Finding Algorithms . . . . .	35
2.6.1	Marching Cubes . . . . .	35
2.6.2	Tracked and Converged Partitioning . . . . .	36
2.6.3	Marching Lines . . . . .	37
<b>3</b>	<b>Medialness</b>	<b>38</b>
3.1	Introduction . . . . .	38
3.2	New Medialness Weighting Functions . . . . .	38
3.2.1	Extensions to Existing Medialness Weighting Functions . . . . .	38
3.2.2	Blum-like Medialness Measurement Functions . . . . .	41
3.3	Invariance to Similarity Transforms . . . . .	44
3.3.1	Functional Definition of Invariance . . . . .	44
3.3.2	Similarity Transformations . . . . .	46
3.3.3	Weighting Functions . . . . .	47
3.3.4	Zoom Invariance of Medialness Measurements . . . . .	48
3.3.5	Zoom Invariance of Medialness Derivatives . . . . .	49
3.4	Summary . . . . .	51
<b>4</b>	<b>Optimal Parameter Height Ridges</b>	<b>52</b>
4.1	Introduction . . . . .	52
4.2	Distinguished Parameter Height Ridges . . . . .	52
4.3	Natural Choice of Transverse Directions . . . . .	53
4.3.1	Choice of Distinguished Parameters . . . . .	54
4.3.2	Choice of Remaining Transverse Directions . . . . .	54
4.4	Reduction of Dimensionality . . . . .	55
4.4.1	Optimal Parameter Hypersurfaces . . . . .	55
4.4.2	Optimal Parameter Manifolds . . . . .	55
4.4.3	Ridges on Optimal Parameter Manifolds . . . . .	56
4.4.4	Optimal Parameter Height Ridges . . . . .	56
4.4.5	Calculating Derivatives of $f$ . . . . .	57
4.5	Proof of Subdimensional Maximum Property . . . . .	59
4.6	Summary . . . . .	62
<b>5</b>	<b>Marching Ridges</b>	<b>63</b>
5.1	Introduction . . . . .	63
5.2	Grid elements . . . . .	64
5.3	Interface . . . . .	64
5.3.1	Control buttons . . . . .	65
5.3.2	Image measurements . . . . .	66
5.3.3	Ridge dimension . . . . .	66
5.3.4	Maximizations . . . . .	66
5.3.5	Parameters . . . . .	66
5.3.6	Optimizations . . . . .	67

5.3.7	Object/Background Polarity . . . . .	67
5.3.8	Mouse . . . . .	67
5.4	General Ridge-finding Strategies . . . . .	68
5.4.1	Actions at Points . . . . .	68
5.4.2	Average Transverse Directions . . . . .	69
5.5	Specific Ridge-finding Strategies . . . . .	69
5.5.1	Ridge-finding on Edges . . . . .	70
5.5.2	Ridge-finding on Faces . . . . .	71
5.6	General Marching Strategies . . . . .	74
5.6.1	Initial Grid Element . . . . .	74
5.6.2	Expanding the Search . . . . .	74
5.6.3	Instantiation of New Grid Elements . . . . .	75
5.7	Specific Marching Strategies . . . . .	75
5.7.1	Curve Ridges . . . . .	75
5.7.2	Surface Ridges . . . . .	76
5.8	Complexity of Marching Ridges . . . . .	78
5.8.1	Time Complexity . . . . .	78
5.8.2	Space Complexity . . . . .	79
5.9	Summary . . . . .	79
<b>6</b>	<b>Discussion &amp; Results</b>	<b>81</b>
6.1	Overview . . . . .	81
6.2	Medialness . . . . .	81
6.2.1	3D Medialness Measurements . . . . .	82
6.2.2	Blum-like Medialness . . . . .	82
6.2.3	Invariances . . . . .	83
6.3	Height Ridges . . . . .	83
6.3.1	Optimal Parameter Height Ridges . . . . .	83
6.3.2	Reduction in Dimensionality . . . . .	84
6.3.3	Subdimensional Maximum Property . . . . .	84
6.3.4	Connectors . . . . .	85
6.4	Marching Ridges . . . . .	85
6.4.1	Synthetic Objects . . . . .	86
6.4.2	Medical Images . . . . .	95
6.4.3	Practical Considerations . . . . .	136
6.5	Conclusion . . . . .	137
	<b>Bibliography</b>	<b>138</b>

## List of Figures

1.1	Example landmarks in an image shown by X's. . . . .	2
1.2	Example boundaries in an image, outlining the corpus collosum and the cerebellum. . . . .	3
1.3	Example skeletons in an image, tracing the middles of a fold in the cerebrum, the corpus collosum, the brain stem and the cerebellum. . . . .	4
1.4	A disk and a fuzzy disk . . . . .	5
1.5	Medialness of fuzzy disk at radii of 20, 25, and 30 pixels (brighter pixels have greater medialness) . . . . .	5
1.6	"Center" and "boundary", determined by maximal medialness, superimposed on fuzzy disk. . . . .	6
1.7	Model medialness based on two antipodal points (left) and based on a semicircle (right). . . . .	6
1.8	A greyscale image of a human head viewed as a height function of the image space. . . . .	8
1.9	Following and crossing spaces of a mountain ridge. . . . .	9
2.1	A 2D Gaussian. . . . .	14
2.2	Example of a Blum symmetric axis (light grey) in 2D. . . . .	16
2.3	Example of a Blum symmetric axis (dark grey) in 3D. . . . .	17
2.4	2D Laplacian of Gaussian medialness weighting function. . . . .	20
2.5	2D Morse medialness weighting function. . . . .	21
2.6	2D Fritsch medialness weighting function. . . . .	22
2.7	2D $L_{pp}$ medialness weighting function. . . . .	22
2.8	2D Puff medialness weighting function. . . . .	23
2.9	2D Clary endness measurement function. . . . .	24
2.10	Low medialness weighting function with $N = 6$ . . . . .	25
2.11	Failure of maximal curvature ridge to identify top of ridge . . . . .	33
2.12	Diagram of a marching cube . . . . .	36
2.13	Marching cube divided into pyramidal simplices . . . . .	37
2.14	A marching line . . . . .	37
3.1	A wedge, a flare, and a cup . . . . .	42

3.2	2D Blum-like medialness measurement function. . . . .	43
3.3	A cone, a trumpet, a cap, and the direct product of a wedge and a flare . . . . .	44
3.4	Blum-like medialness weighting function for tube-like objects . . . . .	45
3.5	Blum-like medialness weighting function for slab-like objects . . . . .	46
4.1	False identification of local maximum. . . . .	59
5.1	Marching Ridges Interface . . . . .	65
5.2	Average transverse direction on an edge . . . . .	70
5.3	Interpolated zero crossing of the first directional derivative . . . . .	71
5.4	Positive identification of ridge point based on second directional derivative . . . . .	71
5.5	Average transverse directions on a face. . . . .	72
5.6	Interpolated zero crossings in first transverse direction . . . . .	72
5.7	Interpolated zero crossing in second transverse direction . . . . .	73
5.8	Positive identification of ridge point based on second directional derivatives . . . . .	73
5.9	Initial grid element . . . . .	74
5.10	A curve ridge continuing into an adjacent cube . . . . .	76
5.11	A surface ridge continuing into adjacent cubes. . . . .	78
6.1	Image of flared tube. The 3D image has been divided into slices of constant z and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices. . . . .	88
6.2	Image of flared tube (greyscale) with object implied by core superimposed (black). . . . .	88
6.3	Evaluation of core of flared tube, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation. . . . .	89
6.4	Image of a slab (greyscale). The 3D image has been divided into slices of constant z and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices. . . . .	91
6.5	Image of a slab (greyscale) with core object superimposed (black). . . . .	91
6.6	Evaluation of core of slab, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation. . . . .	92
6.7	Image of an ellipsoidal shell (greyscale). The 3D image has been divided into slices of constant z and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices. . . . .	94
6.8	Image of an ellipsoidal shell (greyscale) with core object superimposed (black). . . . .	94
6.9	Evaluation of core of ellipsoidal shell, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation. . . . .	95
6.10	Image of cerebral vasculature (metaslice 1 - image slices 5-8) . . . . .	97

6.11 Image of cerebral vasculature (metaslice 1 - image slices 5-8) with object implied by the core superimposed in black . . . . .	97
6.12 Image of cerebral vasculature (metaslice 2 - image slices 9-12) . . . . .	98
6.13 Image of cerebral vasculature (metaslice 2 - image slices 9-12) with object implied by the core superimposed in black . . . . .	98
6.14 Image of cerebral vasculature (metaslice 3 - image slices 13-16) . . . . .	99
6.15 Image of cerebral vasculature (metaslice 3 - image slices 13-16) with object implied by the core superimposed in black . . . . .	99
6.16 Image of cerebral vasculature (metaslice 4 - image slices 21-24) . . . . .	100
6.17 Image of cerebral vasculature (metaslice 4 - image slices 21-24) with object implied by the core superimposed in black . . . . .	100
6.18 Image of cerebral vasculature (metaslice 5 - image slices 25-28) . . . . .	101
6.19 Image of cerebral vasculature (metaslice 5 - image slices 25-28) with object implied by the core superimposed in black . . . . .	101
6.20 Image of cerebral vasculature (metaslice 6 - image slices 29-32) . . . . .	102
6.21 Image of cerebral vasculature (metaslice 6 - image slices 31-32) with object implied by the core superimposed in black . . . . .	102
6.22 Image of cerebral vasculature (metaslice 7 - image slices 33-36) . . . . .	103
6.23 Image of cerebral vasculature (metaslice 7 - image slices 33-36) with object implied by the core superimposed in black . . . . .	103
6.24 Image of cerebral vasculature (metaslice 8 - image slices 37-40) . . . . .	104
6.25 Image of cerebral vasculature (metaslice 8 - image slices 37-40) with object implied by the core superimposed in black . . . . .	104
6.26 Image of cerebral vasculature (metaslice 9 - image slices 41-44) . . . . .	105
6.27 Image of cerebral vasculature (metaslice 9 - image slices 41-44) with object implied by the core superimposed in black . . . . .	105
6.28 Image of cerebral vasculature (metaslice 10 - image slices 45-48) . . . . .	106
6.29 Image of cerebral vasculature (metaslice 10 - image slices 45-48) with object implied by the core superimposed in black . . . . .	106
6.30 Image of abdomen around kidneys (metaslice 1 - image slices 1-4) . . . . .	108
6.31 Image of abdomen around kidneys (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black . . . . .	108
6.32 Image of abdomen around kidneys (metaslice 2 - image slices 5-8) . . . . .	109
6.33 Image of abdomen around kidneys (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black . . . . .	109
6.34 Image of abdomen around kidneys (metaslice 3 - image slices 9-12) . . . . .	110
6.35 Image of abdomen around kidneys (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black . . . . .	110
6.36 Image of abdomen around kidneys (metaslice 4 - image slices 13-16) . . . . .	111
6.37 Image of abdomen around kidneys (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black . . . . .	111
6.38 Image of abdomen around kidneys (metaslice 5 - image slices 17-20) . . . . .	112

6.39 Image of abdomen around kidneys (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black . . . . .	112
6.40 Image of abdomen around kidneys (metaslice 6 - image slices 21-24) . . . .	113
6.41 Image of abdomen around kidneys (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black . . . . .	113
6.42 Image of abdomen around kidneys (metaslice 7 - image slices 25-28) . . . .	114
6.43 Image of abdomen around kidneys (metaslice 7 - image slices 25-28) with object implied by the core superimposed in black . . . . .	114
6.44 Image of abdomen around kidneys (metaslice 8 - image slices 29-32) . . . .	115
6.45 Image of abdomen around kidneys (metaslice 8 - image slices 29-32) with object implied by the core superimposed in black . . . . .	115
6.46 Image of abdomen around kidneys (metaslice 9 - image slices 33-36) . . . .	116
6.47 Image of abdomen around kidneys (metaslice 9 - image slices 33-36) with object implied by the core superimposed in black . . . . .	116
6.48 Image of skull (metaslice 1 - image slices 1-4) . . . . .	118
6.49 Image of skull (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black . . . . .	118
6.50 Image of skull (metaslice 2 - image slices 5-8) . . . . .	119
6.51 Image of skull (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black . . . . .	119
6.52 Image of skull (metaslice 3 - image slices 9-12) . . . . .	120
6.53 Image of skull (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black . . . . .	120
6.54 Image of skull (metaslice 4 - image slices 13-16) . . . . .	121
6.55 Image of skull (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black . . . . .	121
6.56 Image of skull (metaslice 5 - image slices 17-20) . . . . .	122
6.57 Image of skull (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black . . . . .	122
6.58 Image of skull (metaslice 6 - image slices 21-24) . . . . .	123
6.59 Image of skull (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black . . . . .	123
6.60 Image of skull (metaslice 7 - image slices 25-28) . . . . .	124
6.61 Image of skull (metaslice 7 - image slices 25-28) with object implied by the core superimposed in black . . . . .	124
6.62 Image of skull (metaslice 8 - image slices 29-32) . . . . .	125
6.63 Image of skull (metaslice 8 - image slices 29-32) with object implied by the core superimposed in black . . . . .	125
6.64 Image of skull (metaslice 9 - image slices 33-36) . . . . .	126
6.65 Image of skull (metaslice 9 - image slices 33-36) with object implied by the core superimposed in black . . . . .	126
6.66 Image of skull (metaslice 10 - image slices 37-40) . . . . .	127



6.67 Image of skull (metaslice 10 - image slices 37-40) with object implied by the core superimposed in black . . . . .	127
6.68 Image of ventricles (metaslice 1 - image slices 1-4) . . . . .	129
6.69 Image of ventricles (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black . . . . .	129
6.70 Image of ventricles (metaslice 2 - image slices 5-8) . . . . .	130
6.71 Image of ventricles (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black . . . . .	130
6.72 Image of ventricles (metaslice 3 - image slices 9-12) . . . . .	131
6.73 Image of ventricles (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black . . . . .	131
6.74 Image of ventricles (metaslice 4 - image slices 13-16) . . . . .	132
6.75 Image of ventricles (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black . . . . .	132
6.76 Image of ventricles (metaslice 5 - image slices 17-20) . . . . .	133
6.77 Image of ventricles (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black . . . . .	133
6.78 Image of ventricles (metaslice 6 - image slices 21-24) . . . . .	134
6.79 Image of ventricles (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black . . . . .	134

# Chapter 1

## Introduction

### 1.1 Motivation

Since the advent of X-rays, medical imaging has become an increasingly important tool in the diagnosis and treatment of illness. X-ray films, C'T scans, ultrasound, nuclear imaging and other imaging techniques all play a vital role in the practice of modern medicine. This dissertation is broadly about the analysis of these medical images as a tool to aid physicians in their work; of particular importance is the aspect of medical image analysis called shape analysis, in which a physician identifies and makes decisions about the shape of various objects found in medical images. Medical practitioners in fields as diverse as radiology, psychiatry, neurosurgery and radiation oncology use shape analysis in their work; it aids radiologists in identifying whether or not structures in an image are normal or pathological [42], psychiatrists in studying schizophrenia [44], neurosurgeons in conducting minimally-invasive surgery [5], and radiation oncologists in the more accurate direction of fatal radiation to cancerous cells [19]. Until recently, most shape analysis had been done manually, without the aid of computing devices, a process that may be slow and unreliable. However, there is an increasing use of computers to provide automatic and semi-automatic tools in shape analysis, which offer the hope of quick, repeatable analyses.

Such computing tools have typically used three different kinds of image structures in shape analysis: *landmarks* (Figure 1.1), points in the image defined to have particular importance [4], *boundaries* (Figure 1.2), the contour separating an object from its background [10], or *skeletons* (Figure 1.3), structures representing the middles of objects [28, 6, 38].

These analysis methods typically create shape representations for homologous objects in a set of training images using landmarks, boundaries or skeletons, and then perform a statistical analysis to create a model describing the range of objects encountered in the training images. This model is then compared to the shape representation of a homologous object in a new image to provide a quantification of the similarities and



Figure 1.1: Example landmarks in an image shown by X's.

differences between the model and the object. This quantification, in turn, can aid physicians in making diagnoses or deciding on plans of treatment. These structures can also be used in segmentation tasks to penalize shape change of such a model.

Recently, Pizer [21] and the Medical Image Display and Analysis Group (MIDAG) at the University of North Carolina at Chapel Hill (UNC-CH) have done studies of object shape analysis using *cores*, a type of object skeleton defined as a height ridge of medialness. Using this definition of cores requires an understanding of measurements of medial strength called *medialness* (Section 1.3), and *ridges* (Section 1.4) while the extraction of cores requires a knowledge of *locus-finding algorithms* (Section 1.5). This dissertation is about the extraction of cores from objects in medical images for use in shape analysis and contributes in each of the above three areas. The core finding algorithms and tools I have developed here show promise for many medical imaging tasks as well as tasks in other disciplines where imaging is a basic means of gathering information.

## 1.2 Background

Medical image analysis is a multidisciplinary field including medical practitioners and researchers, computer scientists, biologists, mathematicians, and psychologists, to list a few. As a computer scientist, I have written primarily about ideas and algorithms, although I include a solid mathematical background for the ideas and a summary of the



Figure 1.2: Example boundaries in an image, outlining the corpus collosum and the cerebellum.

results of the algorithm on test images and medical images. To support understanding, I have included a broad chapter containing background material I used in creating the work in this dissertation. Included are a section on some basics of medical image analysis as they relate to my work, a brief survey of current ideas about the representation and description of object shape, a basic but rigorous introduction to the Blum symmetric axis, a description of past work in medialness measurements, a definition of height ridges and a survey of such ridges already being used in medical image analysis, and a quick introduction to three relevant locus finding algorithms.

### 1.3 Medialness

Medialness is a way of measuring for the middles of objects and is used in this dissertation for the extraction of cores. The concept of medialness arises from Blum's work on symmetric axes (Chapter 2) and has been refined extensively at UNC-CH. To provide an intuitive notion of what medialness is, I will use an analogy for medialness based on some standard geometric and algebraic notions.

The formula for a disk (Figure 1.4) is  $(x - k)^2 + (y - h)^2 \leq r^2$  in which  $(k, h)$  is the center of the disk and  $r$  the radius. Any disk can be described in this way by a point (the center of the disk) and an associated width (the radius of the disk). This simple description of disks is possible because of their mathematical nature, with complete and perfectly distinct boundaries. Objects in medical images rarely have complete or even

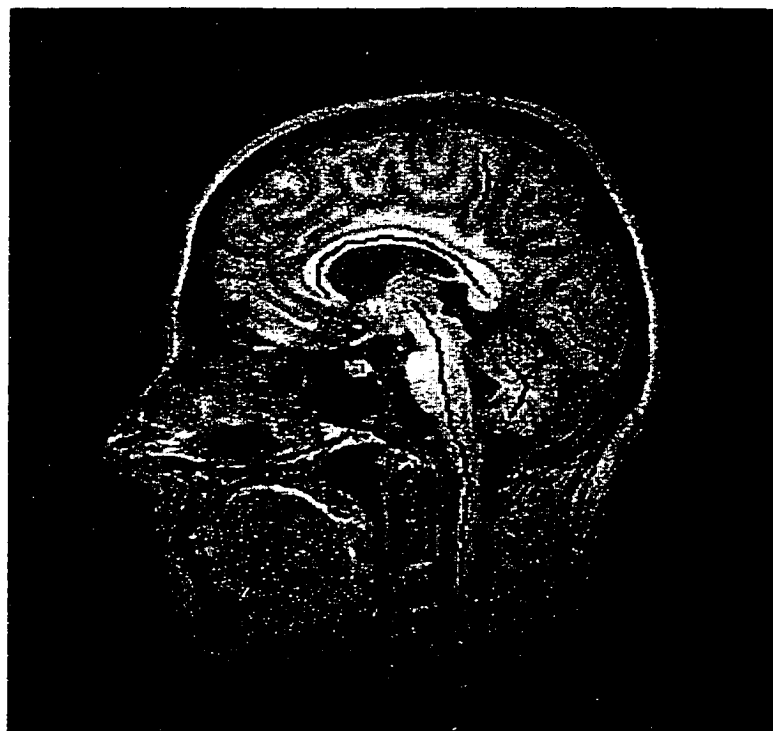


Figure 1.3: Example skeletons in an image, tracing the middles of a fold in the cerebrum, the corpus callosum, the brain stem and the cerebellum.

easily recognizable boundaries, as one tissue transitions into another. However, there are objects that may appear roughly disk-like, and it would be efficient and useful to be able to describe them as simply as one can describe a disk, with a center location and a radius. In fact, this simplification of representation is one of the key components of making useful shape models.

Imagine, then, a disk with an indistinct and perhaps broken boundary (Figure 1.4), as might appear in a medical image. It would be difficult to describe the exact center of the disk and the corresponding radius. In fact, Pizer [38] has taken the approach that it is not necessary (or even possible) to know the center or radius of such a disk exactly. Instead, from among all the points in the disk, and from among all possible radii, choose the point and radius that are most likely the center and the appropriate radius. To make this choice, one must be able to measure for this property of being like a center at an appropriate radius; the choice then amounts to finding the maximum of the measurement. This measurement is called medialness. Thus, to find the "center" of a disk with broken and indistinct boundaries, measure medialness at all positions and all radii. Figure 1.5 shows medialness of the disk measured at three radii. The position and radius pair having the highest value of medialness is the center (as well as it can be known) of the disk. Figure 1.6 shows the center (black dot) of the disk determined by medialness as well as the appropriate radius, shown by the black circle. To look at this in other way, one can imagine that the mathematical equation for circle,

$(x - k)^2 + (y - h)^2 = r^2$ , is a very rigidly defined medialness in which plugging in values of  $(h, k)$  and  $r$  yield a response of 0 (when the circle fails to match the boundary of the disk) for any point not in the center of the disk or not at the correct radius, or a response of 1 (when the circle does match the boundary) for the actual center and correct radius. The center of the disk, with a measurement of 1, would be the maximum of the measurement over all possible positions and radii, since any other combination would produce a measurement of 0. Medialness provides a way of grading this binary value so that in medical images the complexity of the image can be reflected in the medialness values measured.

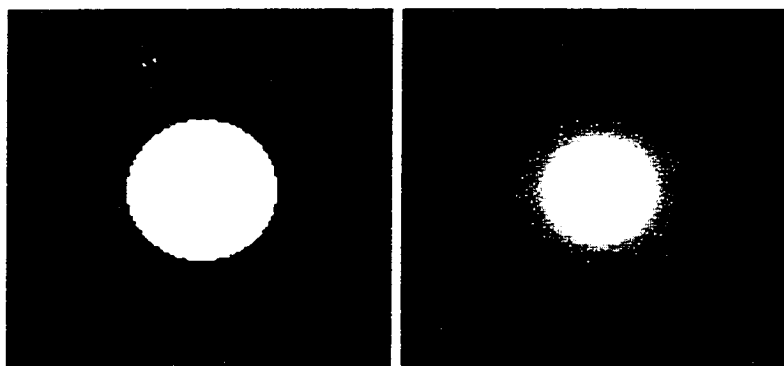


Figure 1.4: A disk and a fuzzy disk

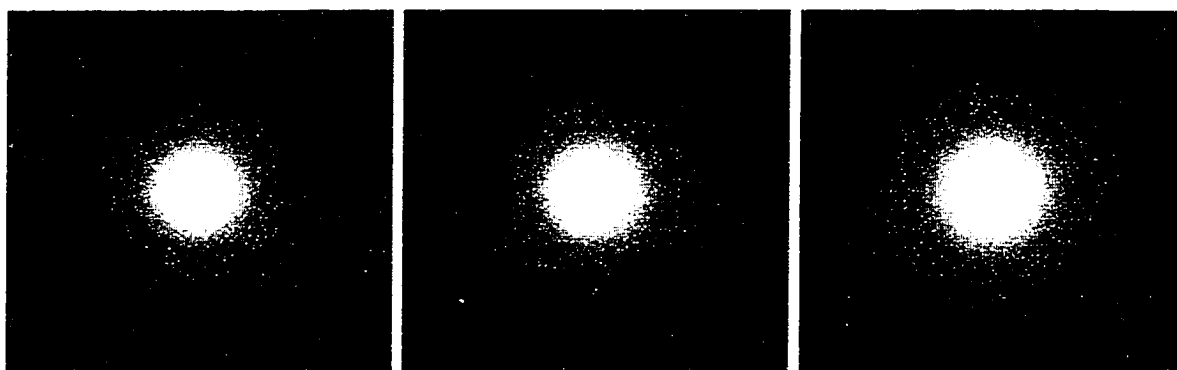


Figure 1.5: Medialness of fuzzy disk at radii of 20, 25, and 30 pixels (brighter pixels have greater medialness)

There are many ways to expand on this intuitive notion of medialness. One is to model medialness not after a circle, but instead as a disk, in which the emphasis is not on the medialness measurement matching the boundary of the object, but on the overlap between the medialness measurement and the object. This has the advantage (in medical images) of being less sensitive to noise at the boundary of the object, since the emphasis is on the difference between the average intensity of the object versus the average intensity of its background. Another way is to use portions of the circle rather than the entire

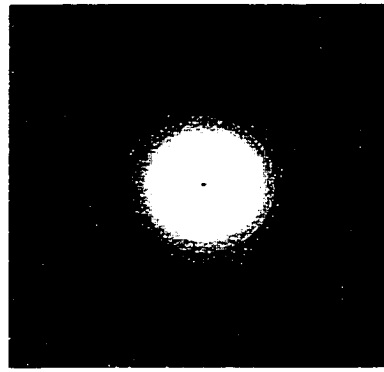


Figure 1.6: "Center" and "boundary", determined by maximal medialness, superimposed on fuzzy disk.

circle. This provides the advantages of being able to measure medialness of non disk-like objects and of speed, since one can check for boundaries at isolated points rather than along the entire circumference of the circle. (For example, when measuring medialness of bar-like objects with parallel sides, only two points on a circle are interesting in regard to matching boundaries of the object.) A natural choice for pieces of the circle might be two antipodal points (as in the example of the bar) (Figure 1.7 - left) or an entire half circle (for instance, if the bar were to have a semicircular end cap) (Figure 1.7 - right).

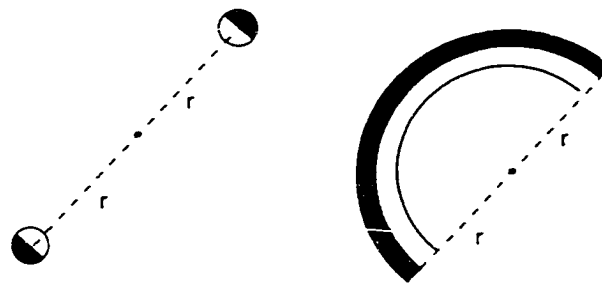


Figure 1.7: Model medialness based on two antipodal points (left) and based on a semicircle (right).

This approach requires some specification of which two points to use, or which particular semicircle, done with an orientation parameter. This way of measuring medialness requires not only checking all possible radii for every position, but also all possible angles of orientation (thus all pairs of antipodal points or all semicircles). The same choices could be made for medialness modeled on the disk. Further, while each of these notions is 2D, all of them can be extended to 3D (spheres and balls) or to any arbitrary dimension using hyperspheres and hyperdisks. (I am interested in static medical images, so I will only use medialness measurements for 2D and 3D spaces.) All of these intuitive definitions have been quantified and used by researchers at UNC-CH in 2D and will be discussed in Chapter 2. My own work on medialness (Chapter 3) includes 1) exten-

sions of the 2D medialness measurements into 3D, involving decisions about the nature of the core to be extracted and providing a basis from which to extrapolate higher dimensional medialness measurements; 2) a new classification of medialness measurements based on implicit properties of the measurement and type of core extracted from medialness, providing a framework from which to choose medialness measurements best suited for a particular task; and 3) a new medialness measurement, called *Blum-like medialness*, modeled on a circle (or sphere) for which any two arbitrary points (requiring two angles to define) are used to measure medialness, providing an efficient and natural way to measure medialness for points in an object of arbitrary shape.

## 1.4 Ridges

Ridges are intuitively extended “high” places. However, this intuitive notion falls far short of any useful application in medical image analysis tasks. Thus, many more specific notions have arisen, which I will describe intuitively here and capture mathematically in Chapter 2.

One kind of ridge can be illustrated by a hypothetical walk in the mountains. As you walk, always maintain the steepest possible drop on both your left and right sides. You can then be sure that you are on a ridge. Another kind of ridge can be illustrated by walking always uphill. If, in your walk, you take a sharp right or left turn to keep going uphill, you have started walking on a ridge. A third kind of ridge can be illustrated by walking along the direction in which the mountain is most curved, keeping track of how curved it is. Every time you get a maximum of curvature, mark that point as being on the ridge. A fourth kind of ridge can be illustrated by walking at a constant altitude in the mountains. Every time you take a sharp left or right turn, you have crossed a ridge. If you walked at all altitudes and marked the ground every time you crossed a ridge, the collection of such marks would be a ridge. A much different way to think intuitively about ridges is to look at the boundaries between watersheds, such as the continental divide of the Rocky Mountains which separates the United States into that part in which water flows to the Pacific basin and that part in which water flows to the Atlantic. The boundaries between the final catch basins (in this case, the Pacific and Atlantic) are ridges. The mathematical foundations for these intuitive notions are an important tool in medical image analysis.

Unfortunately, these intuitive notions all attempt to describe ridges on the familiar two dimensional surface of our planet in terms of mountains. In medical image processing, ridges defined on image intensity (rather than mountains) are of interest. In the case of measuring intensity, used throughout this dissertation, I will define image intensity as a function of image position,  $I(\vec{x})$ ,  $\vec{x} \in \mathbb{R}^l$ , and then consider the graph of the function as a kind of mountain range (Figure 1.8). Ridges then exist in this landscape of intensity analogously to their existence in mountains. In fact, one can find ridges of any function  $f$  in such a manner, by examining the graph of the function as a landscape, and in all of the following discussions I will assume that I am interested in finding a ridge of some



general function.

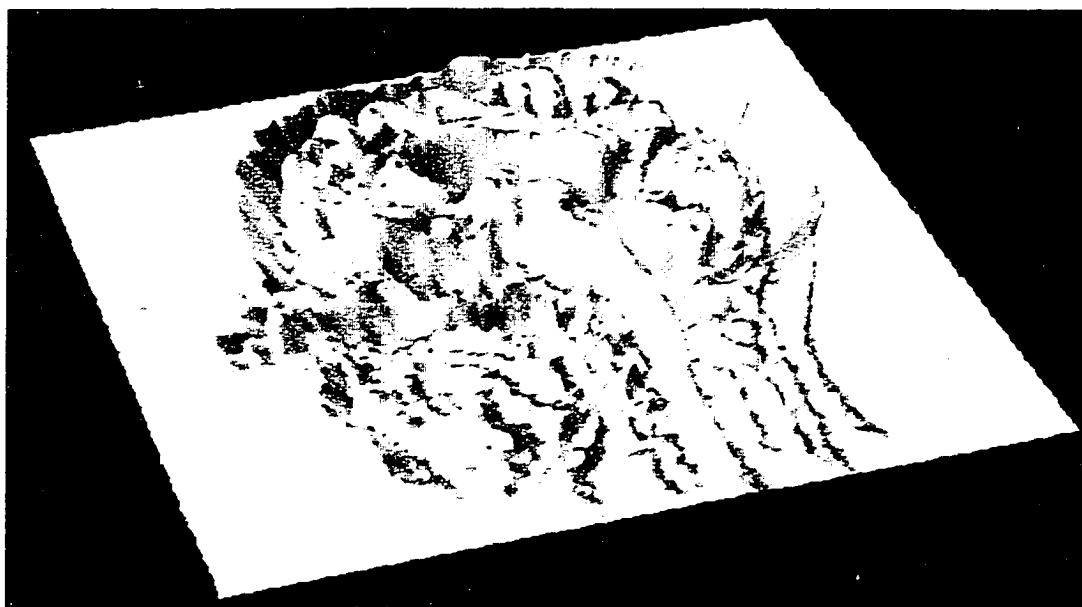


Figure 1.8: A greyscale image of a human head viewed as a height function of the image space.

Thus, ridges may be defined on functions whose domains are spaces of dimension greater than two (*e.g.*, 3D medical images). These ridges are not as easily described intuitively, although one can imagine a cloud of particles, with the densest part of the cloud being traced by a ridge (analogous to always walking on the highest part of the mountains). However, even three dimensions may be inadequate for certain problems, where other, possibly non-Euclidean, spaces are of interest. For example, if the boundary of an object is identified by a ridge, one may be interested in the normal to the boundary as well as the position of the boundary. This may involve finding ridges on a function whose domain is an *orientation space*, in which positional information is augmented by angular information. Another example involves medialness, which is a measurement made on position and radius. Ridges of medialness would then be defined in *scale spaces* in which positional information is augmented with width information. I combine orientation and scale space (Chapter 4) to produce *orientation scale spaces* which may have dimensions as high as eight. Intuition fails in trying to follow a curve through eight dimensional space.

Moreover, ridges may not just be curves. In 2D, these and relative maxima are the only ridges of interest. In 3D spaces, ridges may be surfaces, curves or points. The variety of manifolds which can be ridges increases as the dimensionality of the domain space of the function increases. These are also hard to describe intuitively, although the middle page of a book might help to think about a 2D ridge of medialness. Another way to imagine ridges in higher dimensional spaces is to break the space containing the ridge into two subspaces: one that follows the ridge and one that crosses over the ridge. (In the familiar example of a mountain ridge, Figure 1.9, each of these subspaces is 1D.)

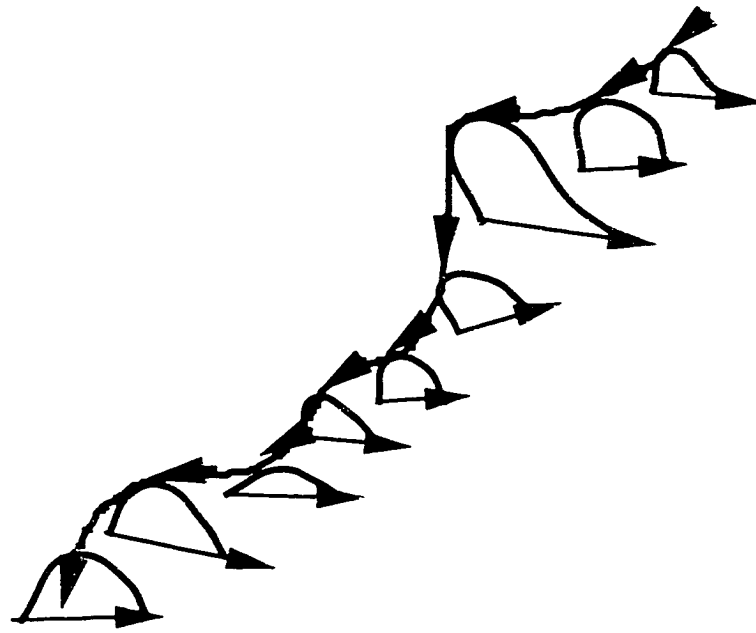


Figure 1.9: Following and crossing spaces of a mountain ridge.

The following space (the subspace that travels parallel to or along the ridge) may contain any behavior of the function, but the crossing space (the subspace that travels perpendicular to or across the ridge) must contain a relative maximum of the function restricted to the crossing space. This is called the *subdimensional maximum* property. It is this property that allows both mountain peaks and mountain passes to belong to the same ridge. This notion describes ridges as a collection of “cross-sectional” mountain peaks collected across the line orthogonal to the cross-sections.

My own work on ridges (Chapter 4) includes the definition for a particular kind of ridge, called an *optimal parameter ridge*, that allows for a natural choice of ridge crossing directions based on a specific image processing task and a reduction in dimensionality, so that, for example, a ridge defined in 6D may be defined in terms of a ridge in one 3D subspace and a relative maximum in the orthogonal 3D subspace. I will also discuss the algorithmic choices possible when calculating optimal parameter ridges. In conjunction with this is a proof that optimal parameter ridges preserve the subdimensional maximum property.

## 1.5 Locus-finding Algorithms

Given an image, it is often desirable to be able to locate a particular subset of the image domain that is important for some particular task. Landmarks, boundaries and skeletons all fall into the category of interesting subsets. Having made the decision to look for a particular subset, the next step is to then define an algorithm for finding that particular locus.

The most common method of finding these loci is manually; that is, having humans locate the subset of interest. As with any manual algorithm, there may be problems of speed and reliability. Thus there has been much research focused on the automatic and semi-automatic extraction of these subdimensional loci. To create any such algorithm, it is necessary to define mathematically the locus of interest. Thus, for example, a common way of identifying landmarks in an image is to define them as local maxima of intensity. Then, to find these loci, one designs an algorithm for finding maxima and inputs the image to the algorithm. Or, as another example, one may define a locus with derivatives and then use a differential equation solver to locate the set.

Of greatest interest to this dissertation is a class of algorithms designed to take advantage of the regularly sampled grid of pixels in a medical image to find loci. Also, in contrast to algorithms that search for the locus of interest in the entire image, these algorithms typically take advantage of the subdimensional and connected nature of a locus (for example, a boundary) to gain speed efficiency by following the locus through the image and thus not having to make any calculations where the locus is not. These algorithms are called *marching* or *tracking* algorithms because they march from one point on the locus to the next using the image grid as a guide. In Chapter 2 I will discuss the most relevant of these: those that can be adapted for finding ridges and thus cores. Chapter 5 will present my own marching algorithm, called *marching ridges*, designed to find ridges defined using the optimal parameter ridge definition.

## 1.6 Thesis

**Oriented medialness and optimal parameter height ridges employed in a marching ridges algorithm will effectively extract cores of 3D greyscale images.**

## 1.7 Contributions

This dissertation presents five contributions:

- a definition of Blum-like medialness, a medialness defined to be efficient and to avoid the effects of interfigural interference in the extraction of cores (Chapter 3).
- a definition of optimal parameter height ridges, designed to simplify ridge finding based on choices natural to a particular task (Chapter 4).
- a proof that optimal parameter height ridges have the subdimensional maximum property (Chapter 4).
- an algorithm to extract optimal parameter height ridges from 2D and 3D greyscale images (Chapter 5).
- extraction of 1D and 2D cores of 3D medical images (Chapter 6).

## 1.8 Organization

This document is organized as follows: Chapter 2 provides an introduction to relevant aspects of medical image analysis, including the use of weighting functions for measuring image properties such as boundariness or medialness, product spaces, and dimensionless derivatives in non-Euclidean product spaces; a basic definition of the Blum symmetric axis and its extension into 3D; a catalog of medialness measurements used in medical image processing; mathematical notions of ridges; and a description of three important locus-finding algorithms. Each of these subjects is helpful for the understanding of and critical to the success of my work. Chapter 3 includes a categorization of the many medialness measurements used in research at UNC-CH and the introduction of new definitions of medialness based on 1) extensions of current medialness measurements and 2) the mathematics of the Blum symmetric axis, specifically introducing the notion of multiple orientations into existing medialness measurements. The emphasis in the discussion of Blum-like medialness will be on 3D medialness measurements, although the work will be shown conceptually sound in 2D. Further, Chapter 3 presents a short discussion of the derivative and invariant nature of medialness kernels. Chapter 4 expands on the notion of distinguished directions in ridge definitions and presents the optimal parameter ridge. It also includes a proof of the subdimensional maximum property of optimal parameter ridges and discusses why optimal parameter ridges are the natural choice when defining cores. Chapter 5 presents an algorithm for finding height ridges, called *Marching Ridges*. I discuss the use of Marching Ridges for finding ridges of 2D images and for finding 1D and 2D ridges of 3D images. This chapter also includes a time analysis of the marching ridges algorithm, a catalog of possible pathological behaviors, and some heuristics employed in the calculation of ridges. Finally, Chapter 6 discusses the contributions of this dissertation including results of the ideas and algorithms on images and identifies directions for future work, including higher dimensional ridge tracking, more efficient and robust ridge tracking, and the use of connectors for shape analysis.

# Chapter 2

## Background

### 2.1 Introduction

This chapter considers six areas of importance to this dissertation and is organized as follows. First, I discuss the basics of medical image processing helpful in understanding the work in this and later chapters. Second, I provide a basic but rigorous introduction to the Blum symmetric axis, including its extension into 3D. Third, I list and discuss the various medialness measurement functions used at UNC-CH. Fourth, I provide a general definition of height ridges, followed by a list of more specific definitions and examples and the relation of each to this work. Finally, I discuss the different manifold-finding algorithms that directly contributed to this work.

### 2.2 Medical Image Analysis

Medical image analysis is an enormous field, covering a wide range of disciplines and applications. I have focused my work on the applications of medialness and height ridges to study the shape of objects in medical images. With sections following dedicated to medialness and height ridges, I will concentrate here on image analysis fundamentals used in this work.

#### 2.2.1 Weighting Functions and Convolution

A frequent task in medical image analysis is measuring an image for some quantity, such as boundary strength, intensity derivative, or medialness. This is often done by the specification of a measurement function that is then applied to the image. The mechanics of this involve a translation of the measurement function to the image location to be measured, followed by an integration over the domain space of the image of the product of the image and the measurement function.

If  $F(\bar{x})$ ,  $\bar{x} \in \mathbb{R}^l$  is the measurement function and  $I(\bar{x})$  the image to be measured, then the actual measured value at a particular location  $\bar{x}_0$  is

$$\int_{\mathbb{R}^l} F(\bar{y} - \bar{x}_0) I(\bar{y}) d\bar{y}$$

Since medical images are defined over finite domains,  $\Omega \subset \mathbb{R}^l$ , the integration is also defined only over the finite space  $\Omega$ . Further, since medical images are discretely sampled, the integration is implemented as a summation, so that the value at  $\bar{x}_0$  is

$$\sum_{\bar{y} \in \Omega} F(\bar{y} - \bar{x}_0) I(\bar{y})$$

Finally, because a smaller domain results in fewer multiplications and thus faster computations, the domain of the summation is frequently truncated to a subset of  $\Omega$ . This subset, which I will call the footprint of the weighting function  $F$ , is dependent on the kind of measurement function used and the measurement accuracy desired. As I will show in Chapter 5, one of the primary advantages of the medialness measurement functions I propose is their smaller footprint, allowing programs to run more quickly.

There are frequent cases in medical image processing in which the desired result of measurement is not a single value, but a new function describing the measurements taken at all locations in the image. This is accomplished by the application of the weighting function at all image locations.

$$f(\bar{x}) = \sum_{\bar{y} \in \Omega} F(\bar{y} - \bar{x}) I(\bar{y}) \quad (2.1)$$

Equation 2.1 contains three functions,  $f(\bar{x})$ ,  $F(\bar{x})$  and  $I(\bar{x})$ . To avoid ambiguity in my discussion of these functions, I will use the following conventions: 1) I will refer to  $I$  as an image, an intensity function or a measured function; 2) I will refer to  $F$  as a weighting function or a measurement function; and 3) I will refer to  $f$  simply as a function. I will also keep the capitalization consistent so that the weighting function will be an uppercase letter and the function resulting from the application of the weighting function to the entire image domain will be the corresponding lowercase letter.

Equation 2.1 can be restated as a convolution, defined as

$$f(\bar{x}) = \sum_{\bar{y} \in \Omega} F(\bar{x} - \bar{y}) I(\bar{y}) = F * I \quad (2.2)$$

in which  $\bar{F}(\bar{x}) = F(-\bar{x})$ . In this case,  $F$  is usually referred to as a convolution kernel. The work in later chapters of this dissertation will be described using weighting functions, while some of the material in this chapter will be described using convolution kernels.

### 2.2.2 Apertured Measurement Functions and Scale Space

One of the possible measurements on an image is a measurement of intensity derivatives. Unlike mathematics, the world of physical measurement rarely has the luxury of a closed form description of data (discrete or continuous) with the consequent ease of symbolic manipulation. Further there is no physical operator with which to acquire derivatives that can measure over the infinitesimally small footprint required to produce a description of instantaneous change. Even more difficult is the problem of dealing with discretely sampled data, in which the smallest footprint any operator can use for measuring derivatives is the distance between samples. Thus, image analysts are forced to deal with ill-posed differentiation. Florack [16] has shown that well-posed differentiation of a discrete signal is possible by convolving the signal with derivatives of the Gaussian (Figure 2.1)

$$G(\vec{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-\frac{|\vec{x}|^2}{2\sigma^2}} \quad (2.3)$$

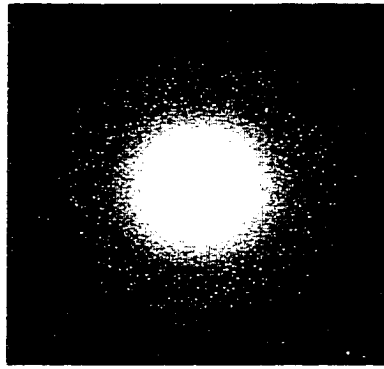


Figure 2.1: A 2D Gaussian.

The standard deviation of the Gaussian,  $\sigma$ , introduces a new parameter into the measurement of derivatives, called the scale (or aperture) of the measurement. Many image scientists have chosen to fix this scale at some predetermined  $\sigma_0$ , but more recently there has been much research that has  $\sigma$  vary continuously over all positive values. The result of the convolution of an image and a Gaussian derivative kernel at all possible scales is a function whose domain is the Cartesian product of position and scale, called a scale space. (In much of the image processing literature, the function produced from the convolution is called a scale-space representation. Sometimes it is simply called a scale space. I will avoid this latter use and only refer to the domain of the function as a scale space.)

$$g(\vec{x}, \sigma) = \sum_{\vec{y} \in \Omega} D(G(\vec{x} - \vec{y}, \sigma) I(\vec{y}))$$

in which  $D$  is an arbitrary derivative.

Because medialness is defined differentially, all medialness weighting functions will also include a scale component. The immediate consequence of this is that medialness functions have a domain whose dimension is at least 1 greater than the domain of the image they measure, explained below.

### 2.2.3 Generalized Product Spaces

Very often, the weighting function used for making image measurements will have a greater number of arguments than the intensity function. That is, the weighting function will be a function of the spatial variables of the image domain and auxiliary parameters. I use the word parameter in this dissertation to distinguish spatial variables from auxiliary variables; I do not intend to imply that auxiliary variables are in any sense fixed properties of the function. The scale example important in image processing was shown above. In another common example, measuring directional derivatives, the weighting function also includes parameters of orientation. The general term for the domains of each of these weighting functions is *product space*, as each is the Cartesian product of more than one space. Scale spaces can be described as  $\mathbb{R}^l \times \mathbb{R}_+$ , the product of the Euclidean space  $\mathbb{R}^l$  and the positive aperture space  $\mathbb{R}_+$ . Single orientation spaces can be described as  $\mathbb{R}^l \times S^{l-1}$ , the Euclidean space  $\mathbb{R}^l$  times the unit  $l - 1$  hypersphere. However, there is no reason why a product space must be restricted to a single scale, a single orientation or even the product of only two spaces. In Chapter 3, for example, I propose a medialness weighting function whose domain is the product space  $\mathbb{R}^l \times \mathbb{R}_+ \times S^{l-1} \times S^{l-1}$ . To define the domain of the function resulting from the convolution of an image and a measurement function, let  $l$  be the number of spatial variables of the image and  $\Omega \subset \mathbb{R}^l$  be the domain of the image. Further, let  $p$  be the number of additional parameters of the weighting function and  $A \subset \mathbb{R}^p$  be the space of those parameters. Then  $\mathbb{R}^l \times A$  is the domain space of a weighting function  $F$  and  $\Omega \times A$  is the domain of the function  $f$ , produced by the convolution of  $F$  and  $I$ . The relation of these spaces to each other and the image play an important role in Chapter 4.

### 2.2.4 Invariance to Similarity Transformations

Shape theory states that object shape must be invariant to similarity transformations: any combination of translation, rotation and uniform scaling. Thus, a shape does not change even though its position, orientation and size may change. This has the important consequence that any measurement function that attempts to quantify a shape property must also be invariant to those same transformations. Because medialness is a shape measurement, in Chapter 3, I will explore the consequences that invariance to similarity transformations has on the definition of medialness measurements.

If  $I(\bar{x})$  is an intensity function, and  $F(\bar{x})$  is a measurement function, and  $T : \mathbb{R}^l \rightarrow \mathbb{R}^l$  is a similarity transformation, then for  $F$  to be invariant to  $T$  requires that

$$\int_{\mathbb{R}^l} F(\bar{y} - \bar{x}) I(T(\bar{y})) d\bar{y} = \int_{\mathbb{R}^l} F(T(\bar{y}) - \bar{x}) I(\bar{y}) d\bar{y}$$



Medialness is an object shape measurement function so must also be invariant to similarity transformations. I will explore the effect of this in Chapter 3.

**Note:** Mathematicians and image scientists use the word *invariance* differently. To a mathematician, for a measurement function  $F$  to be invariant to a transformation  $T$  means that

$$\int_{\mathbb{R}^i} F(\tilde{y} - \tilde{x}) I(T(\tilde{y})) d\tilde{y} = \int_{\mathbb{R}^i} F(\tilde{y} - \tilde{x}) I(\tilde{y}) d\tilde{y}$$

What an image scientist calls invariant, a mathematician would call *equivariant*. Since this is an image processing dissertation, I will adopt the imaging convention and use the word invariant. Mathematicians reading this are to understand that I really mean equivariant.

## 2.3 Blum Symmetric Axis

The Blum symmetric axis is a skeleton for an object with a continuous, closed boundary. Mathematically, let an object in a two dimensional image be described by the set of maximal disks, where maximal disks are disks that fit inside the object but in no other disk. This maximality requirement ensures that the disks are tangent to the boundary of the object at one or more points. The symmetric axis is then the collection of the centers of these disks (Figure 2.2). This formalism has been extended for objects in three dimensional images by using maximal balls. The symmetric axis was first defined by Blum [3] and has been written about extensively both in the mathematics and computer science literature, most relevantly to this work by Nackman [36] for the case of 3D objects.

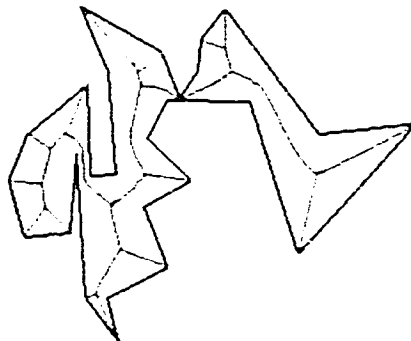


Figure 2.2: Example of a Blum symmetric axis (light grey) in 2D.

For 2D objects, the symmetric axis is generically a branching curve, although a disk's symmetric axis is a point. The curve and its branches end at the centers of disks with but a single point of osculated tangency. For 3D objects, the symmetric axis is generically a branching surface (Figure 2.3), although solid cylinders produce lines and balls produce

points. The symmetric axis transform, a bijective mapping between symmetric axes and object boundaries, can be found if one includes the radius of the disk or sphere as a function of position on the symmetric axis. (See Nackman [36] for details.)

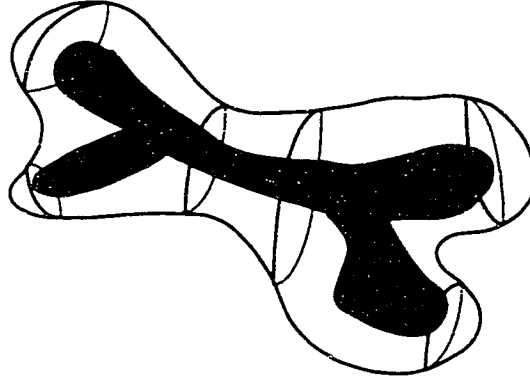


Figure 2.3: Example of a Blum symmetric axis (dark grey) in 3D.

The symmetric axis transform contains two key components for this dissertation. The first component is its linking of boundary points, called *involute*s, to identify the axis. The linked boundary points are the points on the boundary to which the disk (or ball) is tangent; they are associated through the center of this disk. This notion is the origin of medialness; I have refined it to produce Blum-like medialness weighting functions in Chapter 3. The second component is the identification between the “orientation” of maximal disks and the tangent to the symmetric axis at the center of the disks. If the orientation  $\theta$  of a maximal disk is defined as the average of the orientations of the vectors connecting the center of the maximal disk to the two boundary involutes and  $x(s)$  and  $y(s)$  are the coordinate functions of parameterized position along the axis, then

$$\frac{dy}{dx} = \tan \theta$$

This equation defines how the object is oriented at a particular point, by defining the tangent to the symmetric axis. This quantifies intuitive notions of how the object is turning as one progresses along the symmetric axis.

In the case of three dimensional objects, in which the symmetric axes are generically surfaces, Nackman established a similar identity between the “orientation” of maximal balls and the tangent plane of the symmetric axis at the center of the ball. As with the 2D case, the 3D identity quantifies intuitive notions of how the object bends along any direction in the tangent plane of the axis. I will explore the effect that this has on the calculation of optimal parameter ridges in Chapter 5.

In practical application, the Blum symmetric axis has two major drawbacks. First, calculation of the axis requires a closed boundary. This problem was alluded to in

Chapter 1 in the discussion of a circle as a medialness weighting function. Edge detection and boundary finding algorithms are notoriously difficult in medical images, especially in the presence of image disturbances such as noise. Second, even given such a boundary, the symmetric axis is extremely sensitive to small positional changes of the boundaries. Thus, a small pimple on the surface of an object will result in a completely new branch of the symmetric axis. Cores overcome these drawbacks by using medialness.

## 2.4 Medialness

Medialness is a measurement designed to produce medial axes that do not require boundaries as input and that are not sensitive to small boundary perturbations, thus bypassing the two major drawbacks of the symmetric axis. This allows a change in the paradigm for calculating medial shape descriptors from

- Find the boundary of an object in an image.
- Find the medial axis from the boundary.

to

- Find the medial axis of an object in an image.
- Find the boundary implied by the medial axis.
- Displace the boundary from that implied by the medial axis.

Medialness is what makes the first step of Pizer's paradigm [39] possible, while the second and third steps provide a much more stable way of identifying the boundaries of an object.

Thus, given a product space formed by product of the image domain and the range of possible widths of objects, a necessary condition for finding Pizer's medial axes is to be able to measure how similar locations in this product space (spatial positions and associated widths) are to the middles of objects in the image. Medialness is a function designed to quantify this property of image locations using the ideas of Blum and others: a point is most like a middle of an object when it is equidistant from two or more boundary points of that object. However, in the Blum tangent disk paradigm, the similarity of a point in an image to the middle of an object is a binary decision: a point either is or is not a middle point. This is a useful formalism when given binary object boundaries and instantaneous measurements. Unfortunately, most images delivered by physical imaging devices do not contain binary boundaries, and, as mentioned, scientists (as opposed to mathematicians) have no methods for making instantaneous measurements. Into this breach comes medialness. Medialness is a real-valued function of how like the middle of an object of a particular width any image location is. The aperture of the measurement function is proportional to the width of the object and allows the medialness weighting function to be insensitive to small boundary perturbations, where "small" is defined relative to the width of the object.

Applying the medialness weighting function to the entire domain of an image creates a medialness function: a function that provides a measurement of how medial every point in the original image is at every position and radius. If  $M$  is the medialness kernel and  $I$  is an image, then the medialness function  $m$  is defined by

$$m(\bar{x}, r) = \sum_{\bar{y} \in \Omega} M(\bar{x} - \bar{y}, r) I(\bar{y}) \quad (2.4)$$

In more recent practice, ridge finding is done only locally, and the entire medialness function is not calculated. Rather, single values of the medialness function are calculated by applying a medialness weighting function to the original image at isolated points.

### 2.4.1 Isotropic Medialness Weighting Functions

In its simplest form, a medialness weighting function can be typed as follows:

$$M(\bar{x}, r) : \mathbb{R}^l \times \mathbb{R}_+ \rightarrow \mathbb{R}$$

This is an isotropic medialness weighting function, one that has no orientation component. This describes the original medialness weighting functions developed at UNC-CH. Each of the three kinds of medialness weighting functions shown below can be easily generalized to an arbitrary dimension, although they have only actually been implemented in 2 and 3 dimensions.

#### Laplacian of Gaussian Medialness

Fritsch [18] used the negated Laplacian of a Gaussian (Figure 2.4) as a medialness weighting function. Fritsch originally used the Laplacian medialness weighting function for finding cores of objects in 2D medical images. Subsequently, it has been used by both Fritsch and Furst for finding cores of objects in 3D medical images. The general equation for Laplacian medialness is

$$M(\bar{x}, r) = -r^2 \Delta G(\bar{x}, r)$$

Since this medialness weighting function is non-zero at its center and has a large footprint, it responds strongly to a difference in average object and background intensities. This is analogous to the disk example presented in Chapter 1. It will respond maximally to an object that is a disk of the same radius as the measurement. In all other cases, it will respond maximally at a radius greater than the half width of the object for which it is measuring medialness.

#### Morse Medialness

Morse [35] created the Hough-like medial axis transform (HMAT) in an attempt to more closely model the symmetric axis and to reduce the effects of intensity in the interior of objects; the HMAT concentrates on the difference in object and background intensities

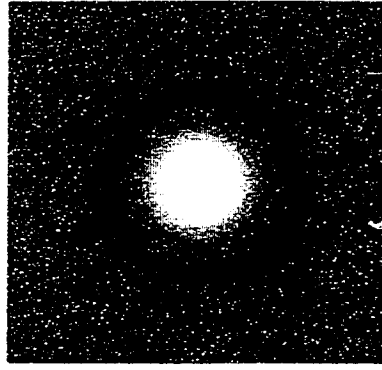


Figure 2.4: 2D Laplacian of Gaussian medialness weighting function.

at the boundary of the object, as the circle analogy describes in Chapter 1. The HMAT integrates boundariness measurements over a circle of radius proportional to the aperture of the boundariness measurement.

Define boundariness as a measurement of how like any point in an image is to the boundary of an object in the image. As with medialness, boundariness measurements must also be made with some measurement aperture. Additionally, a boundariness measurement is an oriented measurement, responding strongly normal to the boundary and weakly tangent to it. Thus, boundariness is a measurement of position, aperture and orientation:

$$B(\bar{x}, \sigma, \vec{u}) : \mathbb{R}^l \times \mathbb{R}_+ \times S^{l-1} \rightarrow \mathbb{R}$$

While there are many boundariness measurements that have been used, Morse defined his boundariness measurement as a directional first derivative of a Gaussian:

$$B(\bar{x}, \sigma, \vec{u}) = \sigma \vec{u} \cdot \nabla G(\bar{x}, \sigma)$$

and then defined medialness as an integration of the absolute value of boundariness:

$$M(\bar{x}, r) = \int_{S^{l-1}} |B(\bar{x} + r\vec{u}, \rho r, -\vec{u})| d\vec{u}$$

where  $\rho$  is the proportion between the standard deviation of the boundariness measurement function and the radius of the medialness weighting function and  $l$  is the dimension of the space containing  $\bar{x}$ . Morse originally created this weighting function in 2D (Figure 2.5); it was later adapted to 3D by Liu and then Furst. Because Morse implemented his core finding algorithms on entire images, he chose to calculate medialness using a convolution. To achieve this, he was required to remove the absolute value from his definition of medialness

$$M(\bar{x}, r) = \int_{S^{l-1}} B(\bar{x} + r\vec{u}, \rho r, -\vec{u}) d\vec{u}$$

As with the Laplacian medialness weighting function, the Morse measurement function will respond maximally to object boundaries at a radius greater than the half width of the object.

Eberly [14] has shown that as  $\rho$  approaches 1, the Morse measurement function approaches the Laplacian, and as  $\rho$  approaches 0, the Morse measurement function approaches the disks defining the Blum symmetric axis. Morse [35] typically used  $\rho = 0.25$ , which he chose as a result of experiments with human subjects.

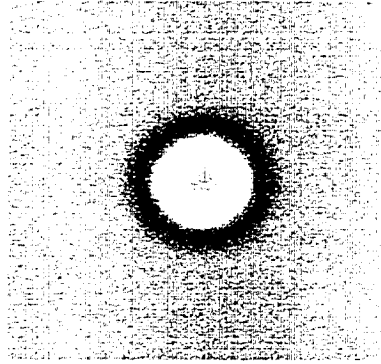


Figure 2.5: 2D Morse medialness weighting function.

### Fritsch Medialness

Because of the computational time involved in calculating the boundariness integral in the Morse medialness weighting function, Fritsch defined an approximation to Morse medialness that uses offset Gaussian derivatives in polar coordinates (Figure 2.6).

$$M(\vec{x}, r) = \frac{r - R}{\rho r} G(r - R, \rho r)$$

where  $R = |\vec{x}|$ .

This measurement function exhibits behavior and response to object boundaries very similar to that of the Morse measurement function, applies to any dimension, and is faster to compute. However, Eberly [14] has shown, in both 2D and 3D, closed form solutions to the linear form of the Morse medialness integral that are not much more computationally expensive than the Fritsch medialness.

### 2.4.2 Oriented Medialness Weighting Functions

As mentioned above, the isotropic medialness weighting functions proved unsatisfactory for making accurate width measurements of bar-like object in 2D images containing parallel sides. To overcome this obstacle, first Fritsch and then others devised medialness weighting functions that were not isotropic and had a preferred orientation:

$$M(\vec{x}, r, \vec{u}) : \mathbb{R}^l \times \mathbb{R}_+ \times S^{l-1} \rightarrow \mathbb{R}$$

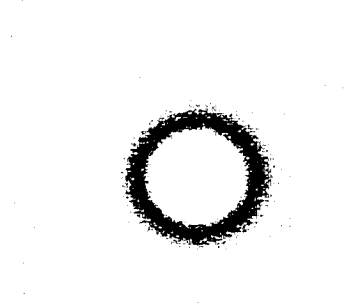


Figure 2.6: 2D Fritsch medialness weighting function.

In each of these medialness weighting functions, an isotropic medialness weighting function was given a single orientation component, changing the kernel so that it measures boundaries only at two antipodal points. As such, they are able to more accurately estimate the widths of bars.

### Oriented Laplacian Medialness

Fritsch took the Laplacian medialness weighting function and created the oriented Laplacian measurement function

$$M(\vec{x}, r, \vec{u}) = -r^2 D_{uu}(G(\vec{x}, r))$$

Fritsch called this measurement function  $L_{pp}$  medialness because in practice, he chose the direction  $\vec{u}$  by an eigensolution of the Hessian matrix of  $I$ , where  $\vec{u}$  was the eigenvector corresponding to the most negative eigenvalue. This direction is often labeled as  $\vec{p}$ . This choice maximizes the medialness weighting function over orientation and forms a foundation for the optimal parameter ridges discussed in Chapter 4. The oriented Laplacian measurement function has been used in both 2D (Figure 2.7) and 3D.



Figure 2.7: 2D  $L_{pp}$  medialness weighting function.

### Puff Medialness

Puff [40] extended Fritsch's idea of an oriented medialness weighting function to the Morse medialness weighting function and created a medialness weighting function that only integrated boundariness at two antipodal points of a circle of radius proportional to measurement aperture and used this to identify stenoses in angiograms. Similarly to the oriented Laplacian medialness, this measurement function is not radially symmetric: it also was designed for use with objects having parallel boundaries.

$$M(\bar{x}, r, \bar{u}) = B(\bar{x} + r\bar{u}, \rho r, -\bar{u}) + B(\bar{x} - r\bar{u}, \rho r, \bar{u})$$

where the boundariness measurement functions are the same as those used by Morse. Also, as did Fritsch, Puff maximized his medialness weighting function over orientation as part of the medialness calculation. Puff only implemented his measurement function in 2D (Figure 2.8). I will discuss 3D extensions of this medialness weighting function in Chapter 3.

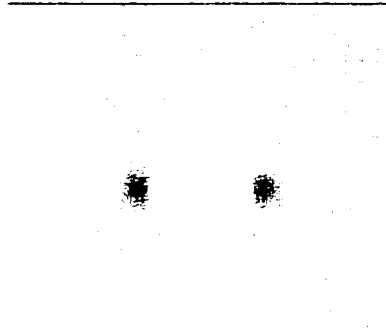


Figure 2.8: 2D Puff medialness weighting function.

### Oriented Fritsch Medialness

Because of the speed of calculating the Puff medialness weighting function, there has been no need for an oriented version of the Fritsch medialness weighting function in 2D. However, one of the possible extensions of the Puff medialness into 3D suffers from the same speed difficulties as the original Morse medialness, making an oriented Fritsch medialness weighting function useful in 3D. I will discuss this more fully in Chapter 3.

### 2.4.3 Other Medialness Weighting Functions

In addition to the medialness measures listed above, there are a number of other medialness weighting functions that have been defined at UNC-CH. I describe Clary endness and Low and McAuliffe medialness below.



### Clary Endness

Clary [9] has used a variant of medialness termed endness that is very similar to the Puff medialness except that it adds a third boundariness in a direction perpendicular to the orientation of the other two boundariness values (Figure 2.9). Thus, this measurement function responds to three points on the boundary and registers very highly at the ends of objects containing parallel boundaries.

$$M(\bar{x}, r, \bar{u}) = B(\bar{x} + r\bar{u}, \rho r, -\bar{u}) + B(\bar{x} - r\bar{u}, \rho r, \bar{u}) + B(\bar{x} + r\bar{u}^\perp, \rho r, -\bar{u}^\perp)$$

Clary is currently using the endness measurement function for 2D problems.

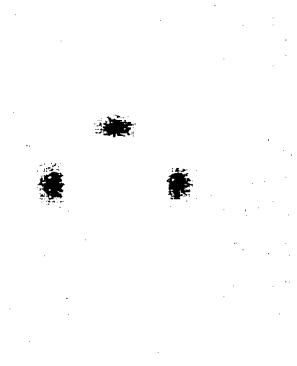


Figure 2.9: 2D Clary endness measurement function.

### Low Medialness

Low [31] has used a medialness weighting function that extends both the Puff medialness and the Clary endness by using up to ten orientation parameters to define points at which to calculate boundaries (Figure 2.10). He uses this medialness for applications involving deformations of shape models.

$$M(\bar{x}, r, \bar{u}_1 \dots \bar{u}_N) = \sum_{i=1}^N B(\bar{x} + r\bar{u}_i, \rho r, -\bar{u}_i)$$

where  $N \leq 10$ . As  $N \rightarrow \infty$ , the Low medialness weighting function approaches the Morse medialness weighting function without absolute value (Equation 2.4.1). Low used this medialness weighting function in 3D image processing tasks using a slice-based approach, effectively meaning the measurement function was 2D, although another orientation may be used to specify the slice of interest.

### McAuliffe Medialness

McAuliffe [33] produced a medial measurement function similar to the HMAT but with adaptive boundary detectors based on possibly nonisotropic Gaussians to reduce the ef-

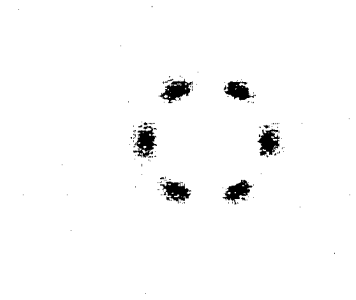


Figure 2.10: Low medialness weighting function with  $N = 6$ .

fects of neighboring objects on the calculation of medialness. The boundary measurement function that McAuliffe used was

$$B(\bar{x}, \bar{u}) = \max_{(\sigma_1, \sigma_2)} D_u(G(\bar{x}, \sigma_1, \sigma_2))$$

where  $\sigma_1^2 + \sigma_2^2 = 2(\rho r)^2$  and where

$$G(\bar{x}, \sigma_1, \sigma_2) = \frac{1}{(2\pi(\sigma_1^2 + \sigma_2^2))^{(d/2)}} e^{-\frac{x^2}{2\sigma_1^2} - \frac{y^2}{2\sigma_2^2}}$$

To keep computation fast, McAuliffe actually implemented his medialness weighting function as did Low: he chose a finite number of points evenly spaced around the area of Morse integration at which to calculate his adaptive boundariness. The McAuliffe boundariness measurement function as defined is 2D and has not been implemented in 3D.

## 2.5 Ridges

Given the number of different ways of describing ridges intuitively in Chapter 1, it is not surprising that mathematical definitions of ridges are at least as varied as the verbal ones. The many different definitions of ridges in the literature can generally be grouped into three different classes: ridges which are based on height, ridges which are based on curvature, and ridges which separate catch basins. Most of the examples provided in this section will be ridges in two dimensional spaces; a few are in three dimensional spaces. Similarly, most of the ridges in these examples are one dimensional ridges. However, this dissertation is interested in the full general notion of a ridge: a  $d$ -dimensional locus on an  $n$ -dimensional manifold, where the only requirement is that  $0 \leq d < n$ . Thus, the definitions presented here are fully generalizable, as will be the definition presented in Chapter 4.

Further, I am interested in ridges that can be defined locally, using differential ge-

ometric measurements. The first two classes of ridges, those which respond to height and curvature, can be so defined. Ridges which separate catch basins cannot be defined locally so are not used as a basis for this dissertation. Height ridges are defined on functions; as described intuitively in Chapter 1, the graph of a function can be viewed as a “landscape” on which to identify ridges. Curvature ridges, on the other hand, are defined on any manifold. In computer science, however, these manifolds are typically restricted to level sets of a function and graphs of functions viewed as Monge manifolds. In both cases, the curvatures of the resulting manifold can be described by functions, which leads to describing these ridges as height ridges of curvature functions. Therefore, in the remainder of this discussion, I will treat all differentially defined ridges as height ridges.

For the purposes of the following discussion of height ridges, I will start with an intensity function  $I$

$$I : \Omega \rightarrow \mathbb{R}$$

and create another function  $f$

$$f : \Omega \times A \rightarrow \mathbb{R}$$

via an application of a weighting function  $F$

$$F : \mathbb{R}^l \times A \rightarrow \mathbb{R}$$

to the entire image domain as described in Section 2.2.1.

In both  $f$  and  $F$  above,  $\Omega \subset \mathbb{R}^l$  denotes the spatial domain of the image  $I$  and  $A \subset \mathbb{R}^p$  denotes the domain of auxiliary parameters, if any, (e.g., scales and orientations) in the product space domains of  $f$  and  $F$ . As mentioned above,  $F$  may be a measurement function for image intensity, curvature, medialness, or any other image quantity.

Pizer [22] has derived a definition for height ridges that intuitively captures the notions of height, following directions and crossing directions described in Chapter 1.

**Definition 2.5.1** (*Height Ridges*) *A  $d$ -dimensional height ridge of a real-valued function  $f : \Omega \times A \rightarrow \mathbb{R}$  (or  $d$ -ridge of  $f$ ), is a  $d$ -dimensional locus in the ( $n$ -dimensional,  $n = l + p$ ,  $n > d$ ) domain of  $f$ . In general, the definition of this locus involves*

1. *a rule for choosing  $n - d$  linearly independent directions,  $\vec{v}^i$ , transverse to the putative ridge at a location  $(\vec{x}, \vec{a}) \in \Omega \times A$ , and*
2. *the requirement that  $f$  be maximized in each of the  $n - d$  one-dimensional subspaces determined by the  $\vec{v}^i$ . That is, the first derivative of  $f$  must vanish ( $f_{\vec{v}^i} = 0$ ) and the second derivative of  $f$  must be less than zero ( $f_{\vec{v}^i \vec{v}^i} < 0$ ) in each linear subspace.*

When searching for a  $d$ -dimensional ridge, there are  $n - d$  linearly independent directions transverse to the ridge. However, the definition of height ridges merely requires

transverse directions; it makes no specifications about the choice of directions. We therefore express the rule for choosing these transverse directions by  $\vec{v}^i(f, I)$ ,  $1 \leq i \leq n - d$ , making no requirements on the choice. As I will show in the following sections, the choice is highly dependent on the particular image processing task.

### 2.5.1 Maximum Convexity Height Ridges

The most natural height ridge is the maximum convexity ridge. By natural I mean that if there is no *a priori* knowledge about the particular task, the maximum convexity definition will supply transverse directions based only on properties of the function. Eberly [14] defines maximum convexity ridges as follows

**Definition 2.5.2** (*Maximum Convexity Height Ridges*) A  $d$ -dimensional maximum convexity height ridge of a function  $f \in C^2(\mathbb{R}^n, \mathbb{R})$  is the locus of points  $\bar{x} \in \mathbb{R}^n$  for which  $\lambda_{n-d} < 0$  and

$$Df \cdot \left[ \prod_{i=1}^d (D^2 f - \lambda_{n-d+i} I) \right] Df = 0$$

where the  $\lambda_i$  are the eigenvalues of the Hessian matrix of second derivatives of  $f$  and  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  and  $I$  is the  $n \times n$  identity matrix.

The maximum convexity height ridges uses as its choice of transverse directions the eigenvectors of  $D^2(f)$ . The rest of the definition then applies naturally to the definition of height ridges, where the eigenvalues are the second derivatives of the function in the transverse directions. The maximum convexity ridge exceeds the requirements of the height ridge to insure that every maximum convexity ridge point is a maximum in the entire space spanned by the transverse directions, not just in each direction independently. This desirable property is just a restatement of the subdimensional maximum property mentioned in Chapter 1.

Eberly [14] has written an excellent book on the subject of maximum convexity ridges; it is highly recommended reading for anyone interested in the nature of ridges. Damon [12] has done extensive work on the generic behavior of maximum convexity ridges of two dimensional functions and related manifolds and on the generic transitions which will occur in one parameter families. Keller [23, 27] and Miller [34] have produced similar results for ridges in higher dimensional spaces. Kalitzin [25] independently developed a height ridge definition which can be applied to any symmetric tensor. When that tensor is the Hessian matrix of second derivatives, his definition produces exactly the maximum convexity ridge.

The maximum convexity height ridge makes no assumptions about the image processing task; it relies solely on the differential structure of the function on which ridges are being calculated. However, many ridges have been defined for particular image analysis tasks. In the following section, I will describe various definitions of ridges for four distinct functions: intensity, boundariness, medialness, and curvatures. In each case, knowledge about the particular problem can influence the choice of transverse directions.

### 2.5.2 Height Ridges of Intensity

When searching for ridges of intensity, there is often no useful *a priori* knowledge of the problem at hand, and the most natural choice for choosing transverse directions is the maximum convexity definition. However, some researchers have made other choices as shown below.

#### Haralick Ridges

Haralick [24] describes the intuitive notion of a ridge in a digital image as a simply connected sequence of pixels having gray-tone intensity values which are significantly higher in the sequence than those neighboring the sequence. He defines a ridge as the collection of surface points which have a zero-crossing of the first directional derivative taken in the direction minimizing the second directional derivative:  $D_{\vec{r}} f = 0$  and  $D_{\vec{r}\vec{r}} f < 0$  where  $\vec{r}$  is the direction that minimizes  $D_{\vec{r}\vec{r}} f$ . This definition is the maximum convexity height ridge definition. To find ridges, he fits a bicubic polynomial to image data in  $\mathbb{R}^2$  and computes the one dimensional ridge of intensity on this bicubic. Thus, Haralick is finding a 1-dimensional ridge of a real-valued function  $f$  defined over  $\mathbb{R}^2$ , where there are no auxiliary parameters. The function  $f$  on which Haralick finds ridges is

$$f(x, y) = k_1 + k_2x + k_3y + k_4x^2 + k_5xy + k_6y^2 + k_7x^3 + k_8x^2y + k_9xy^2 + k_{10}y^3$$

where the  $k_i$  are the polynomial coefficients.

Since Haralick is finding 1D ridges in a 2D space, there is a unique ( $n - d = 2 - 1 = 1$ ) transverse direction. The rule for finding that direction is

$$\vec{r} = (\cos \alpha, \sin \alpha)$$

where Haralick approximates  $\alpha$  as

$$\alpha = \pm \tan^{-1}[k_5/(k_6 - k_4)]$$

Unfortunately, he found that with this definition (thus the maximum convexity definition also), non-pathological simple surfaces, such as radially symmetric surfaces  $f(x^2 + y^2)$ , satisfied this definition everywhere on the surface. Also, this definition, when generalized to 3D densities, produces ridges that are surfaces only; it is not broad enough to produce curves or points.

#### Crowley Ridges

Crowley [11] identifies ridge points as locations where difference of low pass (DOLP) impulse responses are a "best fit" to 2D image data. He defines ridge points in a DOLP transform as points for which any two opposite neighbors have smaller magnitudes or different signs than the point in question. This represents a choice of transverse directions that are coordinate axes. If either transverse direction produces a maximization of

intensity, then that point is a ridge. He then links these ridge points at bandpass levels and between bandpass levels to create a representation of shape. The representation thus defined is multiresolution (coarse-to-fine) and defined on grey scale images, two features it shares with this work. Also, this representation of shape produces a skeleton of an object, much like medial axes and cores, two other foundations of this work. Further, the DOLP representation can be used in image matching algorithms. However, unlike most other ridges, there are points on DOLP ridges that are clearly distinguished and can be used as landmarks in certain matching algorithms. Unfortunately, the identification of ridge points in the DOLP transform relies on the coordinate axes defined at a particular band pass level. This reliance on the coordinate axes to identify ridge points makes Crowley's ridge points fail to be rotationally invariant; that is, there may be points in the image that identify as ridge points for one orientation of the coordinate axes but not for another orientation.

### Lindeberg Ridges

Lindeberg [29] uses Haralick's definition of a ridge, which he describes as

$$\begin{cases} L_u & = 0 \\ L_{uu} & < 0 \\ |L_{vv}| \leq |L_{uu}| \end{cases}$$

or

$$\begin{cases} L_v & = 0 \\ L_{vv} & < 0 \\ |L_{uu}| \leq |L_{vv}| \end{cases}$$

where  $L_u = D_u G * I$  and  $u$  and  $v$  are the eigenvectors of the matrix of second derivatives  $D^2 G * I$ . Lindeberg uses this definition to find multiscale ridges by applying it for each scale, sweeping out a surface in scale space. At each spatial position, he then calculates a maximum with respect to scale of some ridge strength measurement, reducing the surface to a curve. In this sense, Lindeberg is first finding a 2D ridge of a function of two spatial variables and one scale parameter using the maximum convexity definition to identify one transverse direction, and then finding a 1D ridge from the 2D ridge by choosing the scale axis as another transverse direction. He shows visually compelling results with a variety of different images.

### 2.5.3 Height Ridges of Boundariness

Classically, the detection of edges has been treated separately from the detection of ridges. Pizer, however, takes the view that edges, as described in the literature, are frequently height ridges of boundariness. This idea is supported by the edge-finding algorithms described below. Further, while most rules for choosing transverse vectors will require only the function  $f$ , the case of height ridges of boundariness is one in which the gradient

direction in the original image  $I$  provides a natural orientation choice at each respective position. This is the reason that the rule for choosing transverse directions is stated as  $\vec{v}(f, I)$ ,  $1 \leq i \leq n - d$ .

As mentioned above in Section 2.4, boundariness is an image measurement made at a spatial position, with a particular orientation, and at some scale:  $B(\vec{x}, \hat{\theta}; \sigma_B)$ . Typically,  $\sigma_B$  is taken as a constant of  $B$ , but as with Lindeberg below, it can be a parameter of the function. Let  $b$  be the function produced from a convolution of the boundariness measurement  $B$  and the image  $I$ . We can then define a boundary, or edge, of an object as the projection of a ridge of  $b$  from the domain space of  $b$  onto the domain space of the image. The dimensionality of the boundary is one less than that of the image domain, i.e. boundaries are  $l - 1$  dimensional ridges of  $b$ .

### Canny Edges

Boundariness can be defined as the gradient magnitude of an image  $I$

$$B = |\nabla I(\vec{x}; \sigma_B)|$$

in which case boundariness is measured without orientation. If  $\sigma_B$  is taken as a constant, then  $b$  is defined on the same domain of the image, requiring a single transverse direction to define a ridge. The maximum convexity definition can be used for choosing this transverse direction. However, there is *a priori* information known about the image gradient: it will be roughly perpendicular to object boundaries defined by first derivatives, and this information can be used to choose the transverse direction as Canny [7] does. Canny defines an edge point as a local maximum in the  $\vec{u}$  direction of the measurement  $D_{\vec{u}}G$  applied to an image  $I$ , where  $\vec{u}$  is the smoothed gradient direction and  $D_{\vec{u}}G$  is the first directional derivative of the Gaussian in the  $u$  direction. Thus, a ridge point must satisfy

$$D_{uu}G * I = 0$$

and

$$D_{uuu}G * I < 0$$

where  $\vec{u} = \frac{\nabla G * I}{|\nabla G * I|}$  is the single transverse direction defined as the normalized gradient direction. Canny defines his edges only in the case of one and two dimensional signals, but they are readily extendible to three dimensions (with the assumption that an edge will always be a locus of dimension one less than the signal in which it is found). In this case, Canny's choice of transverse direction is simply the smoothed gradient direction  $\vec{u}$ . In [41], Qu uses a similar approach with three dimensional images. In both cases, Canny and Qu use the image  $I$  to choose transverse directions.

### Lindeberg Edges

Lindeberg [29] describes a multiscale approach to finding edges similar to his approach for multiscale intensity ridges. His definition is the same as Canny's:

$$\begin{cases} L_{uu} &= 0 \\ L_{uuu} &< 0 \end{cases}$$

where  $L$  is defined as before and  $u$  is the direction of the image gradient. Lindeberg applies this definition for each scale, again sweeping out a surface. He then finds local maxima with respect to scale of a measurement of edge strength, reducing the surface to a curve. Again, Lindeberg takes a two step approach: first finding a 2D ridge in a 3D space where one transverse direction is the gradient magnitude direction, and then finding a 1D ridge on the 2D ridge by choosing changing scale as the transverse direction.

### 2.5.4 Height Ridges of Medialness

Height ridges of medialness are, by definition, cores. Most of the research on cores has been done at UNC-CH, and a large body of work exists. Here I will present examples most relevant to the work in this dissertation.

#### Maximum Convexity Cores

Pizer, Morse [35] and Eberly all studied cores defined using the maximum convexity definition of height ridges. Morse and Eberly were able to show good results in identifying cores of 2D images, and thus, because medialness has an aperture, height ridges of a function of three variables. However, when Eberly and Furst tried to extend the work into 3D images, and thus medialness functions of four variables, both encountered problems related to tracking maximum convexity ridges in the higher dimensional space.

#### Optimal Scale Cores

Fritsch [18] modified the maximum convexity height ridge definition to produce optimal scale cores. At each spatial position of the medialness function, Fritsch identified local maxima through scale. The collection of these maxima produces implicitly defined manifolds in the medialness domain space of one less dimension than the domain. Fritsch called these manifolds *optimal scale manifolds*. He then applied the ridge definition to the medialness function restricted to the optimal scale manifold to produce cores. Note that the identification of local maxima through scale is equivalent to the conditions that  $D_r m = 0$  and  $D_{rr} m < 0$ ; that is, Fritsch has chosen scale as one of the transverse directions. The remaining transverse direction Fritsch chose using the maximum convexity definition of a new function

$$m(\bar{x}) = m(\bar{x}, \hat{P}(\bar{x}))$$

where



$$\bar{P}(\bar{x}) = \arg \max_r m(\bar{x}, r)$$

Fritsch used the chain rule to calculate the gradient and Hessian matrix of  $m$  in terms of derivatives of  $m$ .

It is useful to contrast Fritsch's definition of a ridge with either of Lindeberg's, in that Lindeberg does successive ridge finding on two separate functions in the order of space and then scale, while Fritsch performs successive ridge finding on two related functions in the order of scale and then space. As I will show in Chapter 4, Fritsch's is a more generally useful paradigm for calculating ridges.

### 2.5.5 Height Ridges of Curvature

There are three distinct kinds of curvature ridges:

- *crests*, or their one dimensional analog *vertices*, which respond to loci of high curvature on manifolds.
- *vertex ridges* formed from a family of crests of level sets of a function  $f$  parameterized by isocontour value, and
- *curvature ridges*, which are height ridges of the "curvature" of the graph of a function  $f$  using second derivative measurements to measure an analog to curvature.

#### Crests

Eberly [14] provides a definition of crests that includes crests of arbitrary dimension on a manifold of arbitrary dimension as follows:

**Definition 2.5.3 (Crests)** *A  $d$ -dimensional crest on an  $n$ -dimensional manifold,  $0 \leq d < n$ , is the locus of all points satisfying the conditions that*

$$\kappa_d > 0, v_i \cdot D\kappa_j = 0$$

and

$$v_i^t D^2 \kappa_j v_i < 0, 1 \leq i \leq n - d, 1 \leq j \leq d$$

where  $\kappa_1 \geq \kappa_2 \geq \dots \geq \kappa_n$  are the principal curvatures of the manifold and  $v_1, v_2, \dots, v_n$  the associated principal directions.

While this definition is applicable to any surface, including the graph of a function  $f$  viewed as a Monge manifold, attempts to use it on the graph of  $f$  run against the problem of mixing measures of height of a function and distance in the function's domain. With only a few exceptions (e.g., range data), height measurements of function values and distance in the function domain are incommensurate, and using the crest

definition on function graphs frequently requires untenable assumptions. This has lead some researchers to the following two curvature ridges.

### “Curvature” Ridges

One way to avoid the problem of incommensurability of measurements between a function graph and its domain is to use second derivatives to approximate the curvature of the graph. The definition of crests can then be modified to produce

**Definition 2.5.4** (*Maximum “Curvature” Ridges*) A  $d$ -dimensional maximum curvature ridge of a real-valued function  $f : \mathbb{R}^l \rightarrow \mathbb{R}$  is a  $d$ -dimensional locus of all points in the ( $n$ -dimensional,  $n > d$ ) domain of  $f$  satisfying the conditions that

$$\lambda_1 < 0, \vec{e}_1 \cdot D\lambda_1 = 0$$

and

$$\vec{e}_i^T D^2 \lambda_1 \vec{e}_i < 0, 1 \leq i \leq n - d$$

where  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  are the eigenvalues of  $D^2 f$  and the  $\vec{e}_i$  are the associated eigenvectors.

This definition avoids the problem of incommensurability but may identify ridges that are not what might be expected. Figure 2.11 show an example in which this definition will identify either dark line on the graph, instead of the intuitively more pleasing “top” of the graph. This problem leads to the final definition for curvature ridges.

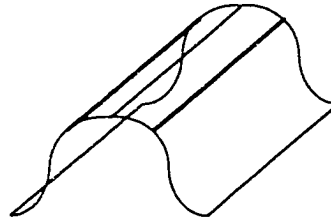


Figure 2.11: Failure of maximal curvature ridge to identify top of ridge

### Vertex Ridges

The definition of crests is limited to a single manifold and unsuitable for a function graph. However, the locus of crests formed by linking crests over all level sets (implicitly defined manifolds) of a function  $f$  is called a vertex ridge and does not suffer the problems of

crests, since level sets of a function are subsets of the function domain space, and all measurements there will be commensurate. If  $C_c$  is the set of crests defined on the level set  $f = c$ , then the vertex ridge  $V = \bigcup_{c \in \mathbb{R}} C_c$ . Note the result that if  $C_c$  is an  $d$  dimensional crest, then  $V$  is an  $d + 1$  dimensional vertex ridge.

Thirion [45] carefully describes the vertex ridges of functions of two and three variables, paying particular attention to the extraction of these vertex ridges in terms of partial derivatives of the function from which the level sets are derived. For functions of two variables, Thirion labels these loci as *maximal curvature curves* and for functions of three variables, he labels the loci as *crest surfaces*. Thirion goes on to show that the intersection of these vertex ridges with level sets produces crests, but the major mathematical work is in the definition of the vertex ridges. In  $\mathbb{R}^2$ , let  $\kappa(x, y)$  be the function of curvature of level curves, and let  $\vec{l}(x, y)$  be the tangent vector to level curves. Then  $\nabla \kappa(x, y) \cdot \vec{l} = 0$  is the condition Thirion uses for finding maximal curvature curves. The analogous result for three dimensional images is  $\nabla \kappa_1 \cdot \vec{l}_2 = 0$ , where  $\kappa_1$  is the first principal curvature of the level surfaces, and  $\vec{l}_2$  is the second tangent vector of the level surfaces. In our framework, the function  $f$  on which they are finding ridges is  $\nabla \kappa$ , where  $\kappa$  is the function of curvature of level curves in 2D and the first principal curvature of level surfaces in 3D. Their transverse directions are the tangent to level curves in 2D and the second principal direction for level surfaces in 3D.

Curiously enough, Thirion makes no requirements on  $\kappa(x, y)$  or  $D^2\kappa(x, y)$ , the Hessian matrix of second derivatives. Since he does not require that the second derivative in the transverse directions be negative, the maximal curvature curves and crest surfaces are supersets of ridges. This treatment of ridge supersets foreshadows connector sets, discussed in Chapter 6.

Fidrich [15] extends Thirion and Gourdon's work on crest lines into scale space by measuring for crest lines through scale and extends their ridge definition into scale space.

Maintz [32] is interested not in finding ridges explicitly but in measuring how like a ridge is each point in an image. This approach leads to ridge strength images, rather than ridge loci, although one can imagine finding height ridges of ridge strength measurements to find ridges. Maintz is interested in using these ridge strength images for the registration of images taken with different modalities. To do this, Maintz evaluates the effectiveness of two different ridge measurements:  $L_{\vec{v}\vec{v}}$ , the second derivative in the  $\vec{v}$  direction ( $D_{\vec{v}\vec{v}}G \star I$ ), and  $\frac{-L_{\vec{u}\vec{u}}}{L_{\vec{u}\vec{u}}}$ , the isophote curvature, where  $\vec{v} = (L_x, -L_y)$  and  $\vec{u} = (L_x, L_y)$  for some  $L : \mathbb{R}^2 \rightarrow \mathbb{R}$ . While he does not extract ridges, he defines them as the places where  $L_{\vec{v}\vec{v}}$  or  $\frac{-L_{\vec{u}\vec{u}}}{L_{\vec{u}\vec{u}}}$  is locally maximal in the  $\vec{v}$  direction; that is,  $D_{\vec{v}} \frac{-L_{\vec{u}\vec{u}}}{L_{\vec{u}\vec{u}}} = 0$  and  $D_{\vec{v}\vec{v}} \frac{-L_{\vec{u}\vec{u}}}{L_{\vec{u}\vec{u}}} < 0$ . Note that this is just the vertex ridge formed from the locus of crest points of all level curves of  $L$ .

Evaluations of these ridges, while natural in two dimensions, involve a nontrivial extension into three dimensions. Further, the definition of the vertex ridge involves fourth derivatives of the original intensity function. Finally, as with crests, vertex ridges may fail to identify what we intuitively think of as a ridge on a graph. However, Maintz reports good results in using both measurements and the resultant ridge strength images

for registering 2D and 3D medical images.

### 2.5.6 Subdimensional Maximum Property

The property of being a subdimensional maximum is an important property of any height ridge point. Without this property, there is no guarantee that a ridge point is in any sense a “high” point. Formally, let  $\mathcal{T}(\bar{x}, \bar{a})$  be a linear space spanned by a specific set of transverse directions of a ridge point of  $f$  at  $(\bar{x}, \bar{a})$  and passing through  $(\bar{x}, \bar{a})$ . Let  $\nabla_{\mathcal{T}} f|_{(\bar{x}, \bar{a})}$  be the gradient of  $f|_{\mathcal{T}}$  at  $(\bar{x}, \bar{a})$  and  $D_{\mathcal{T}}^2 f|_{(\bar{x}, \bar{a})}$  be the Hessian matrix of second derivatives with respect to an orthonormal basis of  $f|_{\mathcal{T}}$  at  $(\bar{x}, \bar{a})$ . Then, for the ridge point  $(\bar{x}, \bar{a})$  to have the subdimensional maximum property, the function  $f$  must be a local maximum at  $(\bar{x}, \bar{a})$  when restricted to the space  $\mathcal{T}$ . That is,  $\bar{u} \cdot \nabla_{\mathcal{T}} f|_{(\bar{x}, \bar{a})}$  must vanish and  $\bar{u}^T D_{\mathcal{T}}^2 f|_{(\bar{x}, \bar{a})} \bar{u}$  must be negative for all non-zero  $\bar{u} \in \mathcal{T}$ . As already mentioned, any ridge point of a maximum convexity ridge has the subdimensional maximum property. This follows from the choice of transverse directions as eigenvectors of the Hessian matrix of  $f$ . In Chapter 4 I will prove that any optimal parameter ridge point also has this property, making optimal parameter ridges an attractive choice for many image analysis tasks.

## 2.6 Locus-Finding Algorithms

Having defined ridges as level sets of first derivatives, I then need to locate them. The literature provides a rich set of algorithms designed to locate level sets, and the marching ridges algorithm presented in Chapter 5 is built on three of them: marching cubes, tracked and converged partitioning, and marching lines. All four algorithms share the common property of using the lattice of the function domain space to locate level sets of the function.

### 2.6.1 Marching Cubes

Lorensen [30] created an algorithm called *marching cubes* that creates triangle models of constant density surfaces from 3D medical data. If  $I$  is the image intensity and  $c$  is the constant, he finds an implicitly defined surface in a volume by trapping sign changes of the function  $I - c$  and uses linear interpolation to identify intersections of the surface and the Cartesian grid of the sample space. From this, he uses a case table to define the local triangle topology.

The basic marching cubes algorithm creates logical cubes from the original 3D image, where each vertex of the cube is a data point of the image. The algorithm then determines how the level surface intersects each of these cubes (Figure 2.12). Given a starting cube, the algorithm only intersects the surface with cubes adjacent to the starting cube if they share a face which the surface intersects. This is the marching part of the algorithm. This provides an enormous speed advantage over having to intersect the surface with all the cubes, since most of the cubes will not have any surface intersections.

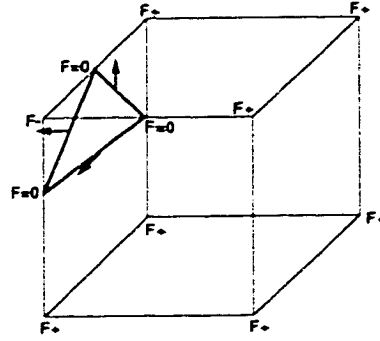


Figure 2.12: Diagram of a marching cube

The actual intersection is done by assigning a 1 (identified by  $F^+$  above) to a cube vertex if the image data at that point is higher than the level value and a 0 (identified by  $F^-$  above) if the image data is lower than the level value. This creates a cube with eight vertices labeled either 0 or 1. Of the 256 possible cases from this labeling, Lorensen reduced the number of patterns to 14 based on rotational symmetry and complementary cases where 0's are interchanged with 1's. Each of these 14 patterns is entered in an edge table, indicating which edges of the cube contain intersections with the surface. In half of these patterns, the intersections do not completely specify the topology of the surface in the cube. In these cases of topological ambiguity, Lorensen specified what he determined to be the most likely topology of the surface. When a cube's vertices have been labeled, the state of this labeling serves as a pointer into the edge table, and the intersections of the surface (identified by  $F = 0$  above) are linearly interpolated on the intersected edges. These points are joined into 1 to 4 triangles, each providing a linear approximation of the surface in that cube. Each face that contains a segment of a triangle indicates a neighboring cube (identified by the arrow) in which to search for surface intersections.

### 2.6.2 Tracked and Converged Partitioning

Bloomenthal [2] produced two variants of the marching cubes algorithm, called *tracked partitioning* and *converged partitioning*. While the tracked partitioning is very similar to the marching cubes algorithm, Bloomenthal added the notion of using simplexes (Figure 2.13) instead of the Cartesian grid to remove topological ambiguities. The geometry of simplexes allows an ordering of edge intersections that can produce only one triangulation. Bloomenthal also advocates adaptively subdividing cubes in which the intersecting surface is highly curved.

Bloomenthal's converged partitioning adds a coarse to fine component that speeds the operation considerably in areas where the surface is not changing much and is well suited to a multiscale approach. The converged partitioning also does not require a seed cube from which to start. However, small surface detail may be missed by a coarse cube, thus terminating subdivision early. Also, if the extent of the desired surface is not correctly estimated, it may be truncated.

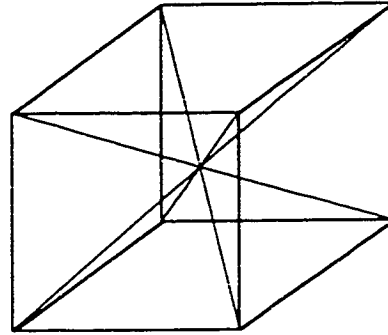


Figure 2.13: Marching cube divided into pyramidal simplices

### 2.6.3 Marching Lines

Thirion produced an improvement and an extension to the marching cubes algorithm called *marching lines*. In this algorithm, he uses cycles and segments to ensure sound topological properties of the extracted surface. Also with this algorithm, he was able to identify curves defined by the intersection of two implicit surfaces (Figure 2.14). The initial stages of this algorithm are similar to the marching cubes algorithm: it produces a polygonal approximation to one level set. However, the marching lines algorithm then traps zeroes of a second level set on the edges of this polygonal model, producing a piecewise linear representation of a curve. Thus, each voxel containing the implicitly defined curve produces a line segment. Any face of the voxel containing an endpoint of this line segment identifies a neighboring cube in which to search for subsequent line segments.

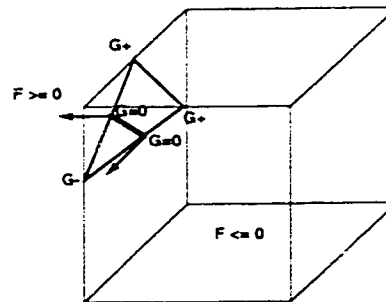


Figure 2.14: A marching line

This algorithm is ideally suited to the calculation of 1D ridges in a 3D ambient space, since such ridges are the intersection of two implicit surfaces themselves. This dissertation uses the techniques of Thirion for the purpose of finding cores of tubular objects in 3D: objects whose cores are curves.

# Chapter 3

## Medialness

### 3.1 Introduction

The ideas of medialness have been well established by the image processing community at UNC-CH and have been implemented in a large variety of image analysis tasks. This dissertation extends the work of those researchers in three ways. First, most of the prior work in medialness has been done in 2D. I present higher dimensional medialness measurements derived from the 2D measurements in current use. I describe these extensions first in a way that generalizes to any dimension, and then in the specific case of the 3D measurements used in the marching ridges algorithm to locate cores. Second, existing medialness measurements respond most strongly to non-generic image objects such as disks and tubes. I propose medialness measurement functions that mimic the action of the maximal disks and balls of the Blum symmetric axis and respond strongly to generic image objects. Finally, because medialness is a shape measurement, I discuss invariance to similarity transformations through a functional approach and show what it implies for the structure of medialness measurement functions and their derivatives.

### 3.2 New Medialness Weighting Functions

#### 3.2.1 Extensions to Existing Medialness Weighting Functions

The first medialness weighting functions designed at UNC-CH were isotropic measurements. They responded to boundaries equally in all directions from a center point. Puff, in his work using cores to identify stenoses in arteries, noticed that isotropic medialness measurement functions overestimated the width of the stenoses. In response, Fritsch designed an anisotropic medialness measurement: one that responded to boundaries only at two antipodal points of the measurement function. Intuitively, Fritsch divided the image domain space into two orthogonal subspaces such that his medialness measurement responded to boundaries in one space and did not respond to boundaries in the other

space. Fritsch's choice was a natural one to make in 2D and provided more accurate width estimates of objects resembling bars, with parallel sides. The same decision to divide the image domain space into a boundary response space and a no response space can be made for medialness measurements of any dimension, with the higher dimensional spaces offering more choices for the division.

To formalize this, let  $\mathbb{R}^l$  be the domain of the image to which the medialness weighting function will be applied. Divide this space into two orthogonal spaces,  $\mathbb{R}^\mu$  (the boundary response space) and  $\mathbb{R}^{l-\mu}$  (the no response space),  $\mu \leq l$ . Let  $U$  be an orthonormal set of vectors spanning  $\mathbb{R}^\mu$  and let  $\mathbf{u}$  be the matrix  $[\tilde{u}_1, \dots, \tilde{u}_\mu]$  whose columns are the elements of  $U$ . Let  $V$  be an orthonormal set of vectors spanning  $\mathbb{R}^{l-\mu}$  and let  $\mathbf{v}$  be the matrix  $[\tilde{v}_1, \dots, \tilde{v}_{l-\mu}]$  whose columns are the elements of  $V$ . Then, a  $\mu$ -oriented,  $l$ D medialness measurement is defined to be a weighting function that has a spatial domain of  $\mathbb{R}^l$  and that responds to boundaries only in the space spanned by  $U$ . In 2D images, therefore, the choices for  $\mu$  are 2, 1, and 0.  $\mu = 2$  defines an isotropic medialness measurement function that responds to boundaries in the entire domain of the image. The measurement functions defined in Section 2.4.1 are of this type.  $\mu = 1$  defines an anisotropic medialness measurement function that responds to boundaries in one linear subspace of the image domain but does not respond in the orthogonal subspace. The measurement functions defined in Section 2.4.2 are of this type.  $\mu = 0$  creates a medialness weighting function that does not respond to boundaries at all; since an essential aspect of medialness is the linking of boundary points,  $\mu = 0$  will be considered no further. Let  $l = 3$  (3D images), however, and the choices for  $\mu$  become 3, 2, and 1, while higher dimensional images provide even greater choice.

In general, the three major medialness measurements, Laplacian, Morse, and Fritsch, can be described for any  $\mu$  and  $l$  using  $\bar{x} \in \mathbb{R}^l$ ,  $r$ ,  $U$ ,  $\mathbf{u}$ ,  $V$ , and  $\mathbf{v}$ . Thus, the  $\mu$ -oriented,  $l$ D Laplacian medialness weighting function is defined as

$$M(\bar{x}, r, \mathbf{u}) = \left( \mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r} \right) G(\bar{x}, r) \quad (3.1)$$

the  $\mu$ -oriented,  $l$ D Morse medialness weighting function is defined as

$$M(\bar{x}, r, \mathbf{u}) = \int_{S^{l-1} \cap \langle U \rangle} |B(\bar{x} + r\vec{q}, \rho r, -\vec{q})| d\vec{q} \quad (3.2)$$

where  $\langle U \rangle$  denotes the subspace spanned by the vectors in  $U$ , and the  $\mu$ -oriented,  $l$ D Fritsch medialness weighting function is defined as

$$M(\bar{x}, r, \mathbf{u}) = \frac{(r - |\bar{x}\mathbf{u}|)}{\rho r} G(\sqrt{((r - |\bar{x}\mathbf{u}|)^2 + |\bar{x}\mathbf{v}|^2), \rho r}) \quad (3.3)$$

The medialness measurements defined in Sections 2.4.1 and 2.4.2 are specific examples of the above definitions. The medialness measurements defined in Section 2.4.3 are special cases of the above medialness measurements in which the weighting function does not respond to boundaries in the entire subspace  $\mathbb{R}^\mu$ . For example, Clary's endness is essentially a 2-oriented, 2D Morse medialness measurement that samples  $\mathbb{R}^\mu$  at three distinct points, while the Low measurement is a 2-oriented, 3D Morse medialness



measurement function that samples  $\mathbb{R}^n$  at up to 10 distinct points.

In the following sections, I will give specific examples of these general definitions (Equations 3.1, 3.2 and 3.3) for 3D medialness measurements used in marching ridges.

### Ball Response from Disk Response

The isotropic medialness measurement functions in 2D can be described as responding most accurately to disks. That is, they will estimate the radius of a disk better than the half width of any other object. Each has the property that  $\mu = l$ , and maintaining that property results in 3D medialness measurements that respond most accurately to balls. These are isotropic medialness weighting functions in 3D and do not require any special definition. The definitions of Section 2.4.1 are fully generalizable to any dimension.

### Tube Response from Disk Response

More interesting is the case in which the property  $\mu = 2$  is maintained when moving these 2D isotropic medialness measurement functions into 3D. This defines measurements that respond to boundaries in a plane, but don't respond in the direction normal to the plane. This generally describes a tube, in which any cross-section through the axis of the tube will resemble a disk. Let  $\mathbf{u} = [\vec{u}_1, \vec{u}_2]$  be a matrix of orthonormal vectors spanning the plane of boundary response, and let  $\mathbf{v} = [\vec{v}_1]$  be the matrix whose column is the direction in which there is no boundary response.

Then, the 2D isotropic Laplacian medialness weighting function becomes

$$M(\vec{x}, r, \vec{v}_1) = (2 - \frac{d^2}{r^2})G(\vec{x}, r) \quad (3.4)$$

where  $d = \sqrt{(\vec{x} \cdot \vec{u}_1)^2 + (\vec{x} \cdot \vec{u}_2)^2}$ , the 2D isotropic Morse medialness measurement becomes

$$M(\vec{x}, r, \vec{v}_1) = \int_{S^1} |B(\vec{x} + r\vec{q}, \rho r, -\vec{q})| d\vec{q}$$

where  $S^1$  is the space of unit vectors that are linear combinations of  $\vec{u}_1$  and  $\vec{u}_2$ , and the 2D isotropic Fritsch medialness measurement function becomes

$$M(\vec{x}, r, \vec{v}_1) = \frac{(r - d)}{\rho r} G(\sqrt{(r - d)^2 + (\vec{x} \cdot \vec{v}_1)^2}, \rho r)$$

where  $d$  is the same as was used in Equation 3.4.

All of these weighting functions can then be described as anisotropic, tube-response medialness measurements. Note that in each of these cases,  $M$  is defined as a measurement function of  $\vec{v}_1$  and not  $\mathbf{u}$ , as in Section 3.2.1. 3D has the special property that a single vector can specify a plane, so a more compact representation results by using  $\vec{v}_1$  instead of  $\mathbf{u}$ .

### Slab Response from Bar Response

The anisotropic medialness measurements in 2D can all be described as responding most accurately to bars. That is, with their antipodal points response, they most accurately estimate the half width of bar-like objects. Each of these has the property that  $\mu = 1$  and each is described in Section 2.4.2. Maintaining the property that  $\mu = 1$  when extending the medialness weighting functions into 3D produces measurements that still respond to boundaries at two antipodal points; however, in 3D this best describes a slab-like object with parallel sides. Let  $\mathbf{u} = [\vec{u}_1]$  be the matrix whose column describes the normals to the parallel sides, thus the linear subspace of boundary response, and let  $\mathbf{v} = [\vec{v}_1, \vec{v}_2]$  be the matrix whose orthonormal columns span the plane with no boundary response. In the case of both the Laplacian and Morse 2D anisotropic medialness measurements, the equations in 3D are identical to those given in Section 2.4.2, and both the Laplacian and the Morse slab-response medialness measurement functions have been used in the calculation of cores of slab-like objects. The 2D anisotropic Fritsch medialness, when extended to 3D, becomes

$$M(\vec{x}, r, \vec{u}_1) = \frac{r - \vec{x} \cdot \vec{u}_1}{\rho r} G(\sqrt{((r - \vec{x} \cdot \vec{u}_1)^2 + d^2)}, \rho r)$$

where  $d = \sqrt{(\vec{x} \cdot \vec{v}_1)^2 + (\vec{x} \cdot \vec{v}_2)^2}$ .

All of these measurements can then be described as anisotropic, slab-response medialness measurements.

### 3.2.2 Blum-like Medialness Measurement Functions

Each of the medialness measurements described above is defined in terms of mutually orthogonal, linear subspaces. The result of this is that they respond to specific object types: disks and bars in 2D, and balls, tubes and slabs in 3D. None is designed for optimal response to a wedge in 2D, for example, or a cone in 3D. That is, the parallel/perpendicular nature of their definitions produces responses to objects with the same parallel/perpendicular structure. I have defined a Blum-like medialness weighting function that has two orientations, such that the measurement can respond to any two points or integrate over any circle. Thus, these measurements will respond well to boundaries at any orientations to each other. Because of this, the Blum-like medialness measurement function has no isotropic form: it is inherently anisotropic. Further, the Blum-like medialness measurement has the potential to serve as its own endness measurement: at the end of a figure, the two points will come together at a single point or the circle of integration will shrink to a point. This corresponds to the endpoints of the Blum symmetric axis where the maximal disks and balls osculate the boundary of the figure. Finally, the Blum-like medialness, as with the oriented Morse medialness measurement, has a very small footprint when used as a weighting function and can be computed quickly. Initial results in 2D [37] have shown promise for this kind of medialness.

## 2D Measurements

The Laplacian, Morse, and Fritsch oriented medialness measurement functions can all be described as bar-response, since they respond to parallel boundaries. However, while the Blum-like medialness measurement can also respond this way, it is more general and cannot correctly be described as bar-response. I will borrow the terms *wedge*, *flare*, and *cup* from Blum [3] to describe the more general shapes for which the Blum-like medialness measurement is designed to respond (Figure 3.1). In 2D, this essentially describes any figure that has a linear extent; this underlies the appeal of the Blum-like medialness weighting function.

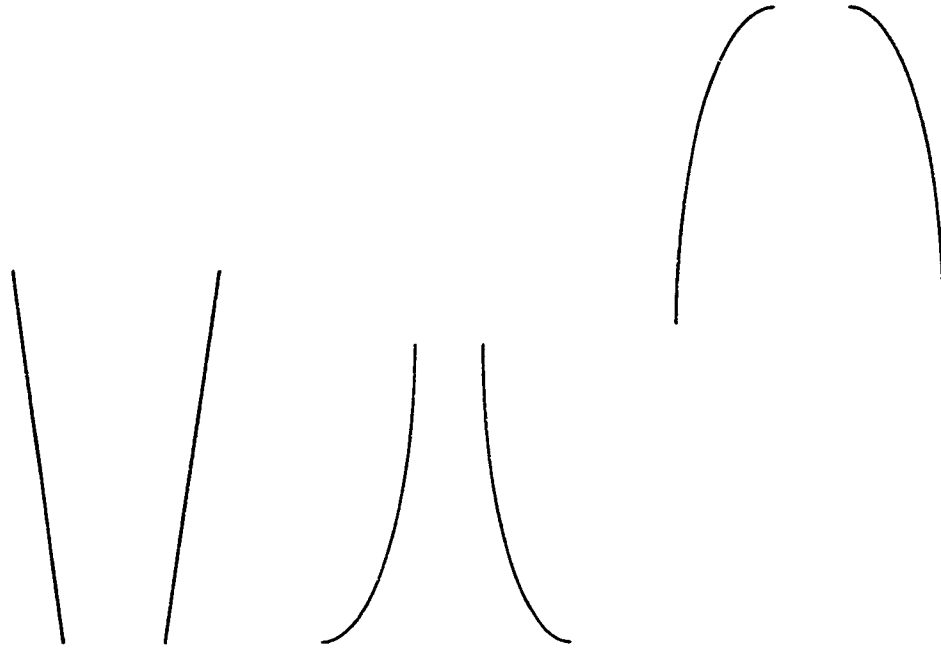


Figure 3.1: A wedge, a flare, and a cup

Given some boundariness function  $B$  defined as a measurement of position, aperture and orientation, the Blum-like medialness (Figure 3.2) is then defined in 2D as

$$M(\bar{x}, r, \vec{b}, \theta) = B(\bar{x} + rR(\theta)\vec{b}, \rho r, -R(\theta)\vec{b}) + B(\bar{x} - rR(\theta)\vec{b}, \rho r, R(\theta)\vec{b})$$

where  $R(\theta)$  is the rotation matrix describing a rotation by  $\theta$ .

The list of variables to the medialness measurement function are, in order, the spatial position of the origin of the weighting function ( $\bar{x}$ ), the length of each of the "arms" ( $r$ ), the bisector of the two arms ( $\vec{b}$ ), and the angle between the bisector and each arm ( $\theta$ ). I choose this particular representation because of the relationship between the medialness measurement and the ridges I will find on the resulting medialness function (Chapter 4).

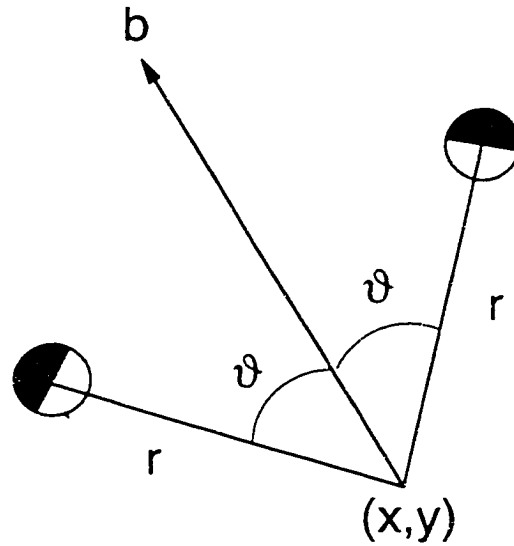


Figure 3.2: 2D Blum-like medialness measurement function.

### 3D Measurements

As with the the other oriented 3D medialness measurements, the 3D Blum-like medialness weighting function has two varieties: one to respond to tubes and the other to respond to slabs, although as with the case of the 2D Blum-like measurement function, the objects are actually more general. I will use the terms *cone*, *trumpet*, and *cap* to describe the tube-like objects. The slab-like objects can best be described as the direct product of any combination of two wedges, flares or cups (Figure 3.3).

When seeking a response for tubes-like objects, the Blum-like medialness weighting function is defined as

$$M(\bar{x}, r, \bar{b}, \theta) = \int_{S^1} B(\bar{x} + rR(\alpha, \theta)\bar{b}, \rho r, -R(\alpha, \theta)\bar{b})d\alpha$$

where  $R(\alpha, \theta)$  is the matrix describing the rotation of a point from the north pole of  $S^2$  to a point on the surface defined by the spherical angle  $(\alpha, \theta)$  along a great circle. This medialness measurement can be described as a cone which integrates boundariness values along its base, facing its terminus, and accumulates the integral at its terminus. Alternately, imagine rotating the 2D Blum-like measurement completely around  $\bar{b}$  to produce the tube response 3D Blum-like measurement. As before,  $\bar{x}$  defines the location of the terminus,  $r$  defines the distance along the outside of the cone from terminus to base,  $\bar{b}$  defines the direction of the central axis of the cone, and  $\theta$  defines the angular distance from the axis to the outside of the cone (Figure 3.4).

When seeking a response for slab-like objects, the Blum-like medialness measurement function is defined as

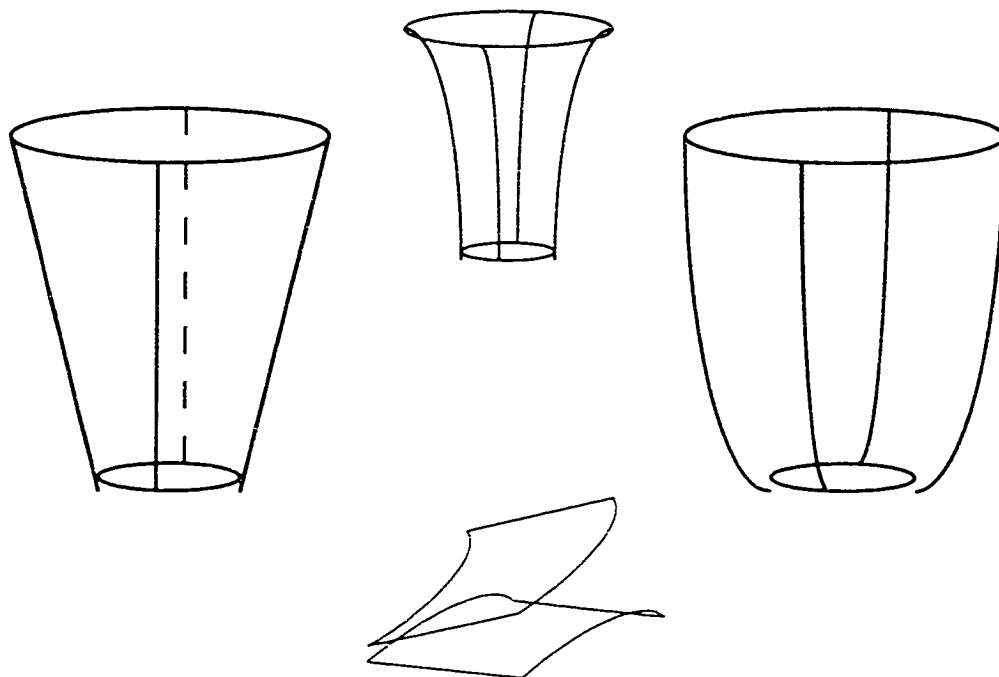


Figure 3.3: A cone, a trumpet, a cap, and the direct product of a wedge and a flare

$$M(\bar{x}, r, \bar{b}, \bar{\theta}) = B(\bar{x} + rR(\bar{\theta})\bar{b}, \rho r, -R(\bar{\theta})\bar{b}) + B(\bar{x} - rR(\bar{\theta})\bar{b}, \rho r, R(\bar{\theta})\bar{b})$$

where  $R(\bar{\theta})$  is the rotation matrix describing a rotation by the spherical angle  $\bar{\theta}$ . This medialness measurement is more easily described similarly to the 2D Blum-like medialness weighting function as a central point ( $\bar{x}$ ) with two arms of length ( $r$ ), bisected by the vector  $\bar{b}$  with  $\bar{\theta}$  describing the spherical angle from the bisector to the arms (Figure 3.5).

### 3.3 Invariance to Similarity Transforms

Shape theory requires that measurements of shape be invariant to similarity transformations (rotations, translation, and uniform scalings, also called zoom) of objects. Thus, medialness measurements, designed to measure shape properties, must also be invariant to these transformations. This treatment of invariance is more rigorous than that supplied in Section 2.2.4 and will show the consequence that invariance has on the design of medialness weighting functions.

#### 3.3.1 Functional Definition of Invariance

Let  $\mathcal{I}$  be a set of  $lD$ , scalar images:

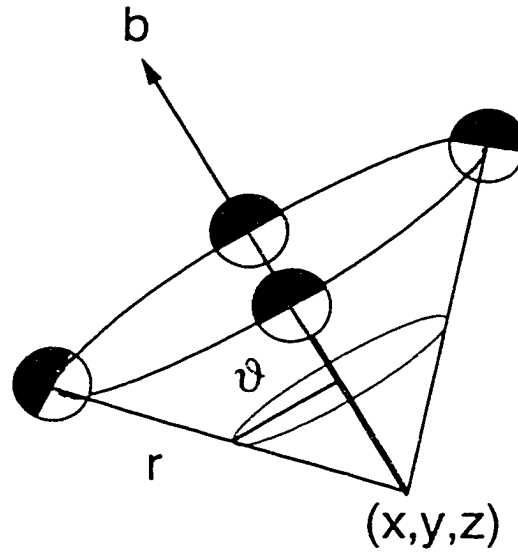


Figure 3.4: Blum-like medialness weighting function for tube-like objects

$$\mathcal{I} = \{I | I : \mathbb{R}^d \rightarrow \mathbb{R}\}$$

The set  $\mathcal{I}$  can have various properties appropriate to specific applications.

Let  $\mathcal{T}$  be a set of image transformations:

$$\mathcal{T} = \{T | T : \mathcal{I} \rightarrow \mathcal{I}\}$$

Examples of image transformations include all the similarity transformations, as well as such operations as contrast enhancement, convolution with a specific kernel and intensity scaling.

Let  $\mathcal{F}$  be a set of image measurement functions:

$$\mathcal{F} = \{F | F : \mathcal{I} \rightarrow \mathcal{I}'\}$$

The set  $\mathcal{F}$  is intuitively a set that accepts an image and produces another image, with  $\mathcal{I}'$  denoting a space image with possibly different properties than  $\mathcal{I}$ .  $\mathcal{T}$  still acts on  $\mathcal{I}'$ . Examples of image measurements include differentiation, boundariness, and medialness. Note that this typing is more general than that already given for medialness and boundariness. I use this function type here so that I am not restricted to talking only about weighting functions. The type given above is for a more general measurement function.

Given this, a particular measurement function  $F \in \mathcal{F}$  is invariant to a particular transformation  $T \in \mathcal{T}$  if and only if  $T \circ F = F \circ T$ . In other words, measuring an image and then transforming the measurement must produce the same result as measuring a transformed image. A classic example is the invariance of differentiation to translation.

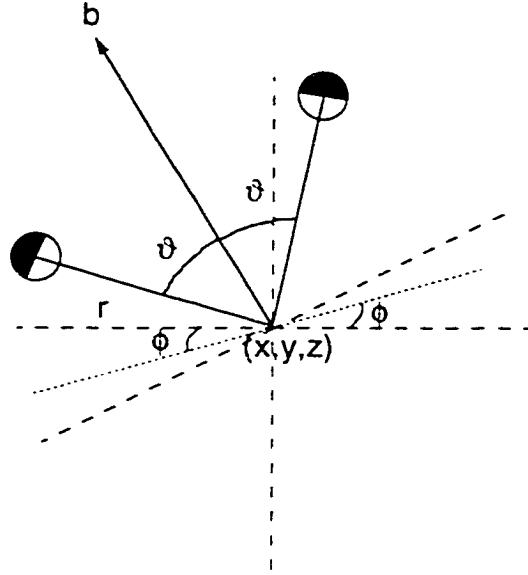


Figure 3.5: Blum-like medialness weighting function for slab-like objects

rotation, and uniform scalings. This is the heart of invariance: I will use this equality in the case of the application of weighting functions to show the requirements of those weighting functions is they are to be invariant to a particular transformation.

### 3.3.2 Similarity Transformations

Since similarity transformations are really not transformations of an image so much as transformations of the domain on which an image is defined, I will define another set of transformations as follows:

$$\mathcal{T}_X = \{T|T : \mathbb{R}^l \rightarrow \mathbb{R}^l\}$$

Examples of such coordinate transformations include such things as similarity transformations.

Specifically, a translation transformation  $T_t \in \mathcal{T}_X$  is defined as

$$T_t(\bar{x}) = \bar{x} + \bar{c} \quad (3.5)$$

a rotation transformation  $T_r \in \mathcal{T}_X$  is defined as

$$T_r(\bar{x}) = (\bar{x} - \bar{c})R(\bar{\theta}) + \bar{c} \quad (3.6)$$

where  $R(\bar{\theta})$  is the matrix describing the rotation of a point from the north pole of  $S^{l-1}$  to a point on  $S^{l-1}$  defined by the spherical angle  $(\bar{\theta})$  along a great circle and  $\bar{c}$  is the fixed point of the rotation, and a zoom translation  $T_z \in \mathcal{T}_X$  is defined as

$$T_z(\bar{x}) = m\bar{x} - \bar{c} \quad (3.7)$$

where  $\frac{\bar{c}}{m-1}$ ,  $m \neq 1$  is the fixed point of the zoom.

Invariance to these transformations is given by the requirement that  $F(I \circ T) = F(I) \circ T$ . As before, this requires that a measurement made on an image with a transformed coordinate system be the same as the measurement with a transformed coordinate system and describes the key invariance property of measurements. Likewise, I will use this equality in the specific context of weighting functions to derive the properties that weighting functions must have if they are to be invariant to certain transformations.

### 3.3.3 Weighting Functions

I will now return to a discussion of weighting functions, expanding on the material presented earlier, concentrating on weighting functions with an aperture (scale) parameter.

Let  $\mathcal{F}_W$  be a set of apertured weighting functions:

$$\mathcal{F}_W = \{F|F : \mathbb{R}^l \times \mathbb{R}_+ \rightarrow \mathbb{R}\}$$

and let  $f$  be any function defined as the application of the weighting function  $F$  to  $I$

$$f(\bar{x}, \sigma) = \int_{\mathbb{R}^l} F(\bar{y} - \bar{x}, \sigma) I(\bar{y}) d\bar{y}$$

Then, based on the invariance property already stated ( $F(I \circ T) = F(I) \circ T$ ), the definition of  $f$  as the application of the weighting function  $F$  to the image  $I$ , and the change of variables formula for multiple integrals,  $f$  is invariant to some coordinate transformation  $T \in \mathcal{T}_X$  if and only if  $\exists h : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  such that

$$\int_{\mathbb{R}^l} F(\bar{x} - \bar{y}, \sigma) I(T(\bar{y})) d\bar{y} = \int_{\mathbb{R}^l} F(T(\bar{x}) - \bar{y}, h(\sigma)) I(\bar{y}) d\bar{y} \quad (3.8)$$

Here, I will only show that for a certain kind of apertured weighting function, convolution is zoom invariant. Substituting Equation 3.7 into Equation 3.8 yields

$$\int_{\mathbb{R}^l} F(\bar{x} - \bar{y}, \sigma) I(m\bar{y} - \bar{c}) d\bar{y} = \int_{\mathbb{R}^l} F(m\bar{x} - \bar{c} - \bar{y}, h(\sigma)) I(\bar{y}) d\bar{y}$$

The coordinate transformation  $\bar{y} = \bar{z} + \bar{c}$  on the right side of the equality produces

$$\int_{\mathbb{R}^l} F(\bar{x} - \bar{y}, \sigma) I(m\bar{y} - \bar{c}) d\bar{y} = \int_{\mathbb{R}^l} F(m\bar{x} - \bar{c} - (m\bar{z} - \bar{c}), h(\sigma)) I(m\bar{z} - \bar{c}) m^l d\bar{z}$$

followed quickly by the coordinate transformation  $\bar{z} = \bar{y}$

$$\int_{\mathbb{R}^l} F(\bar{x} - \bar{y}, \sigma) I(m\bar{y} - \bar{c}) d\bar{y} = \int_{\mathbb{R}^l} m^l F(m(\bar{x} - \bar{y}), h(\sigma)) I(m\bar{y} - \bar{c}) d\bar{y}$$

implies that any zoom invariant weighting function must satisfy



$$F(\bar{x}, \sigma) = m^l F(m\bar{x}, h(\sigma)) \quad (3.9)$$

for some  $h$ .

### 3.3.4 Zoom Invariance of Medialness Measurements

Medialness weighting functions have aperture, so they must satisfy Equation 3.9. I will show, for the Laplacian, the Morse, the Fritsch, and the Blum-like medialness measurements defined in Sections 3.2.1 and 3.2.2, zoom invariance for  $h(r) = mr$ .

The Laplacian medialness measurement

$$M(\bar{x}, r, \mathbf{u}) = (\mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r})G(\bar{x}, r)$$

is a zoom-invariant measurement.

$$\begin{aligned} m^l M(m\bar{x}, h(r), \mathbf{u}) &= m^l M(m\bar{x}, mr, \mathbf{u}) \\ &= m^l (\mu - \frac{m\bar{x}\mathbf{u}}{mr} \cdot \frac{m\bar{x}\mathbf{u}}{mr})G(m\bar{x}, mr) \\ &= m^l (\mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r})G(m\bar{x}, mr) \\ &= (\mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r})m^l \frac{1}{(2\pi(mr)^2)^{(l/2)}} e^{(\frac{m\bar{x}\mathbf{u}}{2(mr)^2})} \\ &= (\mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r}) \frac{1}{(2\pi r^2)^{(l/2)}} e^{(\frac{\bar{x}\mathbf{u}}{2r^2})} \\ &= (\mu - \frac{\bar{x}\mathbf{u}}{r} \cdot \frac{\bar{x}\mathbf{u}}{r})G(\bar{x}, r) \\ &= M(\bar{x}, r, \mathbf{u}) \end{aligned}$$

The Morse medialness measurement

$$M(\bar{x}, r, \mathbf{u}) = \int_{S^{l-1} \cap \langle \mathbf{u} \rangle} |B(\bar{x} + r\vec{q}, \rho r, -\vec{q})| d\vec{q}$$

is also a zoom-invariant measurement.

Morse defined boundariness as

$$B(\bar{x}, \sigma, \vec{q}) = \sigma \vec{q} \cdot \nabla G(\bar{x}, \sigma)$$

so

$$\begin{aligned} B(m\bar{x} + mr\vec{q}, \rho mr, -\vec{q}) &= -m\sigma \vec{q} \cdot \nabla G(m\bar{x} + mr\vec{q}, \rho mr) \\ &= -m\sigma \vec{q} \cdot \nabla G(m(\bar{x} + r\vec{q}), \rho mr) \\ &= -m\sigma \vec{q} \cdot \nabla (\frac{1}{m^l} G(\bar{x} + r\vec{q}, \rho r)) \end{aligned}$$

$$\begin{aligned}
&= -\sigma \frac{1}{m^l} \vec{q} \cdot \nabla G(\vec{x} + r\vec{q}, \rho r) \\
&= \sigma \frac{1}{m^l} B(\vec{x} + r\vec{q}, \rho r, -\vec{q})
\end{aligned}$$

Using this information,

$$\begin{aligned}
m^l M(m\vec{x}, h(r), \mathbf{u}) &= m^l M(m\vec{x}, mr, \mathbf{u}) \\
&= m^l \int_{S^{l-1} \cap \langle \mathbf{u} \rangle} |B(m\vec{x} + mr\vec{q}, \rho mr, \vec{q})| d\vec{q} \\
&= m^l \int_{S^{l-1} \cap \langle \mathbf{u} \rangle} \left| \frac{1}{m^l} B(\vec{x} + r\vec{q}, \rho r, \vec{q}) \right| d\vec{q} \\
&= \int_{S^{l-1} \cap \langle \mathbf{u} \rangle} |B(\vec{x} + r\vec{q}, \rho r, \vec{q})| d\vec{q} \\
&= M(\vec{x}, r, \mathbf{u})
\end{aligned}$$

Because the Blum-like medialness measurements sum boundariness as does the Morse measurement, they are also zoom invariant for  $h(r) = mr$ .

Finally, the Fritsch kernel

$$M(\vec{x}, r, \mathbf{u}) = \frac{(r - |\vec{x}\mathbf{u}|)}{\rho r} G(\sqrt{((r - |\vec{x}\mathbf{u}|)^2 + |\vec{x}\mathbf{v}|^2)}, \rho r)$$

is also a zoom-invariant measurement.

$$\begin{aligned}
m^l M(m\vec{x}, h(r), \mathbf{u}) &= m^l M(m\vec{x}, mr, \mathbf{u}) \\
&= m^l \frac{(mr - |m\vec{x}\mathbf{u}|)}{\rho mr} G(\sqrt{((mr - |m\vec{x}\mathbf{u}|)^2 + |m\vec{x}\mathbf{v}|^2)}, \rho mr) \\
&= m^l \frac{(r - |\vec{x}\mathbf{u}|)}{\rho r} G(m\sqrt{((r - |\vec{x}\mathbf{u}|)^2 + |\vec{x}\mathbf{v}|^2)}, \rho mr) \\
&= m^l \frac{(r - |\vec{x}\mathbf{u}|)}{\rho r} \frac{1}{m^l} G(\sqrt{((r - |\vec{x}\mathbf{u}|)^2 + |\vec{x}\mathbf{v}|^2)}, \rho r) \\
&= \frac{(r - |\vec{x}\mathbf{u}|)}{\rho r} G(\sqrt{((r - |\vec{x}\mathbf{u}|)^2 + |\vec{x}\mathbf{v}|^2)}, \rho r) \\
&= M(\vec{x}, r, \mathbf{u})
\end{aligned}$$

### 3.3.5 Zoom Invariance of Medialness Derivatives

The height ridge definition requires first and second derivatives of the function on which the ridge is being found. Thus, in the calculation of cores, first and second derivatives of medialness measurements are used as weighting functions to supply the derivative quantities needed. As shape measurements, these derivatives also need to be zoom

invariant.

In following the proof for the zoom invariance of medialness functions given above, it becomes evident in each case that the zoom invariance of the medialness measurements is dependent on the zoom invariance of the Gaussian. So it is with the derivatives of medialness, and rather than show zoom invariant derivatives for each medialness measurement, I will instead do so for Gaussian derivatives. However, differentiation, while rotationally and translationally invariant, is not zoom invariant. To create a zoom invariant Gaussian derivative measurement of order  $k$  requires the premultiplication of the derivative measurement by  $r^k$ . The proof follows.

The Gaussian is

$$G(\bar{x}, r) = \frac{1}{(2\pi r^2)^{(t/2)}} e^{(\frac{\bar{x} \cdot \bar{x}}{2r^2})}$$

and the  $k^{th}$  derivative is

$$D^k G(\bar{x}, r) = \frac{1}{r^k} P^k\left(\frac{\bar{x}}{r}\right) G(\bar{x}, r)$$

where  $P^k(\frac{\bar{x}}{r})$  is a  $k$ -order polynomial in  $\frac{\bar{x}}{r}$ .

The zoom invariant derivative measurement of order  $k$  is

$$r^k D^k G(\bar{x}, r)$$

Thus, using  $h(r) = mr$ ,

$$\begin{aligned} m^t (mr)^k D^k G(m\bar{x}, mr) &= m^t m^k r^k \frac{1}{(mr)^k} P^k\left(\frac{m\bar{x}}{mr}\right) G(m\bar{x}, mr) \\ &= m^t r^k \frac{1}{r^k} P^k\left(\frac{\bar{x}}{r}\right) G(m\bar{x}, mr) \\ &= r^k \frac{1}{r^k} P^k\left(\frac{\bar{x}}{r}\right) G(\bar{x}, r) \\ &= r^k D^k G(\bar{x}, r) \end{aligned}$$

Because of this, the zoom-invariant medialness derivative measurements are all multiplied by  $r^k$ , where  $k$  is the order of differentiation. It is interesting to note that Eberly [13] derived the multiplication by  $r^k$  prior to this dissertation without a zoom-invariance argument. His was a dimensionless derivatives argument for the hyperbolic metric of scale space formed by the convolution with a Gaussian measurement. Further, Yoo [46] has shown that a premultiplication by  $(cr)^k$  does not alter the proof of zoom invariance.

### 3.4 Summary

This chapter presented a primary contribution in the definition of a Blum-like medialness operator that responds well to a large class of object shapes, is quick to compute, and has good potential for recognizing the ends of objects. This chapter also presents two secondary contributions in the general extension of the Laplacian, Morse and Fritsch medialness measurements into any dimension and any subspace of orientation and in the proof of the zoom invariance of these measurements and the reproof of Eberly's dimensionless derivatives by way of a zoom invariance argument. Chapter 4 will show how these medialness measurements can be used to create new kinds of cores.

# Chapter 4

## Optimal Parameter Height Ridges

### 4.1 Introduction

Having seen in Chapter 2 the many choices one can make for transverse directions in defining a height ridge, I expand here on Fritsch's notion of optimal scale ridges, in which he chose the direction of changing scale as a transverse direction. The expanded notion of an optimal scale ridge I call the *distinguished parameter height ridge*. A distinguished parameter height ridge of a function is any height ridge in which a non-empty subset of the coordinates axes of the function domain is chosen *a priori* to be a subset of the transverse directions. In the example of the Fritsch optimal scale ridge, the scale axis is chosen to be a single transverse direction. In this chapter I provide a definition of distinguished parameter height ridges. Following that, I discuss the choice of optimal parameters and remaining transverse directions as well as the mechanics of the reduction in dimensionality available with the distinguished parameter height ridge definition. In particular, I concentrate on and define the optimal parameter height ridge, produced from a specific choice of dimension reduction. Finally, I provide a proof of the subdimensional maximum property for optimal parameter height ridges.

### 4.2 Distinguished Parameter Height Ridges

A distinguished parameter height ridge of a function is a height ridge in which some set of coordinate axes of parameters of the function are chosen as transverse directions, leaving the remaining transverse directions to be chosen in any manner. This divides the set of transverse directions into two subsets: one containing coordinates axes of optimal parameters and the other containing directions chosen by some other rule. The distinguished parameter height ridge definition further requires that the function be maximized in each space spanned by the two subsets of transverse directions, rather than each transverse direction independently.

**Definition 4.2.1** (*Distinguished Parameter Height Ridges*) A  $d$ -dimensional optimal parameter height ridge of a real-valued function  $f : \mathbb{R}^l \times \mathbb{R}^p \rightarrow \mathbb{R}$  (or  $d$ -ridge of  $f$ ), is a  $d$ -dimensional locus in the ( $n$ -dimensional,  $n = l + p$ ,  $n > d$ ) domain of  $f$ . In general, the definition of this locus involves

1. a rule for choosing  $n-d$  linearly independent directions,  $\vec{v}$ , transverse to the putative ridge at a location  $(\bar{x}, \bar{a}) \in \mathbb{R}^l \times \mathbb{R}^p$  in which  $p$  of these transverse directions are coordinate axes of the domain of  $f$  spanning the space  $\mathcal{S} = \mathbb{R}^p$ , and
2. the requirement that  $f$  be maximized in  $\mathcal{S}$  as well as the space  $\mathcal{E} \subset \mathbb{R}^l$  spanned by the remaining  $e = (n - d) - s$  transverse directions. That is,  $\nabla_{\mathcal{S}}(f)$  and  $\nabla_{\mathcal{E}}(f)$  must vanish and  $D_{\mathcal{S}}^2(f)$  and  $D_{\mathcal{E}}^2(f)$  must both be negative definite, where  $\nabla_{\mathcal{S}}(f)$  is the gradient of  $f$  with respect to  $\mathcal{S}$ ,  $\nabla_{\mathcal{E}}(f)$  is the gradient of  $f$  with respect to  $\mathcal{E}$ ,  $D_{\mathcal{S}}^2(f)$  is the Hessian matrix of second derivatives of  $f$  with respect to  $\mathcal{S}$ , and  $D_{\mathcal{E}}^2(f)$  is the Hessian matrix of second derivatives of  $f$  with respect to  $\mathcal{E}$ .

This definition is different than the definition given by Furst at the 1997 Scale Space conference [23]. That definition requires only that  $f$  be maximized over each transverse direction in  $\mathcal{S}$  individually, rather than maximized over the entire space  $\mathcal{S}$ .

The set of distinguished parameter ridges is a subset of the set of height ridges. It is an interesting and useful subset to study because it allows the following two benefits:

- a natural choice of transverse directions for a particular task and
- a reduction of dimensionality and consequently, a simpler ridge finding algorithm.

The distinguished parameter ridge can be further restricted to the optimal parameter ridge, defined fully in Section 4.4.4. The optimal parameter ridge offers the following benefit:

- the subdimensional maximum property.

### 4.3 Natural Choice of Transverse Directions

The distinguished parameter height ridges provide a natural choice of transverse directions by requiring that certain transverse directions be “pure”: that is, only a single function parameter varies along that direction. This restriction can provide a clearer understanding of the behavior of the function along the ridge by isolating the cause for a particular change in the ridge. For example, in the case of Canny edges [7], orientation is chosen as a transverse direction, and this choice provides a normal to the edge at every point along the edge. This choice is also natural in the sense that it can separate parameters that respond to different aspects of the function. In the example of optimal scale cores,  $r$  is chosen as an distinguished parameter because it responds to object width, while the other variables respond to object position.

Another advantage to this natural choice of directions is a simplification in the calculation of derivatives. Eberly [14] has shown that a function domain that is a product space may have a non-Euclidean metric based on the definition of the function. The

result of this is complicated derivatives involving tensor calculus. While Eberly provides the mathematics for such derivatives, the task of finding an appropriate metric for a particular function domain is non-trivial. Further, the computational burden can be large. The distinguished parameter height ridge definition provides a way past this by allowing non-commensurate function parameters to be specified as optimal. In the special case where all the auxiliary parameters are chosen as optimal parameters, the derivatives in the remaining space ( $\mathbb{R}^l$ , the domain of the image) are simply Euclidean derivatives.

### 4.3.1 Choice of Distinguished Parameters

As shown in Chapter 2, there are often choices of transverse directions that are natural for particular tasks. In the case of Canny edges, orientation was a natural choice for one transverse direction since it matched intuitive notions of the image analysis task and could then be used to determine the second transverse direction. The calculation of cores is a task for which the natural choice for one transverse direction is changing radius. That is, regardless of spatial position, the radius parameter of the medialness weighting function should match the half width of the object for which cores are being found. Further, when using oriented medialness weighting functions, the weighting function should be oriented along the core of the object. These choices are also natural in that they provide intuitive measurements about the shape of an object such as turning and widening. Measurements of width and orientation should be distinct from each other and distinct from spatial position.

In general, when finding ridges of a function of spatial variables and auxiliary parameters,  $f(\bar{x}, \bar{a})$ , a natural choice for optimal parameters are the auxiliary parameters  $\bar{a}$ . This provides a clear separation between the action of the auxiliary parameters on the ridge and the actions of the spatial variables on the ridge. This separation is often intuitive and can provide a good understanding of object shape along the optimal parameter ridge. The result of having Euclidean derivatives as specified in Section 4.3 is another benefit associated with this choice.

### 4.3.2 Choice of Remaining Transverse Directions

Frequently, the natural choice for the remaining transverse directions is the maximal curvature choice, especially in the case when the auxiliary parameters have been chosen as the distinguished parameters. This choice guarantees that the ridge will have the property that the function is maximal when restricted to  $\mathcal{E}$ . However, in tasks involving both Euclidean and orientation components, choosing orientations as distinguished parameters can provide a more natural choice of transverse directions in the Euclidean space of the problem. In this paradigm, the distinguished orientations provide a set of directions based on the polar/spherical/cylindrical coordinate transformation between Euclidean and angular coordinates. This is the decision that Canny made: in this example, the direction is the gradient direction. The orientation component of oriented medialness measurement functions can be used similarly to specify transverse directions in  $\mathcal{E}$ . This

is one of the primary benefits of studying the mathematics of the Blum symmetric axis, in which the orientation of the medialness measurement function specifies an axis direction and thus a transverse direction perpendicular to it. In the special cases where a single orientation specifies a plane in the Euclidean space, the maximum convexity definition can be used to distinguish individual directions spanning the plane.

## 4.4 Reduction of Dimensionality

Having decided that certain parameters should be transverse directions, it is possible to reduce the dimensionality of the ridge finding problem by using optimal parameter manifolds, defined implicitly by vanishing gradients of  $f$  with respect to  $\mathcal{S}$  and negative definite Hessians of  $f$  with respect to  $\mathcal{S}$ . This creates an  $(n - s)$ -dimensional manifold  $\mathcal{M}$  in the domain space of  $f$ . This manifold is referred to as the optimal parameter manifold (e.g., optimal scale manifold). Finding ridges of  $f|_{\mathcal{M}}$  reduces the dimensionality of the ridge finding problem from  $n$  to  $n - s$ .

### 4.4.1 Optimal Parameter Hypersurfaces

I will begin the discussion of optimal parameter manifolds by using a single optimal parameter; following that, I will present the generalization to any number of optimal parameters.

Given a function  $f(\bar{x}, a)$ , in which  $a$  is chosen as an optimal parameter, generate an optimal parameter manifold  $\mathcal{M}$  implicitly using the following formulation:

$$\mathcal{M} = \{(\bar{x}, a) | D_a f(\bar{x}, a) = 0 \text{ and } D_{aa} f(\bar{x}, a) < 0\}$$

For  $\mathcal{M}$  to be a manifold, it is sufficient that  $\nabla(f_a) \neq \vec{0}$ , where  $\nabla(f_a)$  is the gradient of  $f_a$  with respect to its entire domain of spatial variables and auxiliary parameters. The definition of  $\mathcal{M}$  requires that  $f_{aa}(\bar{x}) < 0$  (implying that  $\nabla(f_a) \neq \vec{0}$  since at least the  $f_{aa}$  component does not vanish) and thus that  $\mathcal{M}$  is a manifold. Additionally, the condition  $f_{aa}(\bar{x}) < 0$  permits boundaries for these manifolds. As with any implicitly defined manifold,  $\mathcal{M}$  has dimensionality one less than the containing space  $\mathbb{R}^l \times \mathbb{R}^1$ . Thus, for example, if  $l = 2$ ,  $\mathcal{M}$  is a surface. This is the formulation which Fritsch used to generate his optimal parameter surfaces.

### 4.4.2 Optimal Parameter Manifolds

To generalize optimal parameter hypersurfaces to optimal parameter manifolds, I define a function  $f(\bar{x}, \bar{a})$  in which the parameters  $a_1, a_2, \dots, a_s$  are chosen as optimal parameters. Start by generating an implicitly defined set as follows:

$$\mathcal{N} = \{(\bar{x}, \bar{a}) | f_{a_i}(\bar{x}, \bar{a}) = 0 \text{ and } f_{a_i a_i}(\bar{x}, \bar{a}) < 0, 1 \leq i \leq s\}$$



Alternately, the definition may be stated as the intersection of  $s$  implicitly defined manifolds

$$\mathcal{N} = \bigcap_{i=1}^s \{(\bar{x}, \bar{a}) | f_{a_i}(\bar{x}, \bar{a}) = 0 \text{ and } f_{a_i, a_i}(\bar{x}, \bar{a}) < 0\}$$

However, the distinguished parameter ridge definition requires that points on the optimal parameter manifold be local maxima of  $f|_{\mathcal{S}}$ . Restricting the definition of  $\mathcal{N}$  produces the optimal parameter manifold  $\mathcal{M}$ .

$$\mathcal{M} = \{(\bar{x}, \bar{a}) | \nabla_{\mathcal{S}}(f)(\bar{x}, \bar{a}) = \vec{0} \text{ and } D_{\mathcal{S}}^2(f)(\bar{x}, \bar{a}) \text{ is negative definite}\}$$

where  $\nabla_{\mathcal{S}}(f)$  and  $D_{\mathcal{S}}^2(f)$  are the gradient and the Hessian matrix of  $f$  with respect to the space  $\mathcal{S}$ . The manifold  $\mathcal{M}$  is a subset of  $\mathcal{N}$ .

#### 4.4.3 Ridges on Optimal Parameter Manifolds

Once having made the choice of definition for an optimal parameter manifold, there are three possibilities for completing the other distinguished parameter ridge requirement that  $f|_{\mathcal{E}}$  be a local maximum.

- Calculate height ridges of  $f|_{\mathcal{M}}$
- Calculate height ridges of  $f$  on coordinate patches mapping  $\mathbb{R}^l$  to  $\mathcal{M}$
- Calculate height ridges of the projection of  $\mathcal{M}$  onto  $\mathbb{R}^l$

Each of the three choices produces a different ridge. Canny chose the first solution for his calculation of edges in the case of a two-dimensional image space and a single direction of orientation. Eberly [14] has shown how to do this in a space of arbitrary dimensions; it is computationally very expensive and has never been applied except in the case of Canny edges. Kalitzin has used the third method for segmenting structures in two dimensional grey scale images using a single dimension of orientation. He reports good results for a single test case using an optimal orientation manifold. Fritsch [20] uses the second method: I call this case the optimal parameter height ridge. The work in this dissertation is based on that method.

#### 4.4.4 Optimal Parameter Height Ridges

The optimal parameter height ridge requires a coordinate mapping from  $\mathbb{R}^l$  to  $\mathcal{M}$ . Define

$$\mathcal{P}(\bar{x}) = \arg \max_{\bar{a}} f(\bar{x}, \bar{a}) \tag{4.1}$$

in which the  $\arg \max_{\bar{a}}$  is the set of  $\bar{a}$ 's producing local maxima of  $f$  at the position  $\bar{x}$ . Let  $\bar{y} \in \mathbb{R}^l$  and let  $\bar{p} \in \mathcal{P}(\bar{y})$ . Let  $\bar{P}$  be a function that maps spatial positions in the neighborhood of  $\bar{y}$  to auxiliary parameters in the neighborhood of  $(\bar{y}, \bar{p})$  on  $\mathcal{M}$ .  $\bar{P}$  is then a local coordinate chart mapping  $\mathbb{R}^l$  to  $\mathcal{M}$ . This coordinate chart is always well defined except where the projection of the optimal parameter manifold is degenerate (e.g., folds)

with respect to  $\mathbb{R}^l$ . However, these folds occur only where  $H_{\mathcal{S}}(f)$  becomes singular: since the definition requires that  $H_{\mathcal{S}}(f)$  be negative definite, we can always define such coordinate charts.

Define another function

$$f(\tilde{x}) = f(\tilde{x}, \tilde{P}(\tilde{x})) \quad (4.2)$$

The final step of the optimal parameter ridge definition is to find height ridges of this new function  $\hat{f}$ . We are then faced with the decision about which transverse directions to use, depending on the problem as with the case of Canny edges. Chapter 5 will explore the decisions made in this dissertation.

We are now in a position to fully define the optimal parameter height ridge:

**Definition 4.4.1 (Optimal Parameter Height Ridges)** *A point  $(\tilde{x}, \tilde{a})$  is an optimal parameter height ridge point of  $f$  with respect to a space  $\mathcal{E} \subset \mathbb{R}^l$  spanned by a set of transverse directions if and only if  $f|_{\mathcal{E}}$  is locally maximal.*

#### 4.4.5 Calculating Derivatives of $\hat{f}$

The optimal parameter height ridge definition requires that  $\nabla_{\mathcal{E}}(\hat{f})$  vanish and that  $D_{\mathcal{E}}^2(\hat{f})$  be negative definite for any ridge point of  $f$ . However,  $\hat{f}$  is defined only in terms of  $f$  and  $\tilde{P}$ . Therefore, to calculate optimal parameter height ridges, I need derivatives of  $\hat{f}$  in terms of  $f$  and  $\tilde{P}$ . Further, since  $\tilde{P}$  has no closed form representation upon which to perform symbolic manipulation, I need derivatives of  $\tilde{P}$  in terms of  $f$ . This section provides derivatives of  $\hat{f}$  in terms of derivatives of  $f$  which can be used to complete the calculation of optimal parameter height ridges.

The solution for  $\nabla_{\mathcal{E}}(\hat{f})$  involves differentiating Equation 4.2 using the coordinate chart  $\tilde{P}$  (Equation 4.1) and the chain rule

$$\nabla_{\mathcal{E}}(\hat{f}) = \nabla_{\mathcal{E}}(f) + \nabla_{\mathcal{S}}(f) \nabla(\tilde{P}) \quad (4.3)$$

However,  $\nabla_{\mathcal{S}}(f)$  is known to vanish on ridge points by previous requirements of the optimal parameter ridge definition. Thus, Equation 4.3 simplifies to

$$\nabla(\hat{f}) = \nabla_{\mathcal{E}}(f) \quad (4.4)$$

This is also required to vanish for a point to be on the optimal parameter ridge.

The solution for  $D_{\mathcal{E}}^2(\hat{f})$  involves differentiating Equation 4.3 using the coordinate chart  $\tilde{P}$  (Equation 4.1) and the chain rule

$$\begin{aligned} D_{\mathcal{E}}^2(\hat{f}) &= D_{\mathcal{E}}^2(f) + \\ &\quad \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \nabla(\tilde{P}) + \\ &\quad (\nabla(\tilde{P}))' (\nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) + D_{\mathcal{S}}^2(f) \nabla(\tilde{P})) + \\ &\quad \nabla_{\mathcal{S}}(f) D^2(\tilde{P}) \end{aligned}$$

which can be simplified using  $\nabla_{\mathcal{S}}(f) = 0$  to

$$D_{\mathcal{E}}^2(f) = D_{\mathcal{E}}^2(f) + \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \nabla(\tilde{P}) + (\nabla(\tilde{P}))^t (\nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) + D_{\mathcal{S}}^2(f) \nabla(\tilde{P}))$$

which becomes

$$D_{\mathcal{E}}^2(f) = D_{\mathcal{E}}^2(f) + 2\nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \nabla(\tilde{P}) + (\nabla(\tilde{P}))^t D_{\mathcal{S}}^2(f) \nabla(\tilde{P}) \quad (4.5)$$

However, I need derivatives of  $f$  only in terms of  $f$ . Differentiating  $\nabla_{\mathcal{S}}(f) = 0$  with respect to  $\mathcal{E}$  produces

$$\nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) + D_{\mathcal{S}}^2(f) \nabla(\tilde{P}) = 0$$

which becomes, when solved for  $\nabla(\tilde{P})$

$$\nabla(\tilde{P}) = -(D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \quad (4.6)$$

Substituting Equation 4.6 into the right hand side of Equation 4.5 produces

$$\begin{aligned} D_{\mathcal{E}}^2(f) &= D_{\mathcal{E}}^2(f) - \\ &\quad 2\nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) + \\ &\quad \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) D_{\mathcal{S}}^2(f) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \end{aligned}$$

which then simplifies to

$$\begin{aligned} D_{\mathcal{E}}^2(f) &= D_{\mathcal{E}}^2(f) - \\ &\quad 2\nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) + \\ &\quad \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \end{aligned}$$

and then to

$$D_{\mathcal{E}}^2(f) = D_{\mathcal{E}}^2(f) - \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \quad (4.7)$$

This is required to be negative definite for a point to be on the optimal parameter height ridge.

I use these derivatives in this chapter to prove the subdimensional maximum property. In Chapter 5 I show how the marching ridges algorithm uses these derivatives to identify optimal parameter ridge points.

## 4.5 Proof of Subdimensional Maximum Property

As discussed in Chapter 2, it is desirable to have a ridge point of a function be a local maximum of the function restricted to the space spanned by the transverse directions of the ridge at that point rather than just along each direction independently. However, the space spanned by the transverse directions at an optimal parameter ridge point is split into two subspaces, with the only condition being that the function  $f$  is a local maximum in both subspaces. It might be that a point that is a maximum of a function restricted to two different subsets of the function's domain might not be a maximum of the function restricted to the product space of the two domain subsets. For example, in 2D, a thin saddle rotated by  $\pi/4$  from the x-axis would have a maximum at the origin along both coordinate axes, but it would not have a local maximum over the entire plane. In the case of optimal parameter ridges, however, I prove that ridge points are local maxima in the product space spanned by all the transverse directions.

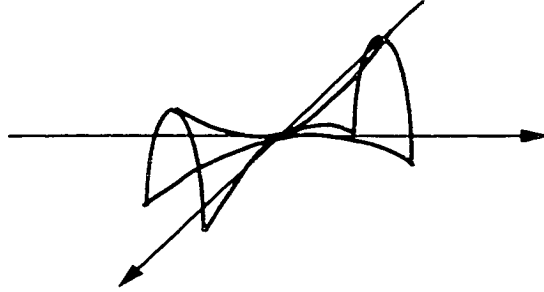


Figure 4.1: False identification of local maximum

**Theorem** For any optimal parameter ridge point  $(\bar{x}, \bar{a})$  of  $f$  as described above,  $f|_{\mathcal{E} \times \mathcal{S}}(\bar{x}, \bar{a})$  is locally maximal.

**Proof** I will divide the proof into two parts. The first part will show that  $\nabla_{\mathcal{E} \times \mathcal{S}}(f)$  vanishes, and the second will show that  $D_{\mathcal{E} \times \mathcal{S}}^2(f)$  is negative definite. Together, the two conditions assure that  $f|_{\mathcal{E} \times \mathcal{S}}$  is a local maximum.

To show the first part of the proof, let  $\bar{u} \in \mathcal{E}$ ,  $\bar{v} \in \mathcal{S}$ , and  $\bar{w} \in \mathcal{E} \times \mathcal{S} = \alpha \bar{u} + \beta \bar{v}$ . To show that  $\nabla_{\mathcal{E} \times \mathcal{S}}(f)$  vanishes, I need to show that  $\bar{w} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) = 0$ .

$$\begin{aligned} \bar{w} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) &= (\alpha \bar{u} + \beta \bar{v}) \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) \\ &= \alpha \bar{u} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) + \beta \bar{v} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) \end{aligned}$$

However, since  $\bar{u} \in \mathcal{E}$  and  $\bar{v} \in \mathcal{S}$ , this can be simplified to

$$\bar{w} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) = \alpha \bar{u} \cdot \nabla_{\mathcal{E}}(f) + \beta \bar{v} \cdot \nabla_{\mathcal{S}}(f)$$

And by Equation 4.4

$$\vec{w} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) = \alpha \vec{u} \cdot \nabla_{\mathcal{E}}(f) + \beta \vec{v} \cdot \nabla_{\mathcal{S}}(f)$$

Since both  $\nabla_{\mathcal{E}}(f)$  and  $\nabla_{\mathcal{S}}(f)$  vanish by definition at any point on an optimal parameter height ridge,  $\vec{w} \cdot \nabla_{\mathcal{E} \times \mathcal{S}}(f) = 0$ , completing the first section of the proof.

To show the second part of the proof, let  $\vec{w}$  be as before. To show that  $D_{\mathcal{E} \times \mathcal{S}}^2(f)$  is negative definite, I need to show that  $\vec{w} D_{\mathcal{E} \times \mathcal{S}}^2(f) \vec{w}^t < 0$ .

$$\begin{aligned} \vec{w} D_{\mathcal{E} \times \mathcal{S}}^2(f) \vec{w}^t &= (\alpha \vec{u} + \beta \vec{v}) D_{\mathcal{E} \times \mathcal{S}}^2(f) (\alpha \vec{u} + \beta \vec{v})^t \\ &= \alpha \vec{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \alpha \vec{u}^t + \\ &\quad \alpha \vec{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \beta \vec{v}^t + \\ &\quad \beta \vec{v} D_{\mathcal{E} \times \mathcal{S}}^2(f) \alpha \vec{u}^t + \\ &\quad \beta \vec{v} D_{\mathcal{E} \times \mathcal{S}}^2(f) \beta \vec{v}^t \end{aligned}$$

and, because  $\vec{u} \in \mathcal{E}$  and  $\vec{v} \in \mathcal{S}$

$$\begin{aligned} \vec{w} D_{\mathcal{E} \times \mathcal{S}}^2(f) \vec{w}^t &= \alpha \vec{u} D_{\mathcal{E}}^2(f) \alpha \vec{u}^t + \\ &\quad \alpha \vec{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \beta \vec{v}^t + \\ &\quad \beta \vec{v} D_{\mathcal{E} \times \mathcal{S}}^2(f) \alpha \vec{u}^t + \\ &\quad \beta \vec{v} D_{\mathcal{S}}^2(f) \beta \vec{v}^t \end{aligned}$$

Further, because  $D_{\mathcal{E} \times \mathcal{S}}^2(f)$  includes only the second partial derivatives of  $f$  once with respect to  $\mathcal{E}$  and once with respect to  $\mathcal{S}$ , I write it more succinctly as  $\nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f))$  or  $\nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f))$

$$\begin{aligned} \vec{w} D_{\mathcal{E} \times \mathcal{S}}^2(f) \vec{w}^t &= \alpha \vec{u} D_{\mathcal{E}}^2(f) \alpha \vec{u}^t + \\ &\quad \alpha \vec{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \beta \vec{v}^t + \\ &\quad \beta \vec{v} \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \vec{u}^t + \\ &\quad \beta \vec{v} D_{\mathcal{S}}^2(f) \beta \vec{v}^t \end{aligned}$$

By definition, at any point of the optimal parameter height ridge,  $D_{\mathcal{E}}^2(f)$  is negative definite, and thus  $\alpha \vec{u}^t D_{\mathcal{E}}^2(f) \alpha \vec{u} < 0$ . Subtracting this quantity from the right side of the equation insures a greater value than on the left.

$$\begin{aligned} \vec{w} D_{\mathcal{E} \times \mathcal{S}}^2(f) \vec{w}^t &< \alpha \vec{u} D_{\mathcal{E}}^2(f) \alpha \vec{u}^t - \\ &\quad \alpha \vec{u} D_{\mathcal{E}}^2(f) \alpha \vec{u}^t + \end{aligned}$$

$$\begin{aligned}
& \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \beta \bar{v}^t + \\
& \beta \bar{v} \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) \beta \bar{v}^t \\
< & \alpha \bar{u} (D_{\mathcal{E}}^2(f) - D_{\mathcal{E}}^2(f)) \alpha \bar{u}^t + \\
& \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \beta \bar{v}^t + \\
& \beta \bar{v} \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) \beta \bar{v}^t
\end{aligned}$$

A slight alteration of Equation 4.7 produces

$$D_{\mathcal{E}}^2(f) - D_{\mathcal{E}}^2(f) = \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f))$$

which substituted into the right hand side of Equation 4.8 yields

$$\bar{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \bar{u}^t < \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \quad (4.8)$$

$$\alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) \beta \bar{v}^t + \quad (4.9)$$

$$\beta \bar{v} \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \quad (4.10)$$

$$\beta \bar{v} D_{\mathcal{S}}^2(f) \beta \bar{v}^t \quad (4.11)$$

The introduction of  $(D_{\mathcal{S}}^2)^{-1}(f) D_{\mathcal{S}}^2(f)$  and its inverse leaves the right hand side unchanged

$$\begin{aligned}
\bar{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \bar{u}^t & < \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \\
& \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) D_{\mathcal{S}}^2(f) \beta \bar{v}^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) (D_{\mathcal{S}}^2)^{-1}(f) \nabla_{\mathcal{E}}(\nabla_{\mathcal{S}}(f)) \alpha \bar{u}^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) (D_{\mathcal{S}}^2)^{-1}(f) D_{\mathcal{S}}^2(f) \beta \bar{v}^t \\
& < \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) (\alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)))^t + \\
& \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) (D_{\mathcal{S}}^2)^{-1}(f) (\beta \bar{v} D_{\mathcal{S}}^2(f))^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) (D_{\mathcal{S}}^2)^{-1}(f) (\alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)))^t + \\
& \beta \bar{v} D_{\mathcal{S}}^2(f) (D_{\mathcal{S}}^2)^{-1}(f) (\beta \bar{v} D_{\mathcal{S}}^2(f))^t \\
& < \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) + \beta \bar{v} D_{\mathcal{S}}^2(f) \\
& (D_{\mathcal{S}}^2)^{-1}(f) \\
& (\alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) + \beta \bar{v} D_{\mathcal{S}}^2(f))^t
\end{aligned}$$

Since  $D_{\mathcal{S}}^2(f)$  is negative definite, by definition, at any optimal parameter ridge point,  $(D_{\mathcal{S}}^2)^{-1}(f)$  is also. Thus,

$$\begin{aligned}
0 &> \alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) + \beta \bar{v} D_{\mathcal{S}}^2(f) \\
&\quad (D_{\mathcal{S}}^2)^{-1}(f) \\
&\quad (\alpha \bar{u} \nabla_{\mathcal{S}}(\nabla_{\mathcal{E}}(f)) + \beta \bar{v} D_{\mathcal{S}}^2(f))'
\end{aligned}$$

and by the transitive property of inequality.

$$\bar{u} D_{\mathcal{E} \times \mathcal{S}}^2(f) \bar{u}^t < 0$$

This completes the second part of the proof. Thus, any point on an optimal parameter ridge has the subdimensional maximum property that  $f|_{\mathcal{E} \times \mathcal{S}}$  is a local maximum.

## 4.6 Summary

This chapter provided a definition for optimal parameter height ridges, listed the benefits of these ridges, and provided intuitive and mathematical support for each of the benefits. I will use this information in the marching ridges algorithm (Chapter 5) to calculate optimal scale and orientation cores of objects in 3D medical images.

# Chapter 5

## Marching Ridges

### 5.1 Introduction

This chapter presents an algorithm for identifying height ridges called *Marching Ridges* derived from earlier work on finding cores [22]. It is designed as a general purpose algorithm that can solve for height ridges of any measurement derived from an image; examples include intensity, boundariness, and medialness. The majority of this chapter is independent of the choice of function. However, parts of this chapter and the results presented in Chapter 6 are specific to medialness functions and optimal parameter height ridges. The Marching Ridges algorithm does not deal with the limiting cases of 0- or  $n$ -dimensional ridges. 0D ridges are local maxima of a function and, as isolated points, require no tracking, rendering the Marching Ridges algorithm superfluous and inefficient. Similarly, an  $n$ D ridge of a function of  $n$  variables is simply the domain of that function and requires no special algorithm to identify.

As mentioned in Chapter 2, Marching Ridges is a method of finding piecewise linear approximations to ridges by using the lattice structure of the domain in which the ridge is contained. The marching also occurs within the lattice structure of the domain by identifying hypercubes adjoining the hypercube containing the ridge from shared hyperfaces which contain a part of the ridge. This chapter is organized to first introduce the interface for Marching Ridges and then describe the algorithm for calculating ridges. The algorithm is divided into ridge finding strategies, both general (not depending on the codimension of the ridge) and specific (depending on the codimension of the ridge), and marching strategies, also both general (not depending on the dimension of the ridge) and specific (depending on the dimension of the ridge). Following this is a brief discussion of the time and space complexity of the Marching Ridges algorithm.



## 5.2 Grid elements

Marching Ridges is organized around a regular partition of the  $n$ -dimensional product space in which the ridge is to be found. The subdivision of the product space into unit  $n$ -cubes produces *grid elements*. I use the term grid element in preference to dimension specific terms such as pixel and voxel for two reasons. First, Marching Ridges has been designed for the general purpose of finding  $d$ -dimensional ridges in an  $n$ -dimensional function domain space, and the use of grid elements allows a dimensionally neutral discussion of the algorithm. Second, a grid element has vertices which are sample points of the function in its domain space rather than the more common notion of pixels and voxels, in which the pixel or voxel is centered at a sample point of the function. Thus, the grid element of a 2D space would contain four sample points, one at each vertex of a square, and the grid element would reference four function values rather than the single value associated with a pixel. I will also refer to *subelements*: lower-dimensional components of grid elements. For example, grid elements of a 3D space are cubes, while the various subelements are six faces (containing four sample points), 12 edges (containing two sample points) and eight points (containing a single sample point). I will further use the term *border element* to distinguish the largest subelement that grid elements can share with each other. This will be a subelement of dimension one less than the grid element; for example, each border element of a cubic grid element is a square element.

## 5.3 Interface

The interface (Figure 5.1) to the Marching Ridges algorithm is a window containing a central image canvas 512 pixels square surrounded by sliders and buttons to provide parameters and instructions to the algorithm. The list of buttons along the upper left side of the interface provides overall control of the window. The set of three buttons below that determines the function on which the algorithm will find ridges. The two numeric buttons below determine the dimensionality of the ridge. The two buttons at the bottom provide the user a choice of maximization techniques for optimal parameter ridges. The six sliders on the right side of the window provide parameters to the program. The set of three buttons below the sliders determines what, if any, parameters of the measurement function will be optimized. Finally, the two buttons at the bottom of the window allow the user to choose white features on a black background or black features on a white background. The controls set and maintain four important quantities:

- the number of optimal parameters  $s$ ,
- the dimension of the grid elements  $g = n - s$ ,
- the dimension of the ridge  $d$ , and
- the codimension of the ridge with respect to the dimension of the grid elements  $c = g - d$ .

Each of these quantities is initially 0 but will change as the user sets the parameters of the ridge-finding task by selecting certain buttons. As I describe each set of buttons below, I will indicate how particular choices affect the values of  $s$ ,  $g$ ,  $d$  and  $c$ .

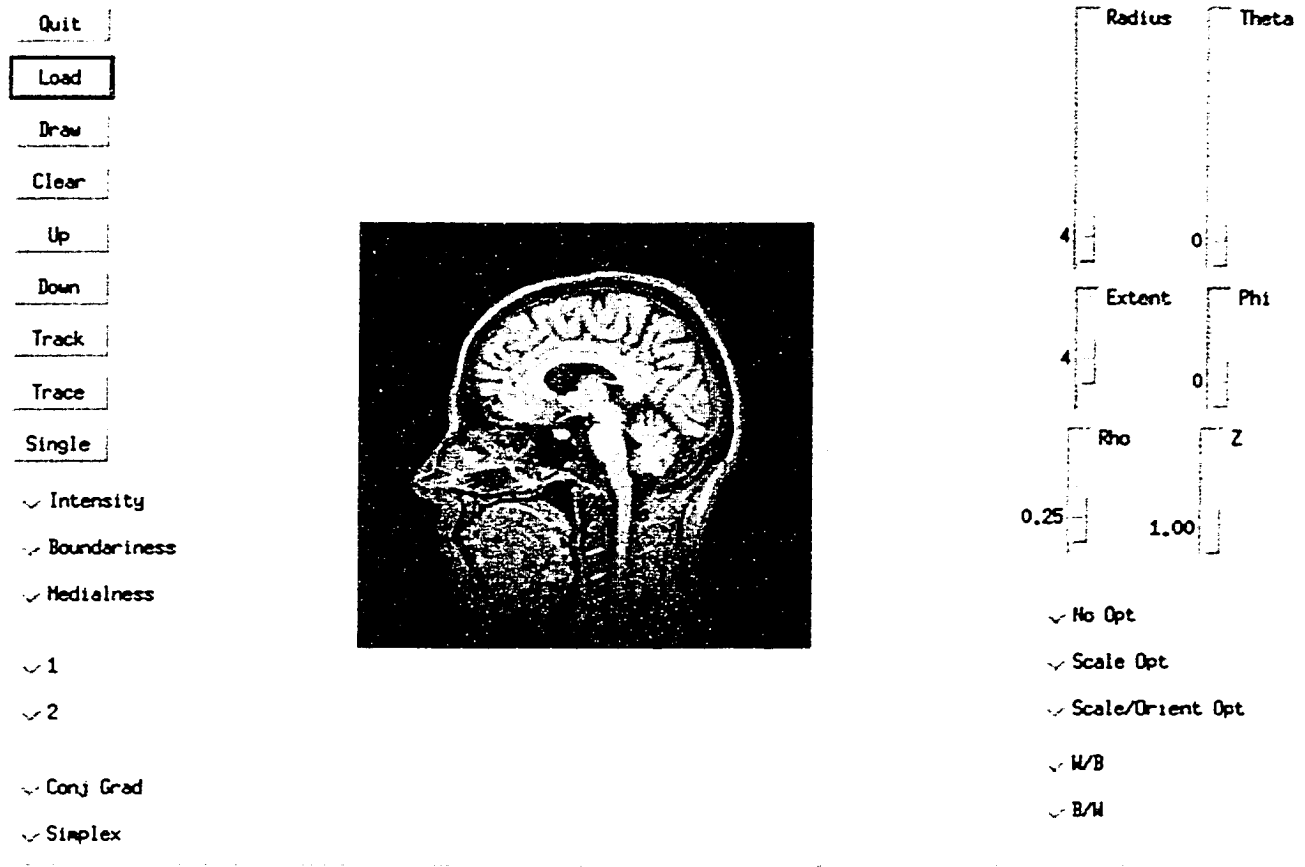


Figure 5.1: Marching Ridges Interface

### 5.3.1 Control buttons

The *quit* button allows the user to exit the program after completing any ridge. It will not interrupt the current ridge calculation. The *load* button allows the user to work with a new image, chosen from a pop-up menu box. When the image is selected and loaded, the value of  $g$  is set to the dimensionality of the image. As with the quit button, this option is only available when the current ridge calculation is complete. The *draw* button allows the user to redraw the image without any ridge points on it, or to redraw the image with the current ridge superimposed. The choice of whether to include the ridge or not is provided by a text menu in the same window from which the Marching Ridges algorithm was initiated. When loading a 3D image, Marching Ridges determines how many slices of the image can be displayed on the image canvas. Marching Ridges then breaks the original image into metaslices, each containing the number of slices than can be simultaneously displayed in the interface. The *up* and *down* buttons are used to view

different metaslices of the images. The *track* button initiates ridge finding based on a user supplied grid element (Section 5.6.1). The *trace* button provides a diagnostic tool for examining function values at a single spatial position parameterized by scale or scale and orientation. The *single* button allows the user to calculate ridges for a single grid element of the function domain space. It provides action identical to the track button but will not extend the ridge if the single element contains a ridge nor search for initial ridge points if the element does not.

### 5.3.2 Image measurements

Marching Ridges allows the user to choose from three functions on which to find ridges. The first choice is an intensity function, in which Marching Ridges measures the intensity of the image using a zero-mean Gaussian weighting function of a specified standard deviation (Section 5.3.5). The second choice is a boundariness function, in which Marching Ridges measures boundariness using either gradient magnitude or oriented first derivatives using a zero-mean Gaussian first derivative weighting function of a specified standard deviation and orientation (in the case of oriented first derivatives). The choice of boundariness functions is provided using a text menu. The third choice is a medialness function, in which Marching Ridges measures medialness using either an isotropic Laplacian, an oriented Laplacian or an oriented Morse/Fritsch medialness weighting function: the Morse function is used to find 2D cores of 3D images while the Fritsch medialness is used to find 1D cores of 3D images. The choice of medialness functions is provided by a text menu. The algorithm uses medialness weighting functions as described in Chapter 3 of a specified standard deviation and orientation for the oriented weighting functions. The choice of measurement may also affect the dimension of the grid elements. Oriented boundariness adds  $l - 1$  to  $g$ , isotropic Laplacian medialness adds 1, and any of the oriented medialness measurements add  $l$ , where  $l$  is the dimension of the image.

### 5.3.3 Ridge dimension

Marching Ridges currently allows the user to find 1D or 2D ridges. In the case of 2D images, only the 1D ridges are appropriate, while both may be sought in the case of 3D images. This choice also sets the ridge dimension  $d$  and codimension  $c = g - d$ .

### 5.3.4 Maximizations

Marching Ridges was originally programmed to accept either simplex maximization or conjugate gradient maximization. However, the current implementation of Marching Ridges only accepts simplex maximization.

### 5.3.5 Parameters

The radius parameter affects the size of the weighting function used to measure the image. In the case of intensity and boundariness measures, the radius is the standard

deviation of the Gaussian weighting function. In the case of medialness measures, the radius is as explained in Section 2.4. The theta orientation parameter is used only for oriented weighting functions: directional derivatives and oriented medialness. Similarly, the phi orientation parameter is used only for weighting functions oriented in 3D. All three parameters, radius, theta and phi, are set by the user to determine an initial value for optimization when finding optimal parameter ridges or to determine the initial grid element for the Marching Ridges algorithm when finding other ridges. The extent parameter determines the footprint of the weighting function. In most cases, the footprint of the weighting function is radius times extent. In the case of Fritsch and Morse medialness, the size of the footprint is  $r(1 + \rho\xi)$ , where  $r$  is the radius and  $\xi$  is the extent. The rho parameter is additionally used as described in Section 2.4 as the ratio between radius and aperture for Morse and Fritsch medialness weighting functions. Finally, the Z parameter is the ratio of the interslice distance to the intraslice distance for 3D images.

### 5.3.6 Optimizations

The three optimization buttons provide the user the choice of no optimization, scale optimization only, or a combination of scale and orientation optimizations, three choices corresponding to choosing certain parameters as transverse directions. These optimizations are used as described in Chapter 4 to calculate optimal parameter ridges. Not shown on the interface is the option of orientation optimization only for boundariness measures made as directional first derivatives. The Marching Ridges algorithm was designed around the decision to make all auxiliary parameters optimal or to make none of them optimal. Thus, for example, it is not possible to find optimal scale ridges of an oriented medialness function. The choice of optimizations may also affect both  $g$  and  $s$ . Scale optimization decreases  $g$  and increases  $s$  by 1. Scale and orientation optimization decreases  $g$  and increases  $s$  by  $l$ .

### 5.3.7 Object/Background Polarity

Marching Ridges is able to find intensity and medialness ridges of white objects on black backgrounds or black objects on white backgrounds. The decision is not material in the case of ridges of boundariness: gradient magnitude is unaffected by polarity and directional first derivatives merely indicate opposite normals, depending on polarity, which does not affect the location of the ridge.

### 5.3.8 Mouse

Marching Ridges supports a three-button mouse. The left button sets a spatial position in the image for the track, trace, and single buttons. The middle mouse button reports the image value and coordinates of the location at which the button is pressed. It does not set an initial location. The right mouse button allows the user to enter an initial spatial position and auxiliary parameter values through the keyboard, rather than the mouse and sliders. These values are then used in a trace.

## 5.4 General Ridge-finding Strategies

Most of the ridge-finding strategies of Marching Ridges are dependent on the codimension  $c$  of the ridge, as that determines what particular subelement identifies ridge elements. However, there are two actions that are essentially independent of  $c$ : actions at points and averaging transverse directions.

### 5.4.1 Actions at Points

Points are the lowest dimensional subelement for any product space. There are three actions that occur at every point, and form the basis for any ridge calculation:

- the calculation of derivatives in coordinate directions,
- the determination of transverse directions, and
- the alignment of transverse directions and the calculation of derivatives in transverse directions.

#### Calculation of Derivatives

The first action at points is the calculation of derivatives. In the case of optimal parameter height ridges, the calculation of derivatives is preceded by a local maximization of the original function  $f$  over the optimal parameters. The point then calculates derivatives of the original function using the coordinates of the point and any optimized parameter values. This is done using symbolic manipulation of the original weighting function to produce derivative weighting functions that are then applied to the image. Each point will apply a large number of derivative weighting functions to the image. In the case of optimal parameter ridges, the derivatives of the optimal parameter function  $f$  are then calculated from the derivatives of  $f$  as described in Section 4.4.5. The result is a set of first and second derivatives at each point.

#### Determination of Transverse Directions

The determination of transverse directions in image space may follow two different paths, depending on whether ridges are optimal orientation or not. In the case of optimal orientation ridges, the transverse directions are determined from the optimal orientation: transverse directions are perpendicular to the optimal orientation. If the ridge finding requires more than one transverse direction (notably 1D ridges from 3D images), they are chosen from the space perpendicular to the optimal orientation using an *a priori* choice that prevents a degenerate set of transverse directions. Given that the weighting function is isotropic in the plane spanned by the two transverse directions, the only critical need is to identify two orthogonal directions. In all other cases, the transverse directions are chosen from an eigen-analysis of the Hessian matrix of second derivatives as described in Section 2.5.1.

### Alignment of Transverse Directions

Because the Marching Ridges algorithm relies on zero-crossings of directional first derivatives, it is important to insure that the directions all share the same sense or sign. Marching Ridges accomplishes this by computing an average set of transverse directions for the element's vertices, and then comparing each point's transverse directions with the average. If the dot product of a transverse direction and its average is negative, then the transverse direction is multiplied by -1. This action does depend on the codimension  $c$  of the ridge since a point has  $c$  transverse directions that it must align. Once each transverse direction has been aligned, the point calculates derivatives in the transverse direction(s)  $\vec{v}_i$

$$D_{v_i}(f) = \vec{v}_i \cdot \nabla(f)$$

#### 5.4.2 Average Transverse Directions

Stetten [43] has a method of performing an eigen-analysis on unit vectors in order to capture an "average" orientation of the vectors without regard to the sense of the vector. This is the method that Marching Ridges uses to create a set of average transverse directions for points.

Given  $h$  vectors  $\vec{v}_i$ ,  $1 \leq i \leq h$  representing a transverse direction,  $\vec{v}$ , construct a matrix  $C$  such that

$$C = \frac{1}{h} \sum_{i=1}^h \vec{v}_i \vec{v}_i^T \quad (5.1)$$

and perform an eigen-analysis of  $C$ . The "average" transverse direction  $\vec{v}$  is the eigenvector of  $C$  corresponding to the greatest eigen-value. If there is more than one transverse direction, perform this analysis separately for each.

### 5.5 Specific Ridge-finding Strategies

The strategy of Marching Ridges is to use the codimension  $c$  of the ridge to push ridge finding to the lowest subelement possible. This will always result in finding ridge points contained in the subelements of a given grid element. If necessary, these points can be "sewn" together to form higher dimensional representations of a ridge (*e.g.*, splines and patches). The particular subelement identifying a ridge point is the one having dimension equal to  $c$ . This is merely a restatement of the fact that two manifolds will generically intersect at points when the sum of their codimensions equals the dimension of the space in which they lie. For example, when finding 1D ridges in a 3D grid element (ridges of codimension two), the actual identification of ridge points is done at the faces (subelements of dimension two) of the cubic grid elements. If necessary, the ridge points on the faces of the cubic grid element can be stitched together to form a curve in the grid element. Marching Ridges currently supports ridges of codimension one and two, so

the following sections concentrate on the actions of edges and faces that produce ridge points.

### 5.5.1 Ridge-finding on Edges

Ridges of codimension one ( $c = 1$ ) share the property of only requiring a single transverse direction. As explained above, the subelement on which to search when  $c = 1$  will be the edge. Examples in which  $c = 1$  include boundaries of objects and generic skeletons of objects. The edge performs three actions in the calculation of ridge points:

- the determination of average transverse directions.
- the calculation of zero-crossings, and
- a check on second derivatives.

#### Determination of Average Directions

(Figure 5.2)

As mentioned, points attempt to align their transverse directions with an average set of transverse directions. Since in this case, edges are identifying ridge points, the edge is responsible for averaging the transverse directions of its endpoints and reporting this information to those points. The edge queries its endpoints and uses Stetten's algorithm (Equation 5.1,  $h = 2$ ) to identify an average transverse direction. Each of its endpoints then aligns its own transverse direction with the average.

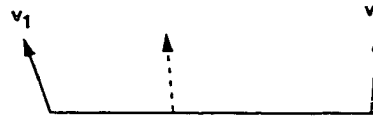


Figure 5.2: Average transverse direction on an edge

#### Calculation of Zero-crossings

(Figure 5.3)

Once the endpoints of the edge have aligned their transverse direction to the average, the edge looks for zero crossings of the first directional derivative of the ridge function in the transverse direction. Assuming that the first directional derivative is a continuous function, the intermediate value theorem ensures that, given first directional derivatives of opposite sign at each endpoint, there will exist at least one point on the edge for which the first directional derivative is zero. Assuming that the first directional derivative is linear, the edge interpolates a location for a single zero crossing. This is a potential location for a ridge point, having satisfied the first condition of being a ridge: a vanishing first derivative in the transverse direction.

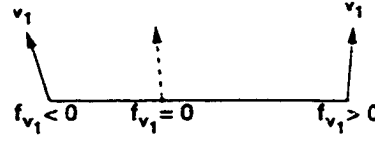


Figure 5.3: Interpolated zero crossing of the first directional derivative

### Second Derivative Check

(Figure 5.4)

Having found a zero crossing of the first directional derivative, the edge must check the second condition for being a ridge: a negative second directional derivative. The interpolated zero crossing performs all the actions that a grid point does in calculating derivatives, finding transverse directions, aligning the transverse direction to the edge average, and recalculating the second derivative in the transverse direction

$$D_{v,v_1}(f) = \vec{v}_1 \mathcal{H}(f) \vec{v}_1^T$$

This derivative is tested against zero, with a negative result marking the point as a ridge point. The point is then drawn to the image window and added to a list of ridge points.

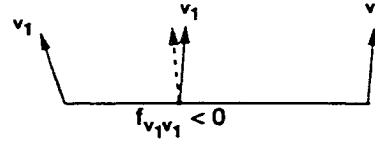


Figure 5.4: Positive identification of ridge point based on second directional derivative

**Note:** the first derivative is not recalculated at the potential ridge point. The linear interpolation is assumed to have provided a close estimate of the zero crossing of the first derivative, so no further check is actually made.

### 5.5.2 Ridge-finding on Faces

Ridges of codimension two ( $c = 2$ ) share the property of requiring two transverse directions. As explained above, the subelement on which to search when  $c = 2$  will be the face. An example in which  $c = 2$  is the skeleton of a tubular object. The face performs four actions in the calculation of ridge points:

- the determination of average transverse directions for the vertices of the face,
- the calculation of zero-crossings of first directional derivatives in the first transverse direction,
- the calculation of zero-crossings of first directional derivatives in the second transverse direction, and



- a check on second derivatives.

### Determination of Average Directions

(Figure 5.5)

As with the edge, a face identifying ridge points must determine a pair of average transverse directions formed from the transverse directions calculated at each of its vertices. Each transverse direction is averaged independently (Equation 5.1,  $h = 4$ ) and the results reported to the vertices of the face.

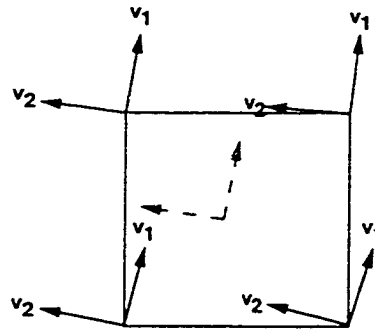


Figure 5.5: Average transverse directions on a face.

### Calculation of Zero-crossings in First Transverse Direction

(Figure 5.6)

Once the vertices of the face have aligned their transverse directions to the average, the face looks for zero crossings of the first directional derivative of the ridge function in the first transverse direction on each of its four edges. At this stage, the face is identifying a codimension one ridge as determined by the first transverse direction. As before, the assumptions of continuity and linearity provide a single location for a zero crossing on any edge containing endpoints with derivatives of opposite sign.

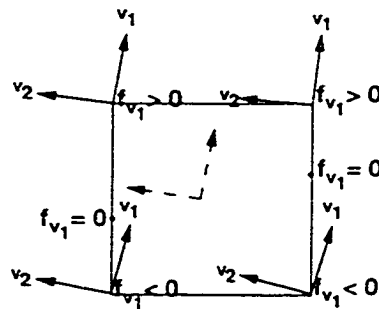


Figure 5.6: Interpolated zero crossings in first transverse direction

### Calculation of Zero-crossings in Second Transverse Direction

(Figure 5.7)

If less than two edges identify zero-crossings, then there is no ridge in the face. If more than two edges identify ridge points, the face chooses the two most convex. (See Section 5.7.1.) These two points then perform the standard actions of points, resulting in first directional derivatives in the second transverse direction. These derivatives are checked for sign, and if they are opposite, a zero-crossing is interpolated along the segment connecting the two. This zero-crossing is a potential ridge point in the face, having satisfied the condition that the first derivatives vanish in each transverse direction.

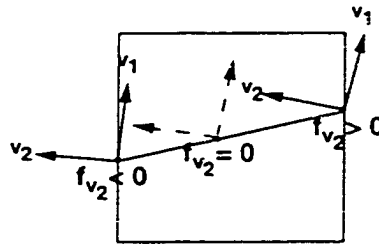


Figure 5.7: Interpolated zero crossing in second transverse direction

### Second Derivative Check

(Figure 5.8)

Having found a zero crossing of the first directional derivative in the second transverse direction, the face must check the second condition for being a ridge: negative second directional derivatives. The interpolated zero crossing performs all the actions that a grid point does in calculating derivatives, finding transverse directions, aligning the transverse directions to the face average, and calculating the second derivatives in the transverse directions. These derivatives are tested against zero, and if both are negative, then the point is marked as a ridge point. The point is then drawn to the image window and added to a list of ridge points.

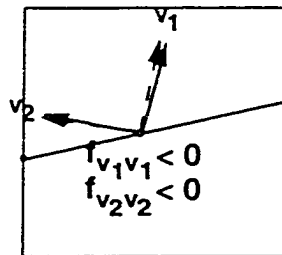


Figure 5.8: Positive identification of ridge point based on second directional derivatives

## 5.6 General Marching Strategies

Most marching strategies are a function of ridge dimension  $d$ . However, three actions are not:

- specifying the initial grid element.
- searching for the initial ridge points, and
- instantiation of new grid elements.

### 5.6.1 Initial Grid Element

Marching Ridges is a semi-automatic algorithm: it requires the intervention of a user to identify a starting point. The user manipulates two mechanisms, the mouse and the parameter sliders, to designate an initial grid element in which to search for a ridge. As mentioned above, the parameter sliders are used to specify the radius and orientation of the initial grid element, while the mouse is used to indicate the spatial position of the initial grid element. (For large 3D images, the up and down buttons may be necessary to locate the correct metaslice before identifying the spatial position with the mouse.) The position so identified is called the anchor vertex of the grid element. The rest of the vertices of the grid element are calculated as a unit cube whose smallest coordinate is the anchor vertex (Figure 5.9). From this initial grid element, Marching Ridges will search for the closest ridge point and then extend the ridge from this identified point.

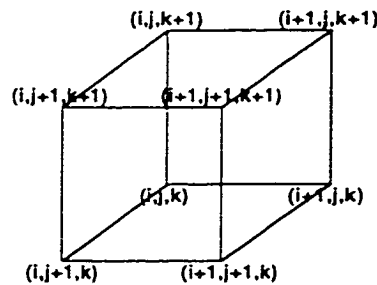


Figure 5.9: Initial grid element

### 5.6.2 Expanding the Search

If the user-specified grid element contains a ridge, then the initial search has succeeded and extending begins. If not, then Marching Ridges begins a breadth first search of the grid elements neighboring the initial grid element until it either finds a ridge point or searches all the grid elements of the search space. If a ridge point is found, Marching Ridges immediately begins extending it. In this way, Marching Ridges finds the ridge point closest to the user specified grid element.

### 5.6.3 Instantiation of New Grid Elements

When a new grid element is required, either in the initial search for a ridge or in the extension of an existing ridge, Marching Ridges creates one. It then identifies the subelements of the new grid element that have already been created and assigns them to the new element. Any subelements that have not been created are then created, with points performing their three actions, and any subelements responsible for identifying ridge points (depending on  $c$ ) performing their necessary actions. In this way, the creation of a new grid element automatically initiates ridge finding procedures.

## 5.7 Specific Marching Strategies

Extending a ridge is similar to the initial search for a ridge point, except that instead of exploring all neighboring grid elements, Marching Ridges only searches grid elements into which the ridge extends and thus depends on  $d$ .

Once ridge points have been found in a particular grid element, Marching Ridges determines the existence of a ridge in each of the border elements of the grid element. Since border elements are shared by two grid elements, each border element that contains the ridge identifies a neighboring grid element into which the ridge extends. The anchor vertex of each of these grid elements is added to a queue of vertices in which Marching Ridges will search for ridge points.

### 5.7.1 Curve Ridges

All 1D ridges are curves, and share the same topological problems, the same heuristics, and the same manner of identifying, continuing, and ending the ridge.

#### Topological Problems

Damon [12] has shown that the maximum convexity height ridge does not generically branch while Miller [34] has shown the same for optimal scale ridges. Marching Ridges uses this assumption even in the case of more general optimal parameter height ridges, for which comparable results are not fully known. To implement this topological constraint, Marching Ridges assumes that any grid element containing a piece of a 1D ridge will only contain two ridge points among its border elements.

#### Heuristic Solutions

The solution to a grid element that identifies too many ridge points is to choose the two ridge points that are most convex, where convexity  $\mathcal{C}$  is defined as the magnitude of the product of second derivatives in each of the transverse directions.

$$\mathcal{C} = \prod_{i=1}^{g-d} |\vec{v}_i \mathcal{H}(f) \vec{v}_i^\top| \quad (5.2)$$

This heuristic is the maximum convexity heuristic and assumes that the most convex ridge points are the ridge points of greatest interest to the user.

### Finding an Initial Ridge Segment

During the initial search for a ridge, a grid element will only identify a ridge if at least two of its border elements identify ridge points. Having identified at least two ridge points, it will then choose the two most convex ridge points from among all those found and report success for the search.

### Continuing the Ridge

Having identified the first grid element containing a ridge, each border element of the grid element containing a ridge point identifies a neighboring grid element in which to extend the ridge (Figure 5.10). The anchor of each of these grid elements is entered into a list of grid elements in which Marching Ridges will search for ridge points. Each such grid element is then required to identify one other ridge point (the exit point) to continue the ridge. If it identifies more than one ridge point among its border elements, it chooses the most convex as the extension of the ridge. This border element in turn identifies a grid element whose anchor is entered into the list. This continues until the ridge ends.

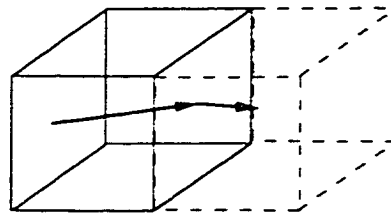


Figure 5.10: A curve ridge continuing into an adjacent cube

### Ending the Ridge

The ridge can end in three ways. It may exit the image when a border element containing a ridge point is unable to identify a neighboring grid element contained in the image. It may close on itself, when a grid element is created with two ridge points already identified among its border elements. Finally, the ridge may end when a grid element with one ridge point (the entry point) fails to identify another ridge point (the exit point).

## 5.7.2 Surface Ridges

All 2D ridges are surfaces and share the following topological problems, heuristics, and manner of identifying, continuing, and ending the ridge.

.

### Topological Problems

The main topological problem with finding a piece of a surface ridge in a grid element is that each border element finds ridge segments independently of the other border elements. Ideally, the ridge segments in each border element of a particular grid element would connect to form a closed loop along the boundary of the grid element: the border of the ridge contained in that grid element. However, with each border element finding ridges independently, there is no guarantee that the ridge segments will connect. Further, there is the same problem of topological ambiguity that appears in the marching cubes algorithm where there are multiple possible connections among ridge segments.

### Heuristic Solutions

Marching Ridges currently uses no heuristics to deal with topological problems in identifying surface ridges. Chapter 6 discusses possible heuristics that may be employed in the future.

### Finding an Initial Ridge Patch

The basic step in identifying an initial ridge patch in a grid element is having the border elements identify ridge curves. This is done exactly as indicated in Section 5.7.1, with the exception that border elements are finding ridge segments, not the original grid element. If any border element of the grid element finds a ridge segment, the grid element contains a ridge patch, and the search has succeeded. The ridge is then continued.

### Continuing the Ridge

As with 1D ridges, having identified the first grid element containing a ridge, each border element of the grid element containing a ridge identifies a neighboring grid element in which to extend the ridge (Figure 5.11). The anchor of each of these grid elements is entered into a list of grid elements in which Marching Ridges will search for ridge points. Each such grid element is then required to identify at least one other boundary element containing a ridge (an exit curve) to continue the ridge. If it identifies more than one ridge curve among its border elements, the ridge continues into all such adjacent grid elements. This continues until the ridge ends.

### Ending the Ridge

A 2D ridge does not “end” in the intuitive way that a 1D ridge does. Rather, the 2D ridge encounters its boundaries. This may happen in two ways. The ridge may exit the image when a border element containing a ridge is unable to identify a neighboring grid element contained in the image. Or the ridge may end when a grid element fails to identify any border elements containing the ridge except for the border elements that contain the incoming ridge. This is the same result that occurs when a 2D ridge closes.

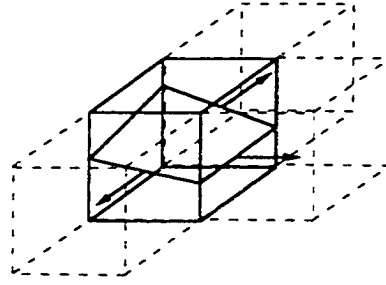


Figure 5.11: A surface ridge continuing into adjacent cubes.

## 5.8 Complexity of Marching Ridges

### 5.8.1 Time Complexity

Marching Ridges runs two major loops. The first searches for a ridge and the second extends an identified ridge. Given a good approximation by the user for an initial ridge point, the search loop will end quickly, so I will concentrate on the extension loop. This loop will iterate once for each grid element that contains a ridge. Let  $p$  be the number of ridge points. Each grid element will be processed in the same fashion, in which the bulk of the work is in the optimization of optimal parameters and the calculation of derivatives of medialness. Each grid element after the first will have no more than half of its vertices performing these calculations. If  $g$  is the dimension of the search space, then  $2^{g-1}$  is the maximum number of new vertices at each step of extension. Additionally, the identification of ridge points requires the same computations as at vertices of a grid element. Each new grid element identifies no more than  $2^{g+c}$  potential ridge points, where  $c$  is the codimension of the ridge. Thus, there are no more than  $2^{2g+c-1}$  optimization and derivative calculations. The number of optimizations depends on the nearness of the initial values to a local maximum, the complexity of the image and the complexity of the medialness weighting function. Experience has shown the optimizer to converge at about 100 iterations; it is coded to stop optimizing after 200 iterations. The number of derivatives calculated depends on the dimension of the search space and the kind of optimization. It will not exceed  $(n(n+1))/2$  where  $n$  is the number of arguments of the weighting function. This limits the number of weighting function applications to  $(200 + (n(n+1))/2)(2^{2g+c-1})$ . The final consideration is the footprint of the weighting function.

#### Isotropic Laplacian and Oriented Laplacian Medialness

The isotropic Laplacian and the oriented Laplacian weighting functions are both used for 1D from 3D cores and 2D from 3D cores. They have identically sized footprints. If  $r$  is the radius of the weighting function and  $\xi$  is the extent, then the footprint of weighting function is  $\frac{4}{3}\pi\xi^3r^3$ . This limits the number of voxel operations in the calculation of a

ridge to

$$\frac{4}{3}\pi(200 + (n(n+1))/2)(2^{2g+c-1})\xi^3 r^3$$

### Oriented Fritsch Medialness

The oriented Fritsch weighting function is used in the calculation of 1D from 3D cores because it is easier to symbolically differentiate. If  $r$  is the radius of the weighting function,  $\xi$  is the extent, and  $\rho$  is the ratio of the radius to the weighting function aperture, then the footprint of the weighting function is  $2\pi r^3 \xi \rho (1 + \xi \rho)^2$ , limiting the number of voxel operations to

$$2\pi(200 + (n(n+1))/2)(2^{2g+c-1})\xi \rho (1 + \xi \rho)^2 r^3$$

The ratio of this to the oriented Laplacian weighting functions is  $3/2(\rho(1 + \xi \rho)^2)/\xi^2$ , and given typical values of  $\rho = 0.25$  and  $\xi = 4$ , the oriented Fritsch weighting function is about 10 times as fast to apply.

### Oriented Morse Medialness

The oriented Morse weighting function is used in the calculation of 2D from 3D cores because of its very small footprint. If  $r$  is the radius of the weighting function,  $\xi$  is the extent, and  $\rho$  is the ratio of the radius to the weighting function aperture, then the footprint of the weighting function is  $8/3\pi \xi^3 \rho^3 r^3$ , limiting the number of voxel operations to

$$\frac{8}{3}\pi(200 + (n(n+1))/2)(2^{2g+c-1})\xi^3 \rho^3 r^3$$

The ratio of this to the oriented Laplacian weighting functions is  $2\rho^3$ , and given a typical value of  $\rho = 0.25$ , the oriented Morse weighting function is about 32 times as fast to apply.

## 5.8.2 Space Complexity

Marching Ridges use  $3 + g$  data structures in the calculation of ridges. The first is the list of grid elements waiting to be searched. The second is the list of ridge points found. The sum of their lengths will not exceed  $p$ , the number of ridge points in the final ridge. The other  $g + 1$  data structures are hash tables that contain all the grid elements and their subelements calculated in the course of finding a ridge. This limits the number of elements to a linear function of  $p$  dependent on the grid dimension  $g$ .

All of the data structures have linear access time and do not affect the time complexity of the algorithm.

## 5.9 Summary

This chapter described a general algorithm for identifying optimal parameter and maximum convexity height ridges. It describes the algorithm in a dimensionally neutral way.



showing that ridge finding strategies are generally dimensionally based while marching strategies are generally codimensionally based. This provides a basis for arbitrary extensions to the algorithm. Chapter 6 shows the results of Marching Ridges for identifying 1D and 2D cores of 3D medical and test images.

# Chapter 6

## Discussion & Results

### 6.1 Overview

This dissertation contains five main contributions to image processing, listed in Chapter 1. In this chapter, I will visit each of these claims and

- relate each of the earlier chapters to specific claims,
- discuss results related to each claim, and
- provide possibilities for future research.

Further, I will examine the thesis presented in Chapter 1 and show how it is supported by results of the marching ridges algorithm, using oriented medialness and optimal parameter height ridges, in finding cores of 3D greyscale images, both synthetic and medical.

### 6.2 Medialness

Chapter 3 presents three additions to the research on medialness measurements: 1) extensions of previously defined 2D medialness measures into 3D, involving decisions about the nature of the core to be extracted and providing a basis from which to extrapolate higher dimensional medialness measures; 2) a new classification of medialness measures based on implicit properties of the measure and type of core extracted from medialness, providing a framework from which to choose medialness measures best suited for a particular task and in which to place new medialness measures; and 3) a new medialness measure, called *Blum-like medialness*, modeled on a circle (or sphere) for which any two arbitrary points (requiring two angles to define) are used to measure medialness, providing an efficient and natural way to measure medialness for points in an object of arbitrary shape.

### 6.2.1 3D Medialness Measurements

The extension of medialness into 3D is a step toward a better shape analysis of 3D images. Much of the shape research done with 2D images could not be applied to 3D images without these medialness measurements. With the availability of 3D medialness measurement functions, some of the work originally performed in 2D is now being conducted in 3D, most notably with deformable shape loci. Further, all of the results displayed in Section 6.4 were created with the 3D medialness measurement functions defined in Section 3.2.1.

A significant direction for future research in medialness measurement is the resampling of medialness weighting functions in proportion to their aperture. Because all the medialness measurement functions in this dissertation are based on Gaussian derivatives, they have the effect of blurring the image when applied as weighting functions: larger apertures blur more. A constant sampling of the weighting function (usually equal to the sampling of the image) provides redundant information at large apertures. Sampling the weighting function in proportion to aperture provides a way of avoiding this redundant information and consequently speeding up the calculation of medialness without loss of accuracy. The time complexity analysis of Chapter 5 shows that in 3D the number of voxel calculations increases as the cube of the aperture: medialness weighting functions at large radii run correspondingly more slowly. Resampling the weighting function offers a constant calculation time for medialness at any radius, making core finding of wide objects no more costly than core finding of narrow objects.

### 6.2.2 Blum-like Medialness

Chapter 3 presented

- a definition of Blum-like medialness, a medialness defined to be efficient and to avoid the effects of interfigural interference in the extraction of cores.

This definition represents a decision to design medialness functions that have a smaller footprint, incorporate more orientation information, and respond better to generic image objects than do other medialness weighting functions. The last two decisions result in a medialness measurement that is less sensitive to the interference of neighboring figure boundaries on the interaction between the boundaries of a single figure.

The results of using Blum-like medialness for locating cores are preliminary. Results for 2D objects were obtained using an algorithm similar to, but predating, marching ridges. The basic objects were binary tear-shapes, with variants including a small indentation, a small protrusion, blurring of the object and perturbing the boundary of the object. In each case, the cores seemed stable despite intensity noise and boundary perturbations, and stopped at the ends of objects. This strengthens the hypothesis that the Blum-like medialness weighting functions may serve as their own endness functions: as the core approaches the end of an object, the two boundary response points of the measurement function come together at a single point, signifying the end of the object. This is equivalent to a Blum symmetric axis with but a single point of tangency to the

boundary of an object. This hypothesis underscores one of the more important aspects of Blum-like medialness: that the Blum symmetric axis has been researched for 30 years and is in many ways well understood. In particular, the relationships defined by Blum and Nackman in their works on symmetric axes provides important clues to the structure of optimal scale/orientation cores. Further, the relationship between symmetric axes and boundaries is well understood and this knowledge could be used in shape models. Medialness that mimics the action of the Blum maximal disks also provides a good opportunity for the comparison of cores and symmetric axes. Katz [26] is currently doing such research with the oriented Laplacian medialness measurements. The application of Blum-like medialness to the marching ridges algorithm and 3D images offers an exciting possibility for further research.

### 6.2.3 Invariances

The section on zoom-invariance of medialness weighting functions presented the requirements for the design of any shape measurement based on Gaussian derivatives. These may prove useful as the study of medialness progresses. This argument arrives at the same conclusion that Eberly has, but uses a simpler and more directed approach than the Eberly argument, in that it does not depend on non-Euclidean metrics and tensor calculus and that it is directly applied to the discussion of shape and the necessary invariances of shape measurements.

## 6.3 Height Ridges

Chapter 4 provided three contributions to the research on height ridges. The first is the definition of the distinguished parameter height ridge, a particular kind of height ridge that requires the specification of certain parameters as transverse directions. The second is the reduction in the dimensionality of the ridge-finding algorithm available with certain parameters as transverse directions. The reduction in dimensionality is explained in the context of three algorithmic choices and is explored closely in the case of the optimal parameter ridge. The third is a proof of the subdimensional maximum property. The subdimensional maximum proof shows that any optimal parameter height ridge point is a local maximum in the space spanned by the directions transverse to the ridge.

### 6.3.1 Optimal Parameter Height Ridges

Chapter 4 provided

- a definition of optimal parameter height ridges, designed to simplify ridge finding based on choices natural to a particular task.

The definition was given as a specific instance of the more general height ridge definition, with the specification of certain parameters as transverse directions and the requirement of maximality in two orthogonal subspaces. The emphasis is on a natural choice of transverse directions. In practice, I have used this definition to find boundaries

of objects in 2D and 3D images using optimal orientation, as well as 1D cores of objects in 2D and 3D images, and 2D cores of objects in 3D images using optimal scale. Further tests of optimal scale-orientation cores for objects in 2D were successful. None of those results are presented here; the 2D results are original only in the application of the optimal parameter height ridge definition, and the 3D results are ignored in favor of the results presented in Section 6.4, calculated using the optimal scale-orientation definition for cores.

A clear direction of future research for the optimal parameter height ridge is the discovery of its generic structure and behavior. Darnon, Miller and Keller have derived the generic structure of maximum convexity height ridges of arbitrary dimension, while Miller has shown the generic structure of optimal scale ridges in 2D. Results for optimal parameter height ridges of arbitrary dimension would be useful in providing expectations of ridge behavior in medical images as well as providing heuristics for the resolution of topological ambiguities in the marching ridges algorithm.

### 6.3.2 Reduction in Dimensionality

This dissertation offers three choices for using the distinguished parameters to reduce the dimensionality of the ridge finding algorithm, important because of the potential for decreased complexity and increased speed. The first, finding ridges on an optimal parameter manifold, is presented as computationally too expensive to offer any time saving in the calculation of ridges, while the third, projecting the optimal manifold onto the image domain, may ignore vital information about the nature of the optimal parameter manifold. I chose the second, the optimal parameter ridge, as providing the best compromise between complexity and loss of information, and I set down the derivative calculations necessary to find an optimal parameter height ridge based on this choice. Further, this choice satisfies the subdimensional maximum property, a property not proved for the other two options for reducing dimensionality. All the results of this dissertation were made using optimal parameter height ridges.

The optimal parameter height ridge relies on an optimization algorithm to satisfy the maximality condition in the optimal parameter transverse directions. Marching ridges currently uses a simplex optimization, a robust but slow method. The speed of the algorithm might be increased with the use of a more sophisticated maximization algorithm. Further, the initial value for the maximizer is provided using only local information; the inclusion of more global information about the core might provide better initialization, both speeding up the algorithm and preventing the identification of maxima far from expected values.

### 6.3.3 Subdimensional Maximum Property

This dissertation also presented

- a proof that optimal parameter height ridges have the subdimensional maximum property.

I showed that every point of an optimal parameter ridge point has the property of being a local maximum of the function restricted to the space spanned by the transverse directions, the directions crossing over the ridge. This is a desirable property of any height ridge point and provides a compelling reason to use the optimal parameter height ridge. The proof is presented such that the space of optimal parameters and the space of remaining transverse directions can be of any dimension. The specific case of optimal orientation determining the remaining transverse directions is natural, but not required for the correctness of the proof.

#### 6.3.4 Connectors

Damon [12] introduced the concept of connectors, a superset of maximum convexity ridges. In his notion, each subset of the set of eigenvectors of the Hessian matrix can identify a different connector, depending on the sign of the eigenvectors. This defines a fully connected set of manifolds whose intersections define lower dimensional connectors. This provides the possibility of more completely defining image geometry without a large increase in algorithmic cost. Examples of the use of connectors include identifying local maxima and saddle points lying on a ridge, the continuation of ridges through noisy or blurred parts of an image object, and the identification of branching subfigures via connectors of medialness that intersect the parent figure's core. The idea of connectors can also be applied to the optimal parameter ridge definition: Miller [34] has begun such work, but much still waits to be discovered. Modifying the marching ridges algorithm to also locate connectors could be a rewarding line of research.

### 6.4 Marching Ridges

The major algorithmic claim of this dissertation is

- an algorithm to extract optimal parameter height ridges from 2D and 3D greyscale images.

The marching ridges algorithm itself is composed of three components : the measurement function, the ridge extraction, and the marching. This modular design enhances the ease with which the algorithm can be extended and incorporated into other work. I have used the algorithm in a number of different ways, and it is general enough to be used in other applications also. I have used the marching ridges algorithm to locate both optimal parameter and maximum convexity ridges. I have located intensity ridges, boundariness ridges, and medialness ridges of 2D and 3D objects, demonstrating the general applicability of both the algorithm and the ridge definition. However, there are still many improvements possible for marching ridges. For example, the added requirement of negative second derivatives means that marching ridges suffers from topological ambiguities at least as great as the ambiguities inherent in marching cubes. While the second derivative does provide a heuristic for making decisions, it does not remove the problem. There are many possible ways to deal with the ambiguities that would be

fruitful directions for further research. Bloomenthal uses simplices. Thirion uses cycles. Researchers at UNC have suggested other approaches involving using more global information about the ridge. Further, the marching ridges algorithm currently supports just 2D and 3D search spaces. While it will likely be used only for 2D and 3D images in the near future, an extension to allow higher dimensional search spaces would allow me to compare the behavior of optimal parameter height ridges with maximum convexity height ridges. The extension into higher dimensions would also allow an exploration of different choices of optimal parameters.

The genesis of and the main impetus behind this entire dissertation is the

- extraction of 1D and 2D cores of 3D medical images.

The satisfaction of this claim and the four preceding it, provide the basis for my thesis: **oriented medialness and optimal parameter height ridges employed in a marching ridges algorithm will effectively extract cores of 3D greyscale images.** Additionally, the results presented here concretely support the algorithmic claim that Marching Ridges does extract optimal parameter height ridges.

I first ran the algorithm on synthetic images; images for which I knew what the behavior of the ridge should be. Satisfied that the results of the test on synthetic images produced very nearly the expected results, I ran Marching Ridges on medical images, images for which I had only an intuitive notion of what the cores should be.

### 6.4.1 Synthetic Objects

There are two main reasons that I tested the algorithm on synthetic images. First, synthetic images are much less complex than medical images. Second, using constructed objects allowed me to know what kind of behavior to expect from the cores of such objects. In each of the results following, I compare calculated ridges points with expected ridge points. Each comparison involves the difference between calculated and expected position, radius, and orientation. Included with each comparison is an image of the synthetic object, with the object implied by the core displayed as a locus of black pixels superimposed on the object. The object implied by the core is generated by drawing a sphere at each point of the core, where the radius of the sphere is the maximal scale at that point. The union of all such spheres is taken as the object implied by the core; this is only approximately the object implied by the core of oriented medialness.

#### Flaring Tube

The first synthetic object is a flaring tube (Figure 6.1). It is oriented along the  $x = y = z$  line in 3D, and its radius is a parabolic function of the parameterized distance along the centerline. It has two hemispherical caps placed at the ends of the tube, beginning at a distance of 32 pixels from the origin along the centerline of the object. The flaring tube provides a test image that includes changing position and changing radius. Given this, I expect a core extending along the  $x = y = z$  line that has an optimal orientation identifying that line and an optimal scale that is a parabolic function of distance from the

origin along the core. I expect the core to end where the hemispherical caps begin. The core shown in Figure 6.2 was generated with the 2-oriented, 3D Laplacian medialness measurement function, responding to boundary circles. Shown is the object implicitly defined by the core overlaid on the original object. I have scaled the original object intensities so that the core object is visible. The actual core contains 64 points and took a little over an hour to generate. Visually, the core behaves as expected until it enters the hemispherical caps, where it deviates from the  $x = y = z$  line, doubling back before it ends. This can be seen in places where the core object is outside the objects of the boundaries. I allowed the core object to extend beyond the boundaries of the image slices to provide a fuller visualization of the core.



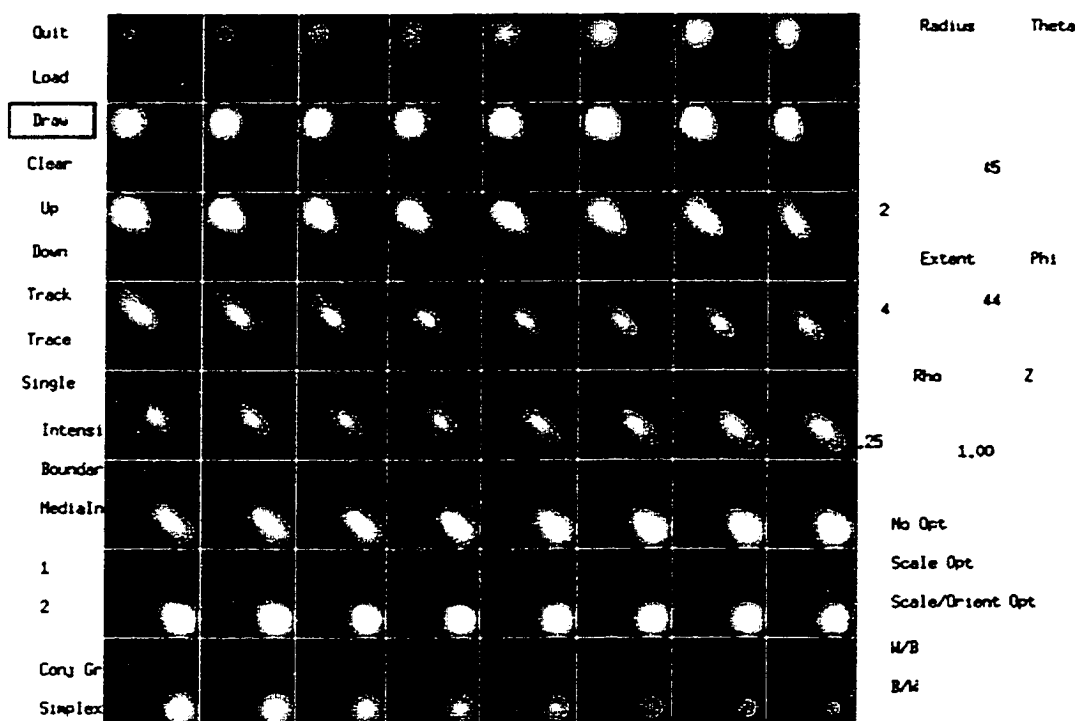


Figure 6.1: Image of flared tube. The 3D image has been divided into slices of constant  $z$  and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices.

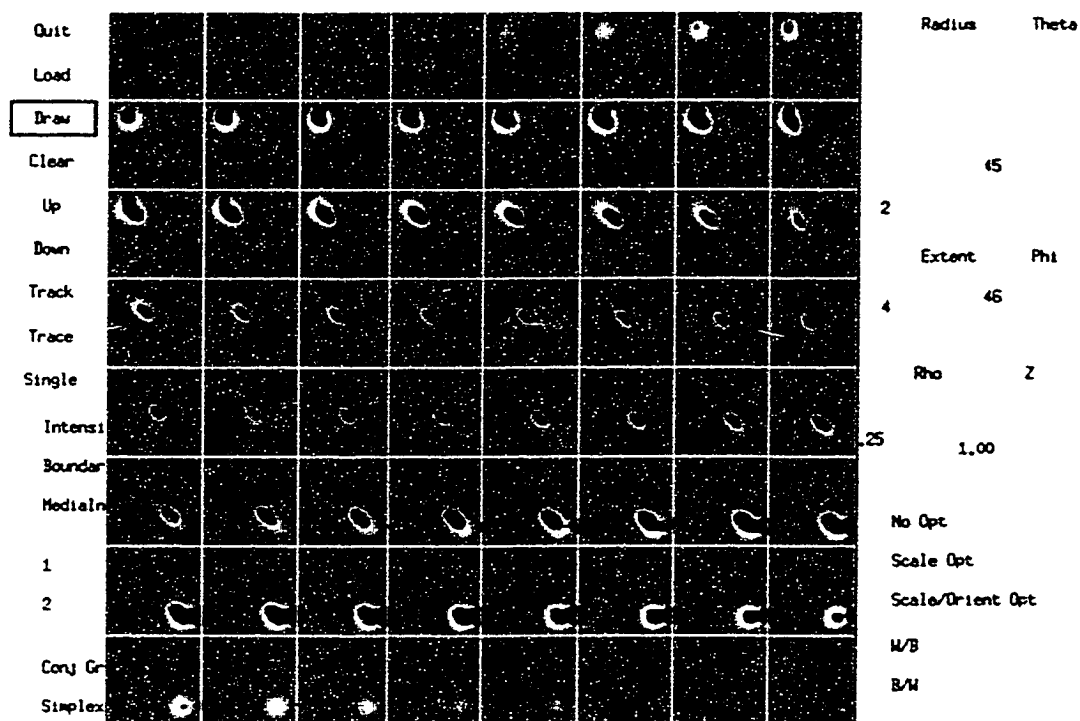


Figure 6.2: Image of flared tube (greyscale) with object implied by core superimposed (black).

Figure 6.3 displays two graphs showing the performance of the core. The X axis of the graph is parameterized distance along the core of the tube, with 0 at the center of the tube. The three functions plotted are radius: on the left, the difference between the actual object radius and the calculated radius of the core, on the right, this distance normalized by the known radius; orientation, the dot product of the orientation of the center line of the tube and the calculated orientation of the core; and position, the distance between each core point and the closest point on the center line of the object. The orientation and position function are both very well behaved until the ends of the objects. This is confirmed visually in the image by the apparent wandering of the core near the hemispherical caps of the tube. This is a potential argument for using either endness or Blum-like medialness in future research. The core consistently underestimates the radius of the tube, getting worse as the tube widens. This may be due to the Gaussian blur applied to the original binary object or it may be that the Laplacian medialness measurement is responding inappropriately to the flared walls of the tube, providing another argument for the continued investigation of Blum-like medialness.

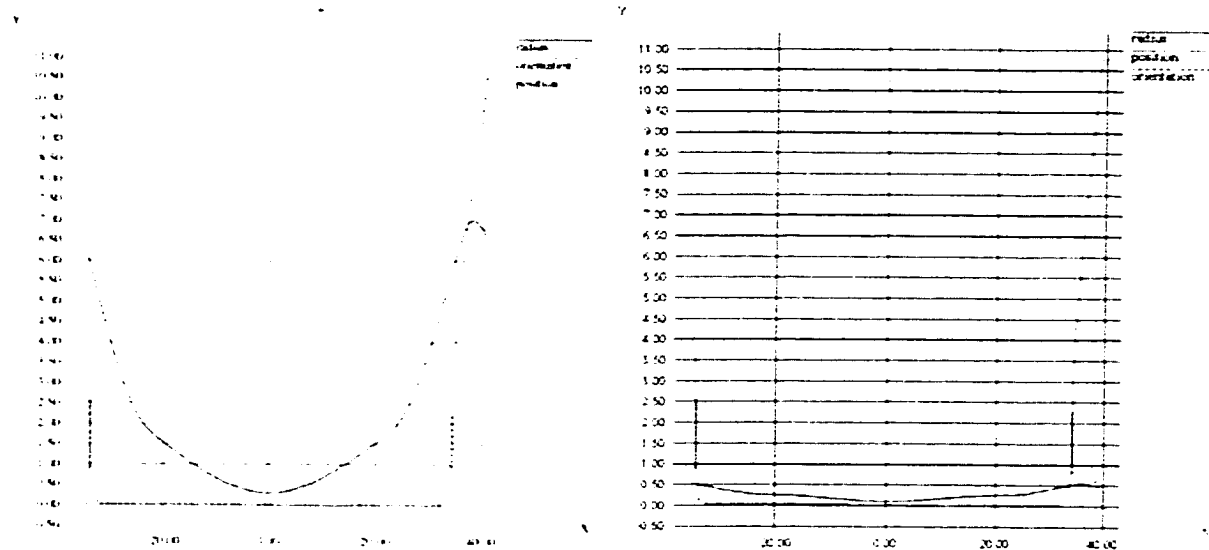


Figure 6.3: Evaluation of core of flared tube, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation.

### Flat Slab

The second synthetic objects is a flat slab (Figure 6.4). It is oriented along the  $x = y$  plane. Its radius is constant and its ends are the boundaries of the image. The core shown (Figure 6.5) was generated with the 1-oriented, 3-dimensional Laplacian medialness measurement, responding to antipodal boundary points. As with the flared tube, the object implied by the core is actually shown. The flat slab provides a test image that includes changing position. Given this, I expect a core extending along the  $x = y$  plane that has an optimal orientation identifying the planar normal and an optimal

scale that is constant. I expect the core to end at the image boundary. The core shown contains 7881 core points and took over an hour to generate. Visually, it lies where expected and ends where expected.

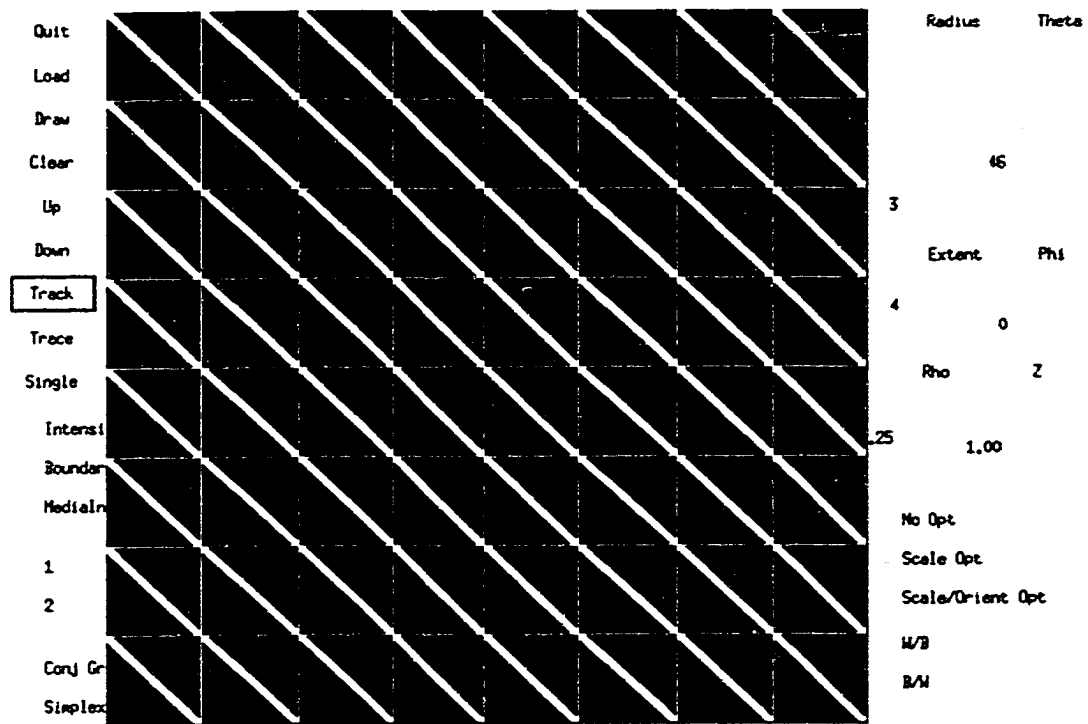


Figure 6.4: Image of a slab (greyscale). The 3D image has been divided into slices of constant  $z$  and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices.

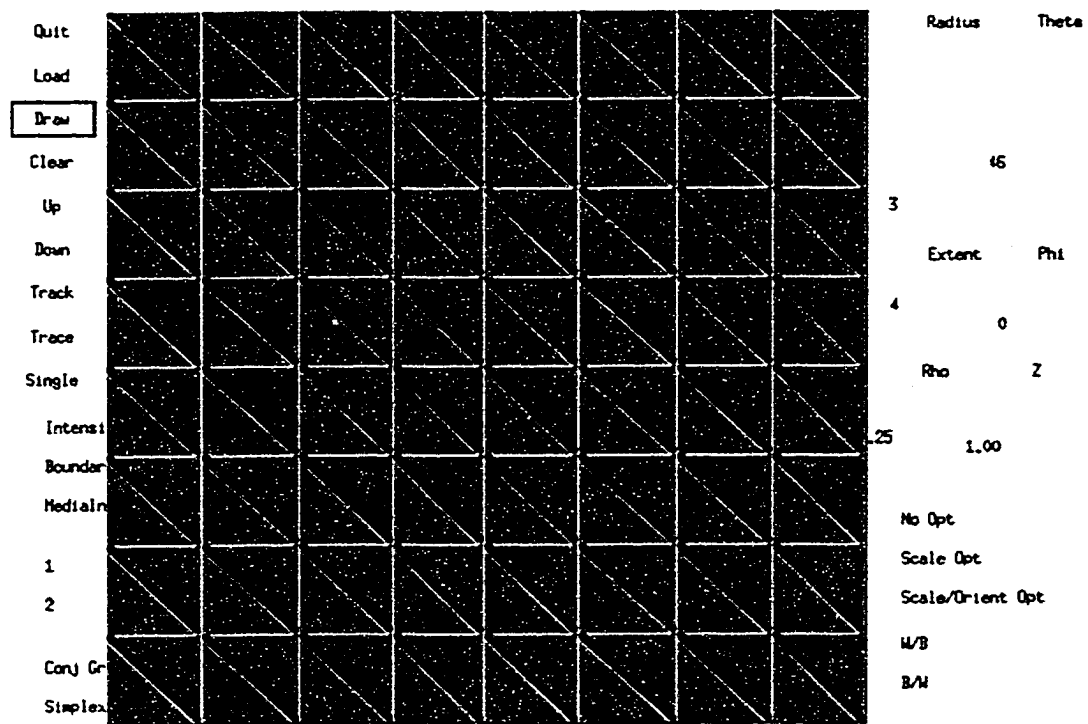


Figure 6.5: Image of a slab (greyscale) with core object superimposed (black).

Figure 6.6 is a graph showing representative performance of the core. I sampled the core on the plane  $z = 31$  and, as before, set the X axis of the graph as the parameterized distance along the core segment, with 0 at the center of the core. The three functions plotted are again radius, the difference between the actual object half-width and the calculated radius of the core; orientation, the dot product of the normal to the center plane of the slab and the calculated normal to the core; and position, the distance between each core point and the closest point on the segment of the core at  $z = 31$ . The orientation is very well behaved. The position has some small variation along the core, but it is under 5% of the interpixel distance for most of the core. As before, the core consistently underestimates the radius of the tube, although with the slab it is generally under half of a pixel. This is likely due to the Gaussian blur applied to the original binary object.

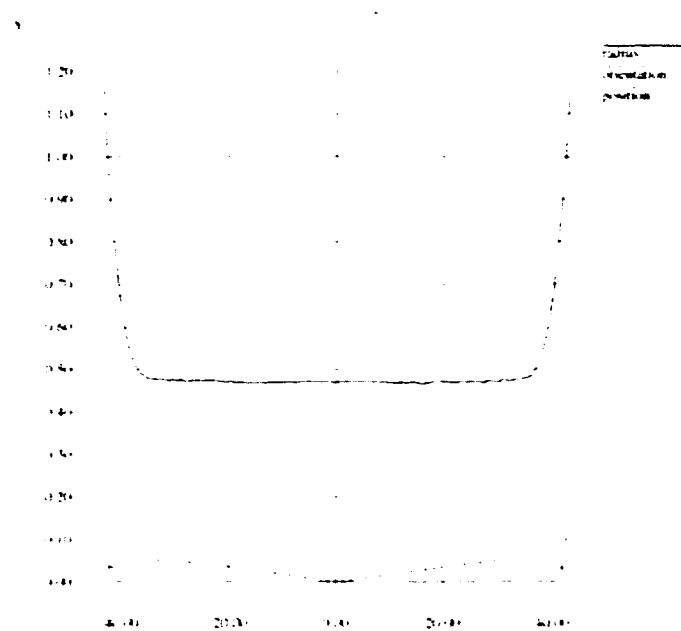


Figure 6.6: Evaluation of core of slab, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation.

### Ellipsoidal Shell

The third synthetic object is an ellipsoidal shell (Figure 6.7). It has a variable radius, thickness and orientation at all points of the shell. The 2D core shown was generated with the 1-oriented, 3D Morse medialness measurement, responding to two antipodal points. The ellipsoidal shell provides a test image that includes changing position, changing radius and changing orientation. I expect the core to lie between the inner and outer boundaries of the shell, with the optimal orientation identifying a vector normal to the shell. The radius will depend on the position of the core point. The core should close in on itself and have no boundaries. The core (Figure 6.8) contains 9255 points and took two

hours to generate; as expected, considerably faster than the generation of the core of the slab using the Laplacian medialness measurement. Visually, it lies where expected and does not end. As with the previous two objects, I have displayed the original ellipsoidal shell and the object implied by the core superimposed on the ellipsoidal shell.

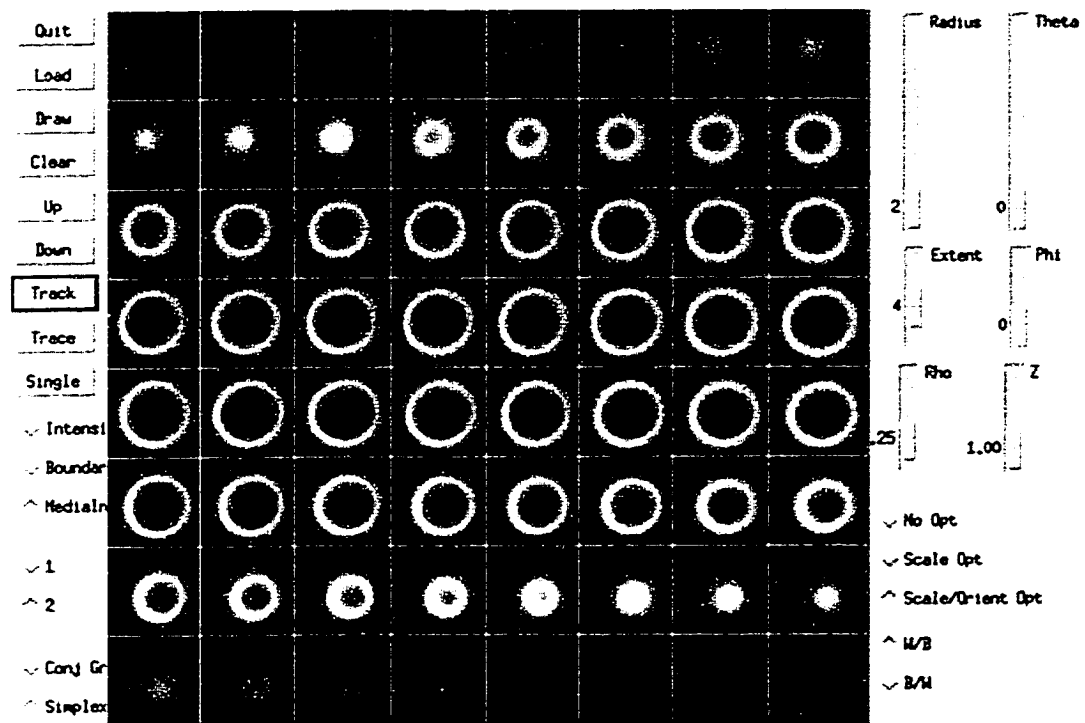


Figure 6.7: Image of an ellipsoidal shell (greyscale). The 3D image has been divided into slices of constant  $z$  and the slices are presented from left to right, then top to bottom. Each slice is 64 pixels by 64 pixels; there are 64 slices.

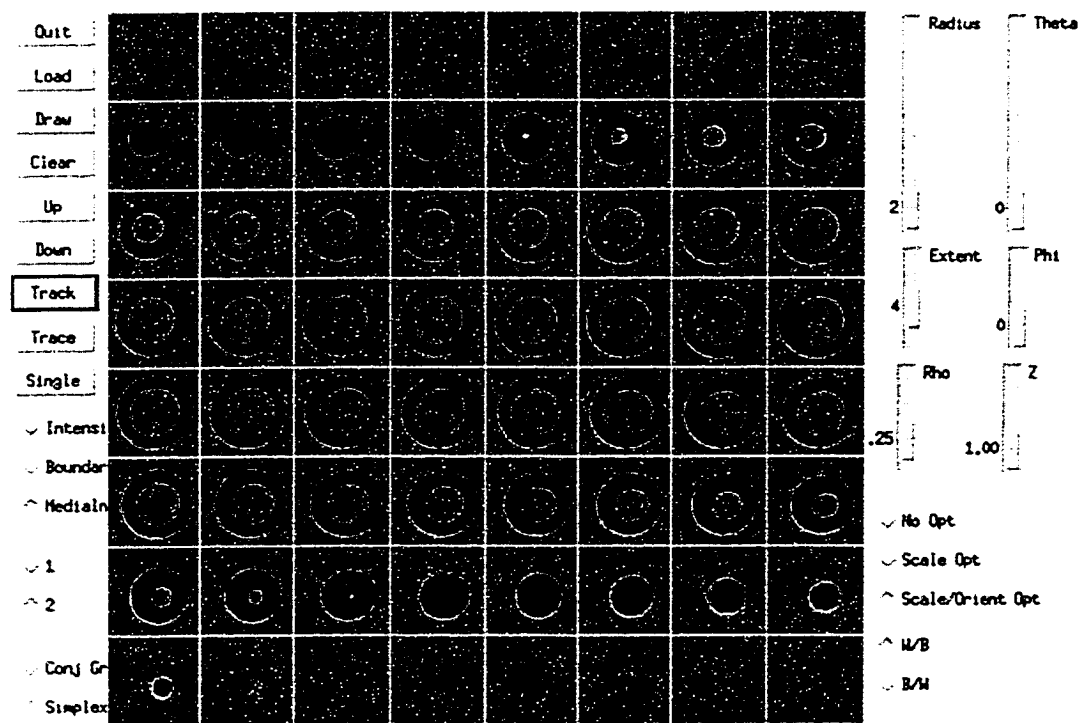


Figure 6.8: Image of an ellipsoidal shell (greyscale) with core object superimposed (black).

Figure 6.9 is a graph showing representative performance of the core. As with the slab, I chose the intersection of the core and the plane  $z = 31$  as a sample on which to examine the performance of the core. The X axis of the graph is parameterized as angle (in radians) around the circle of the core sample. The three functions plotted are again radius, the difference between the actual object half-width and the calculated radius of the core; orientation, the dot product of the normal to the center of the shell and the calculated normal of the core; and position, the distance between each core point and the closest point on the circle of the core at  $z = 31$ . The orientation is very well behaved. The position is consistently between 0.2 pixels and 0.4 pixels from the actual center of the shell: less than half the interpixel distance. The radius overestimates where the shell is thin and underestimates where the shell is thick. This may be pixel noise causing the optimizer to find a small local maximum or it may be the response of the Morse medialness measurement to non parallel boundaries, again arguing for further research of Blum-like medialness measurements.

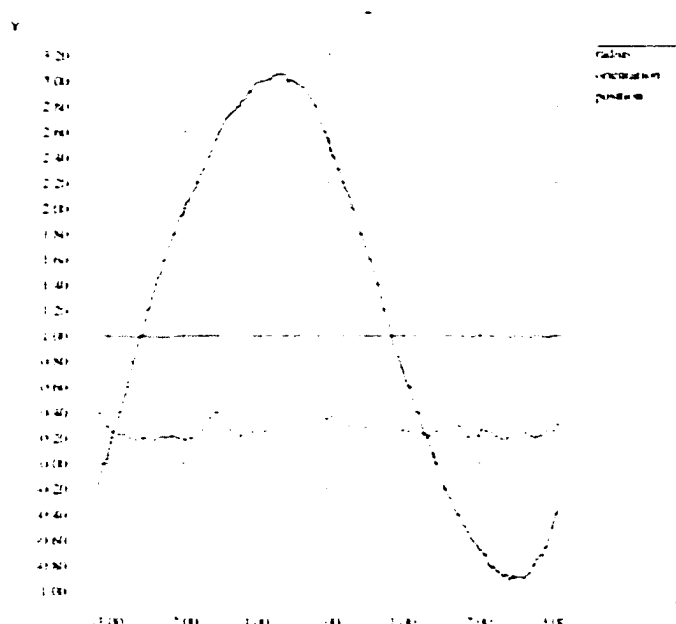


Figure 6.9: Evaluation of core of ellipsoidal shell, showing difference between expected and actual position and radius, and the dot product of the expected and actual orientation.

### 6.4.2 Medical Images

After gaining some confidence in the ability of marching ridges to extract cores that are correct, I tested it on medical images. I have included four results here: the core of cerebral blood vessels, a skull, kidneys and brain ventricles. As before, there are two displays for each medical image: the first is the medical image displayed four slices at a time and the second is the object implied by the core superimposed in black on the medical image.



### Cerebral Blood Vessels

The first medical image is of cerebral vasculature (Figures 6.10 through 6.29). The image is a MRA scan taken from the top of the head down, with slices parallel to the  $z$  plane. Each slice is 256 by 256 pixels; there are 48 slices. Each pixel is 0.859 mm by 0.859 mm; the distance between slices is 1 mm. The figures I have included each contain four slices of the original image. The slices are arranged with the slice highest in the body in the upper left corner and are arranged left to right and then top to bottom with the lowest slice in the lower right corner. I have included only those slices of the original image which contain calculated core points. The points were generated at the rate of about 1 point every 20 seconds. The blood vessels are tube-like structures for which I used the 2-oriented, 3D Laplacian weighting function to calculate medialness. I expect the 1D cores of blood vessels to be curves running along the center of the vessel, with an optimal orientation along the direction of the blood vessel and an optimal scale proportional to the width of the blood vessel. Ridge theory indicates that ridges don't branch. The cerebral vasculature does branch, and I expect the core to stop when it encounters a branch. Further, the vessels narrow, fade, and turn sharply, all conditions that make ridge tracking difficult. The endings of the cores in the figures can all be attributed to such events. The only comparable research results are from Stephen Aylward [1], who performed an analysis of cerebral vasculature using an algorithm that first found an intensity ridge and then maximized medialness over scale along the intensity ridge. He also had trouble with vessels ending; the ridges in he found in the same image averaged 25 voxels in length. Recently, Aylward has found that using connectors has increased the length of his ridges, opening the possibility of using connectors for optimal parameter ridges.

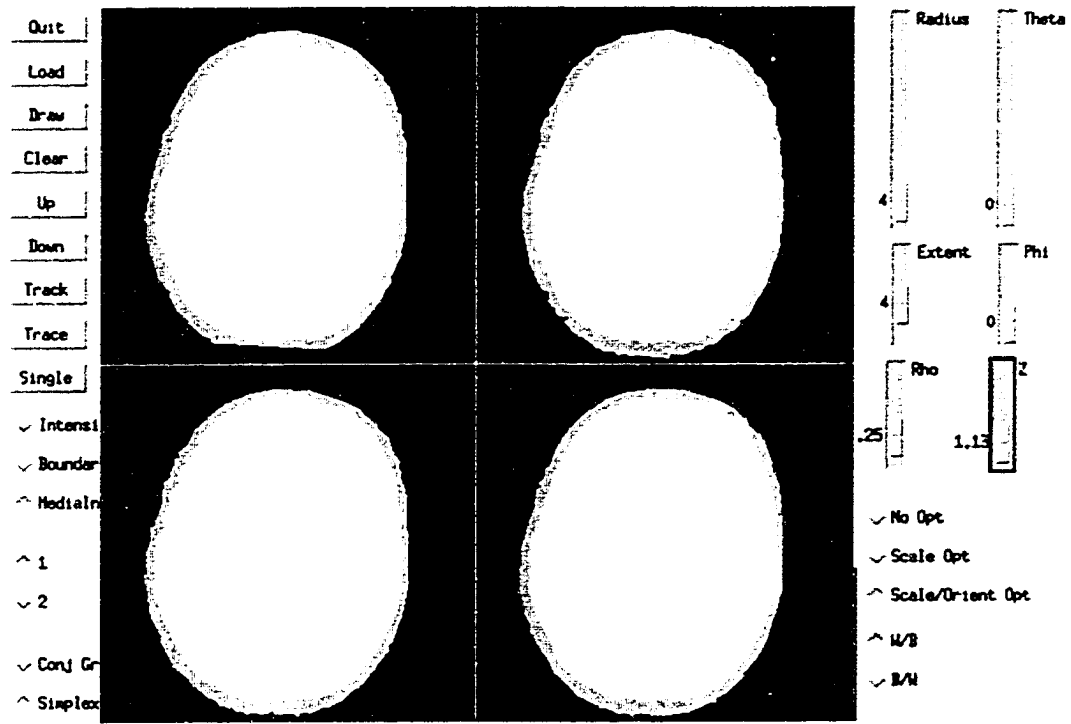


Figure 6.10: Image of cerebral vasculature (metaslice 1 - image slices 5-8)

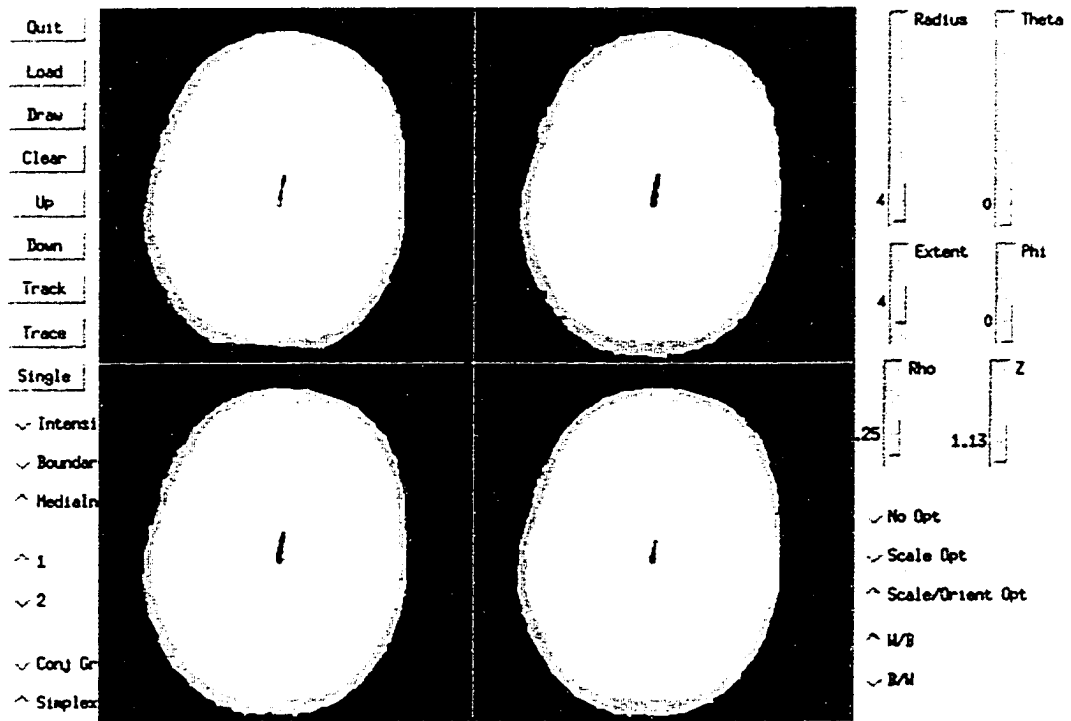


Figure 6.11: Image of cerebral vasculature (metaslice 1 - image slices 5-8) with object implied by the core superimposed in black

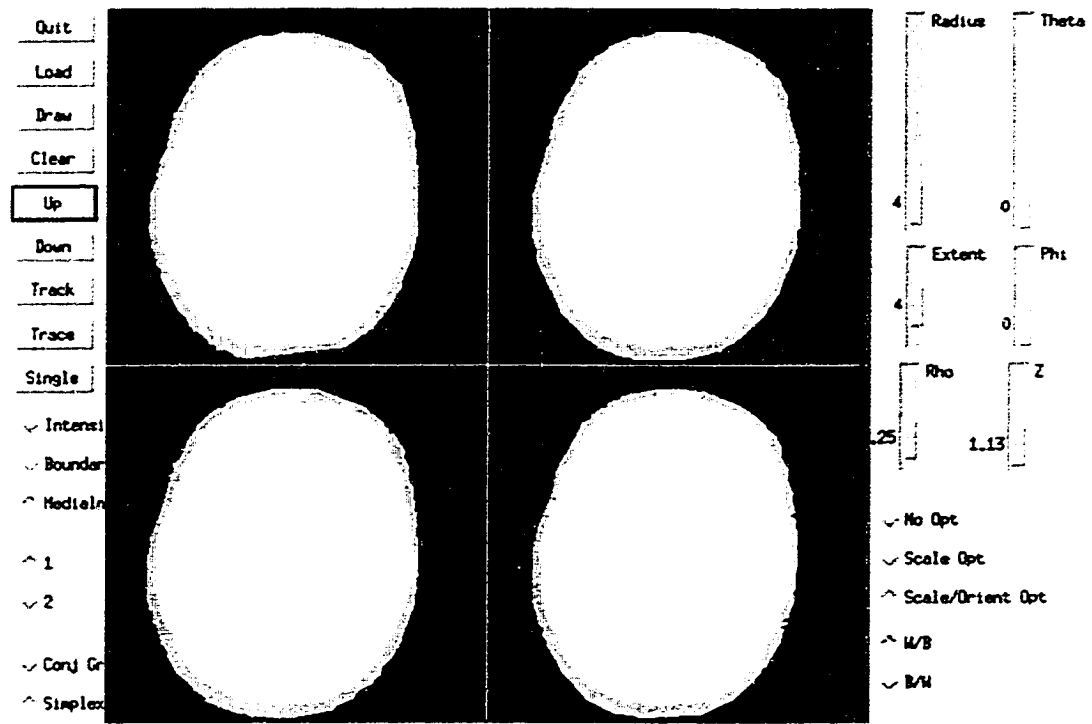


Figure 6.12: Image of cerebral vasculature (metaslice 2 - image slices 9-12)

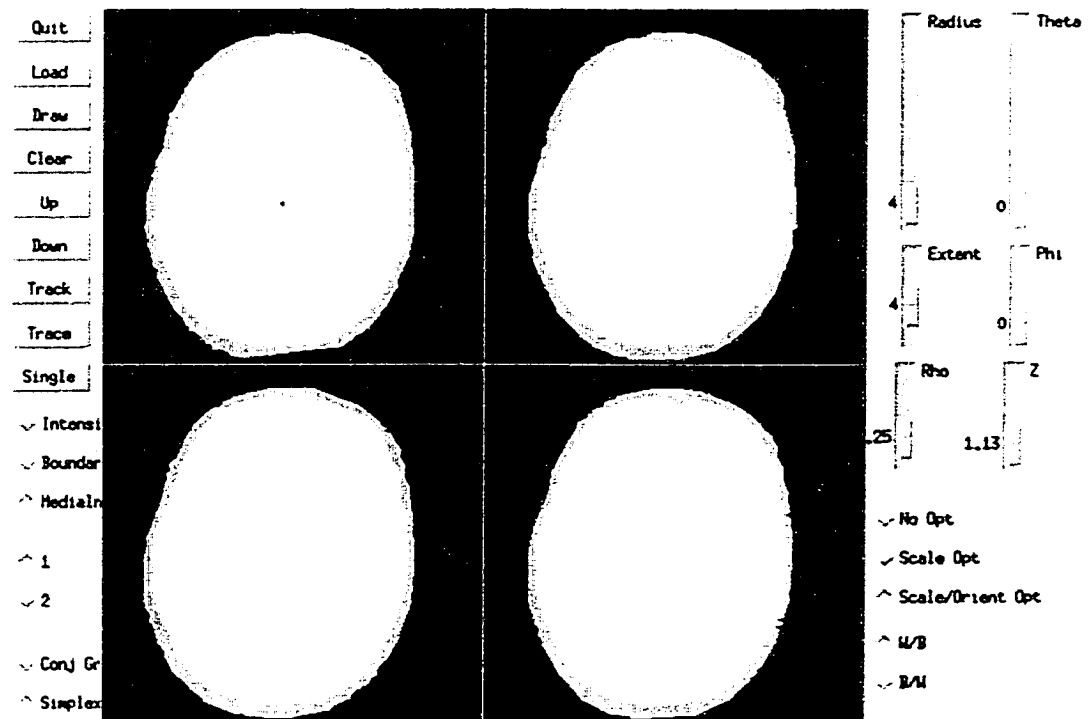


Figure 6.13: Image of cerebral vasculature (metaslice 2 - image slices 9-12) with object implied by the core superimposed in black

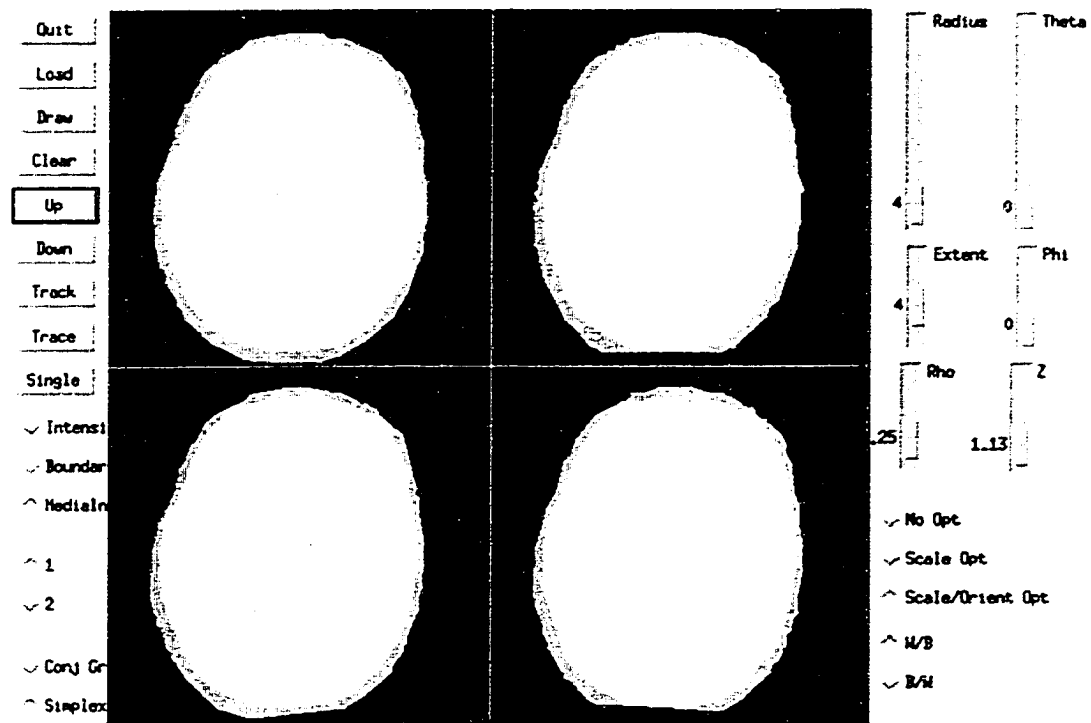


Figure 6.14: Image of cerebral vasculature (metaslice 3 - image slices 13-16)

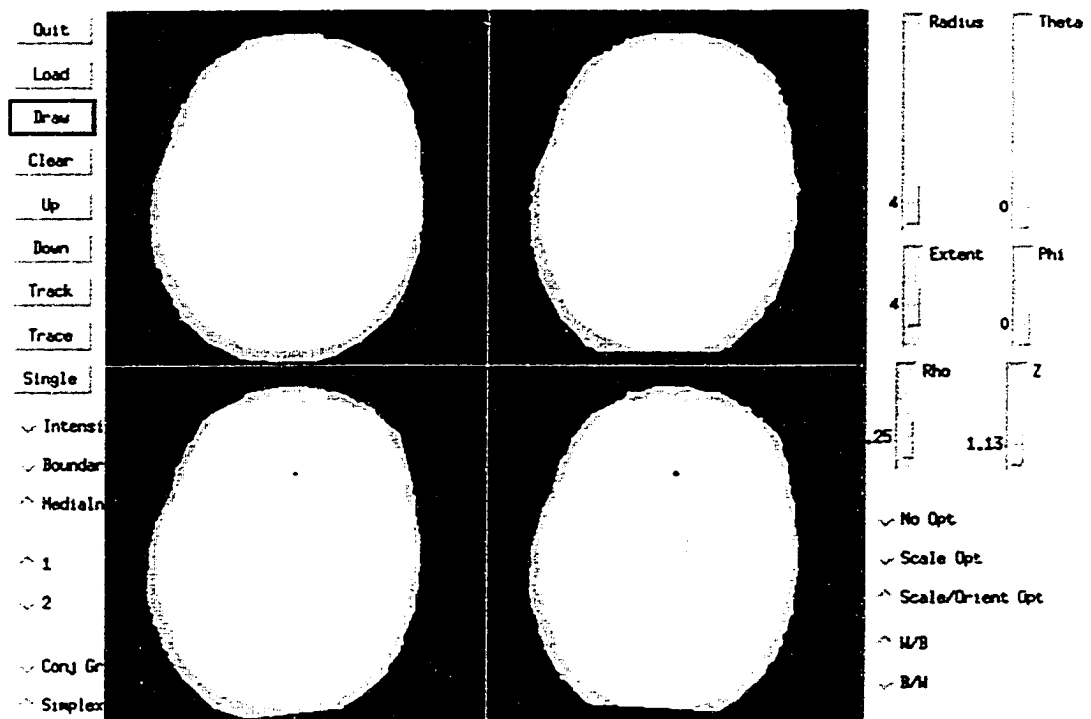


Figure 6.15: Image of cerebral vasculature (metaslice 3 - image slices 13-16) with object implied by the core superimposed in black

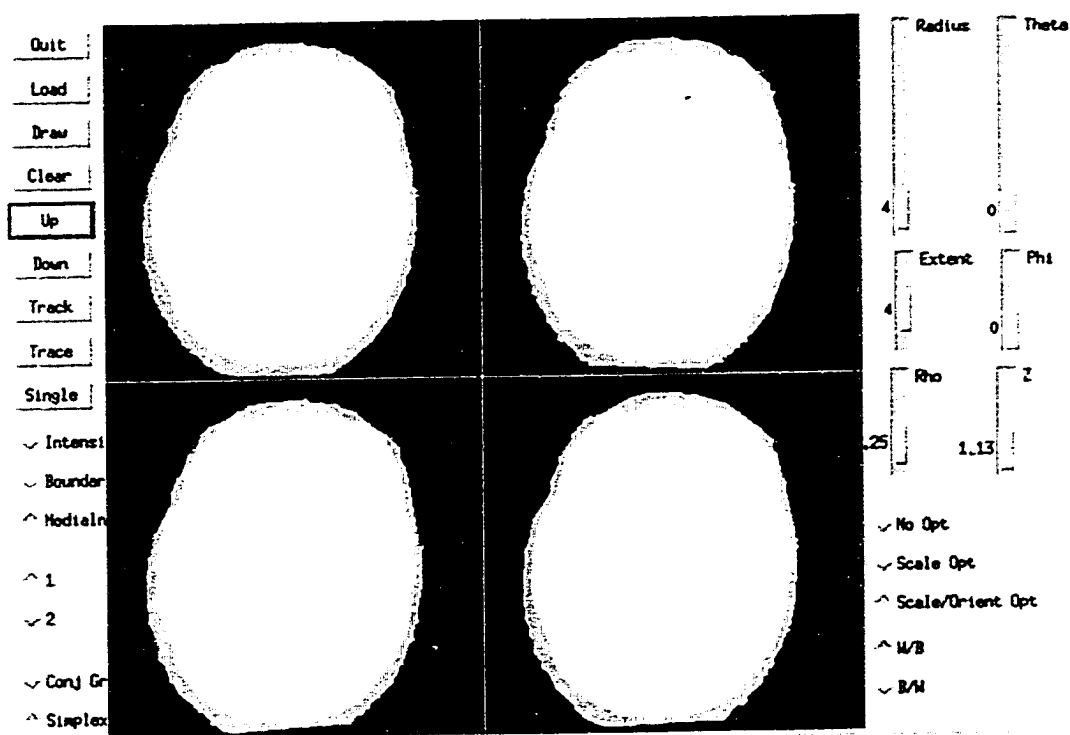


Figure 6.16: Image of cerebral vasculature (metaslice 4 - image slices 21-24)

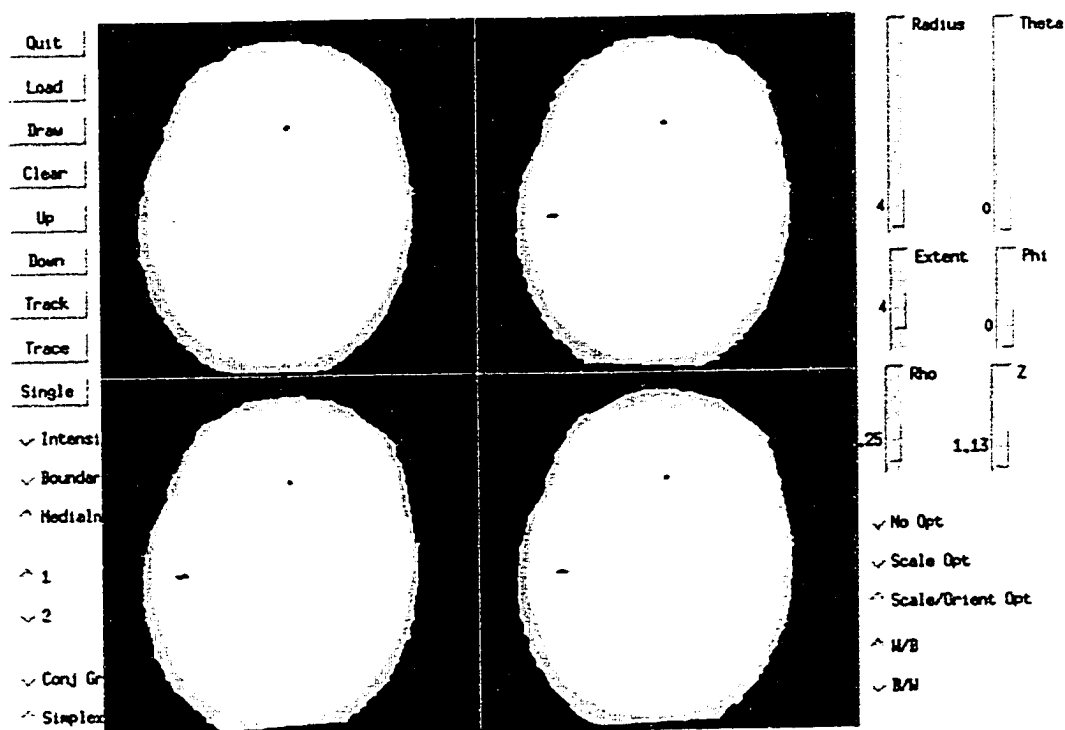


Figure 6.17: Image of cerebral vasculature (metaslice 4 - image slices 21-24) with object implied by the core superimposed in black

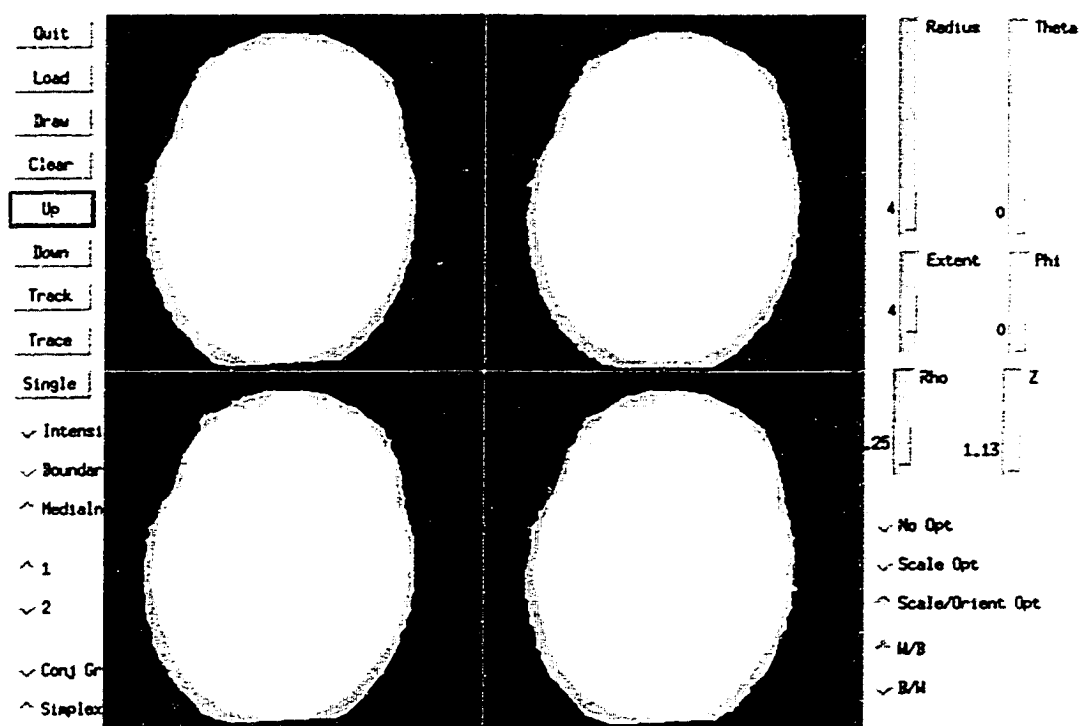


Figure 6.18: Image of cerebral vasculature (metaslice 5 - image slices 25-28)

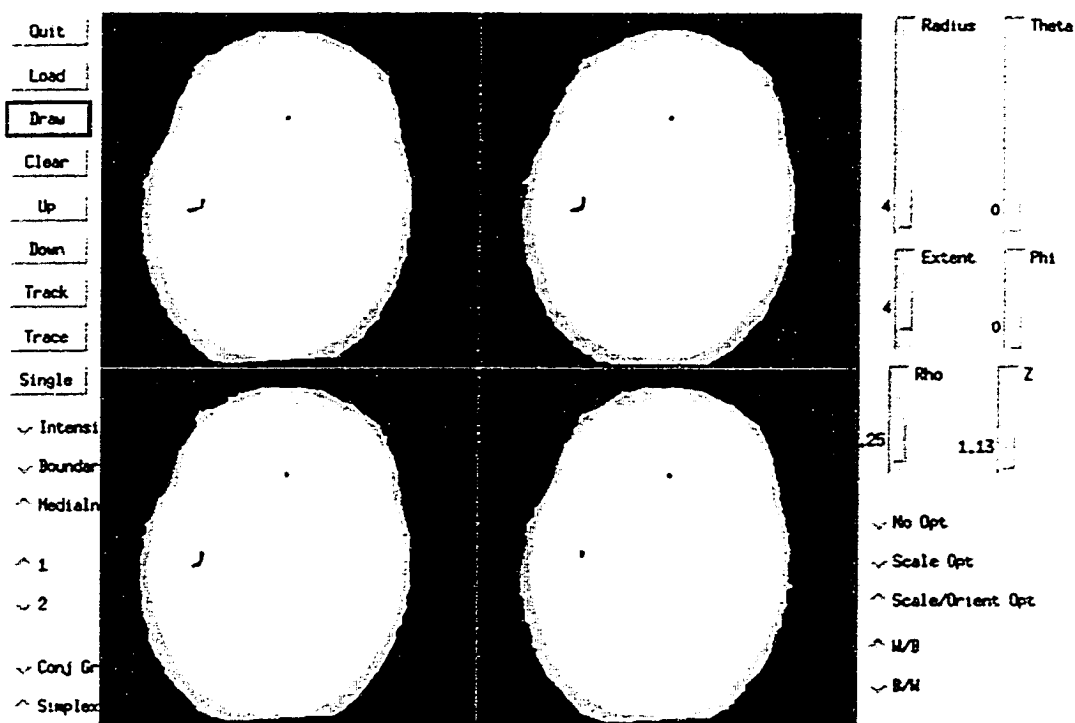


Figure 6.19: Image of cerebral vasculature (metaslice 5 - image slices 25-28) with object implied by the core superimposed in black

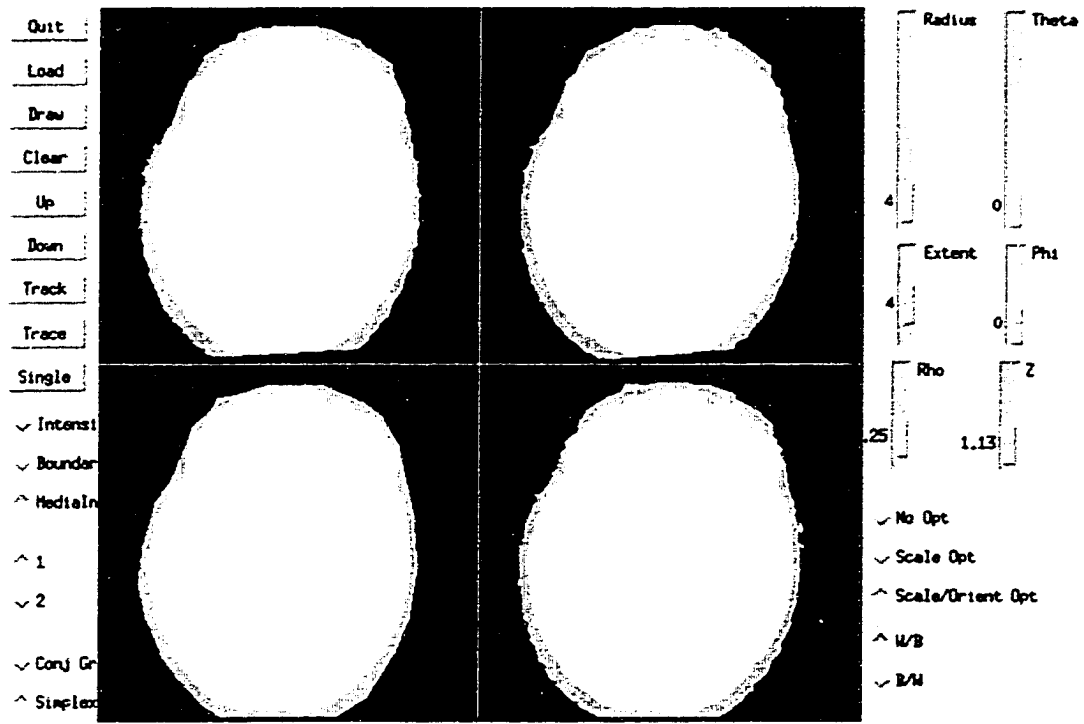


Figure 6.20: Image of cerebral vasculature (metaslice 6 - image slices 29-32)

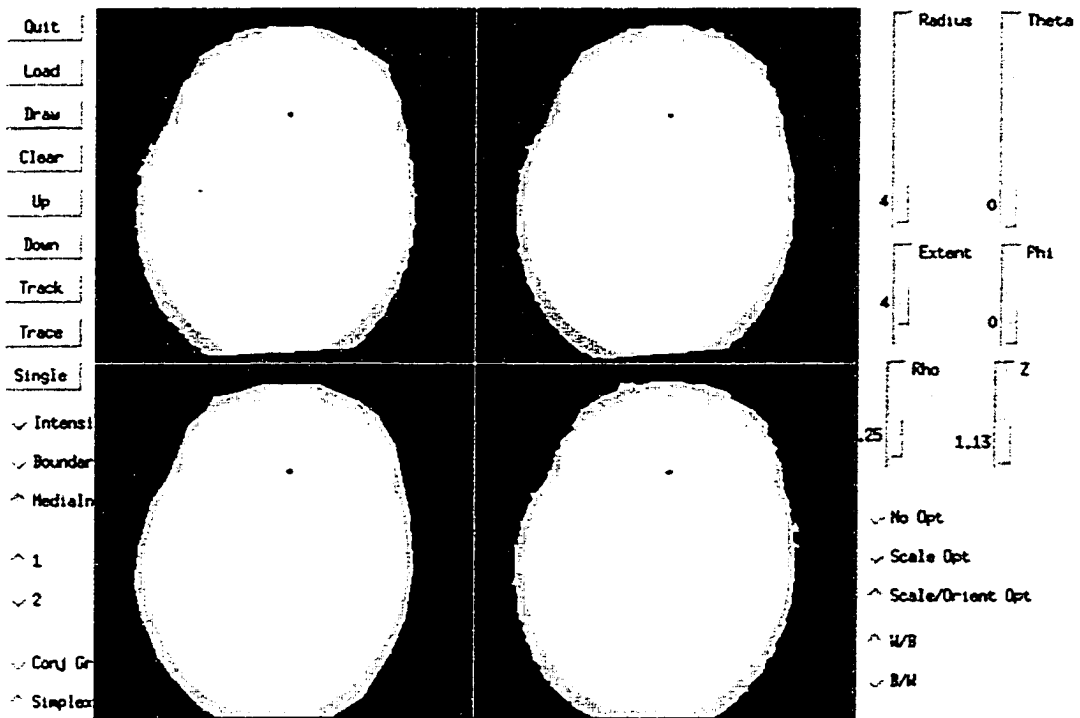


Figure 6.21: Image of cerebral vasculature (metaslice 6 - image slices 31-32) with object implied by the core superimposed in black

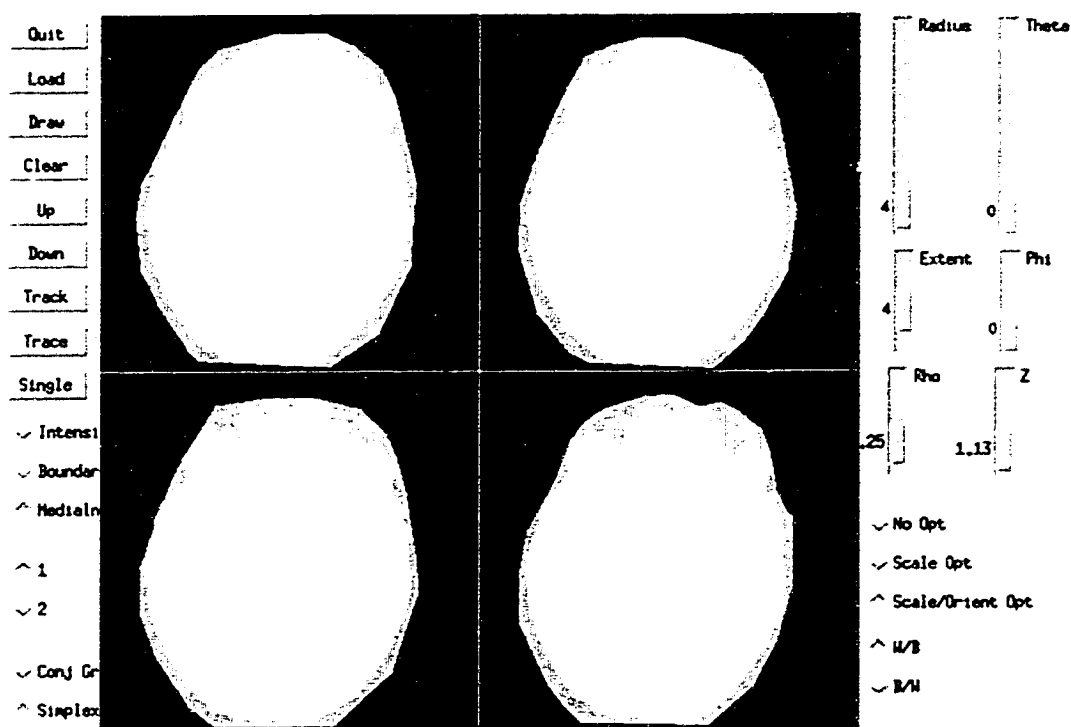


Figure 6.22: Image of cerebral vasculature (metaslice 7 - image slices 33-36)

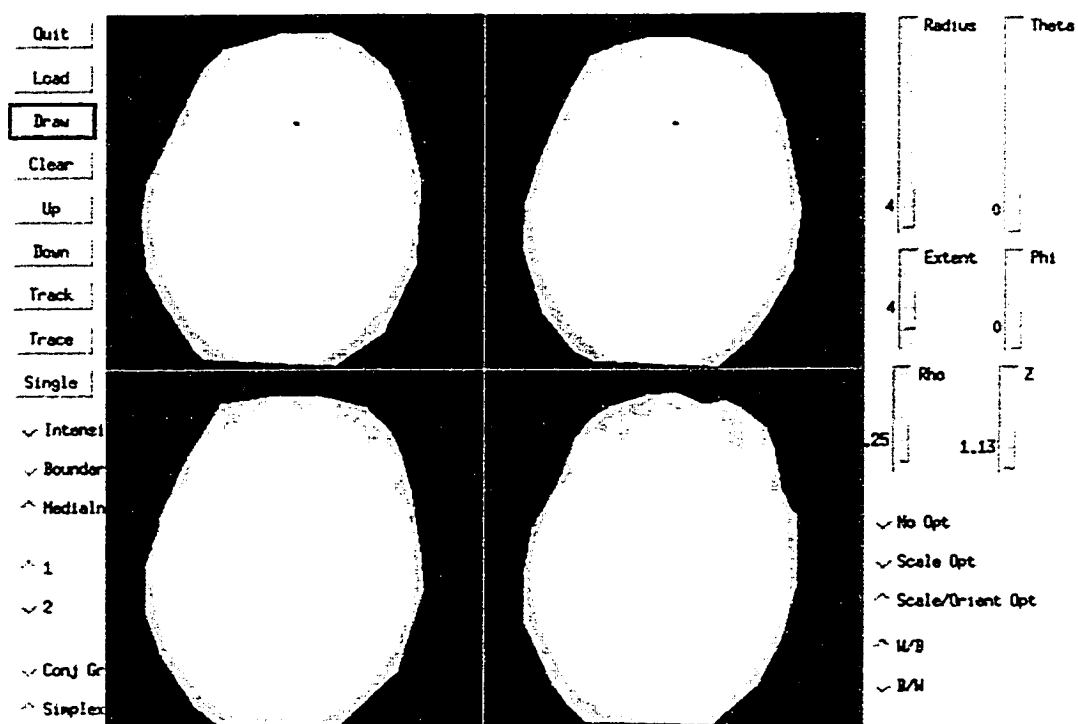


Figure 6.23: Image of cerebral vasculature (metaslice 7 - image slices 33-36) with object implied by the core superimposed in black



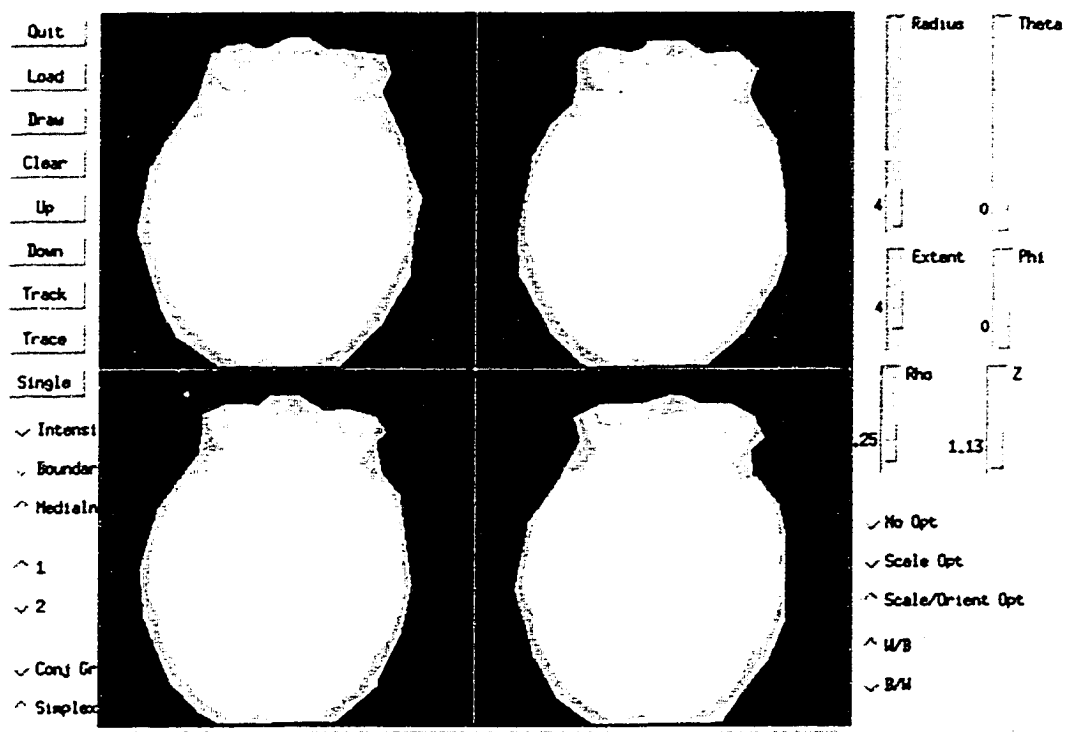


Figure 6.24: Image of cerebral vasculature (metaslice 8 - image slices 37-40)

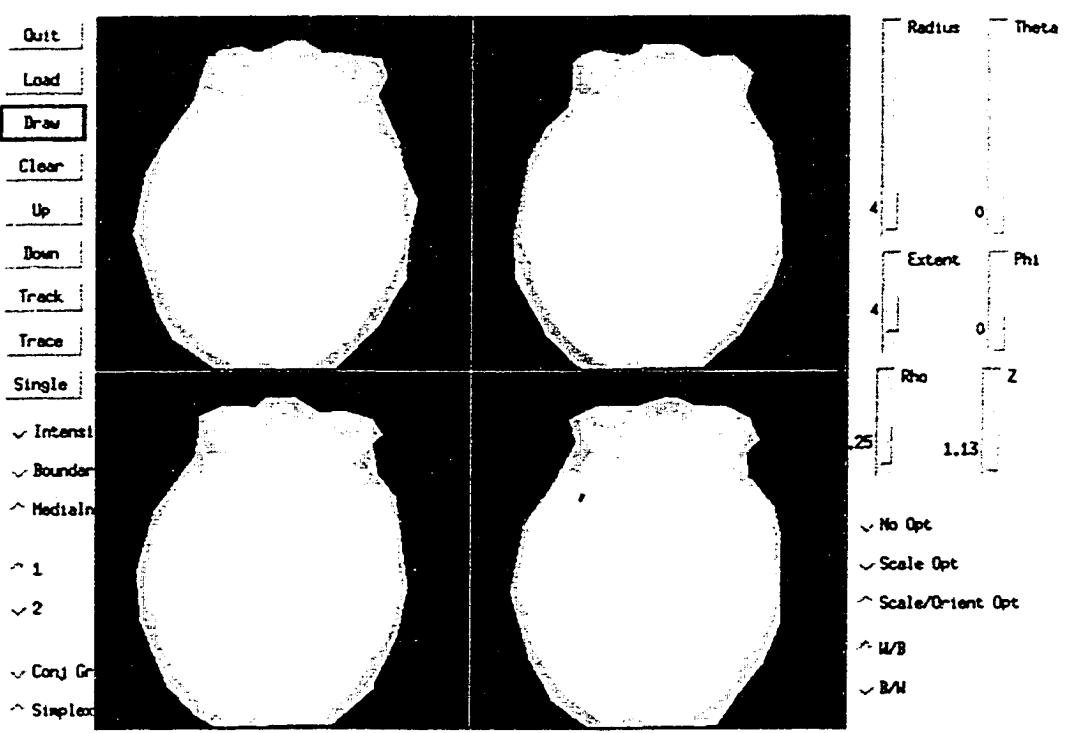


Figure 6.25: Image of cerebral vasculature (metaslice 8 - image slices 37-40) with object implied by the core superimposed in black

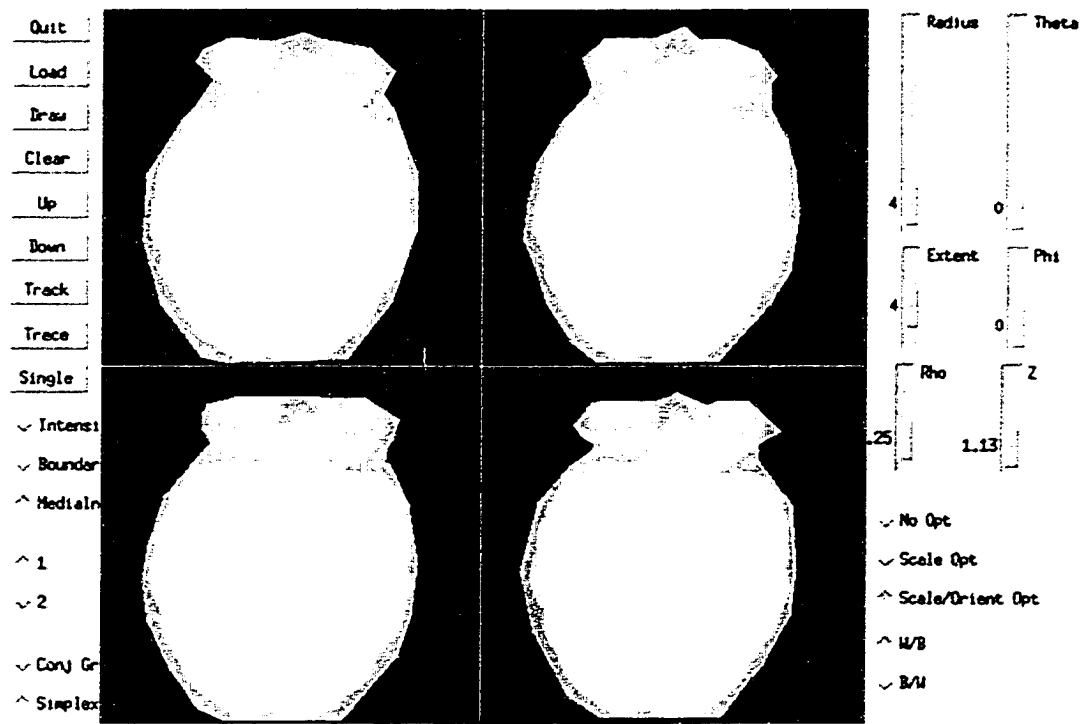


Figure 6.26: Image of cerebral vasculature (metaslice 9 - image slices 41-44)

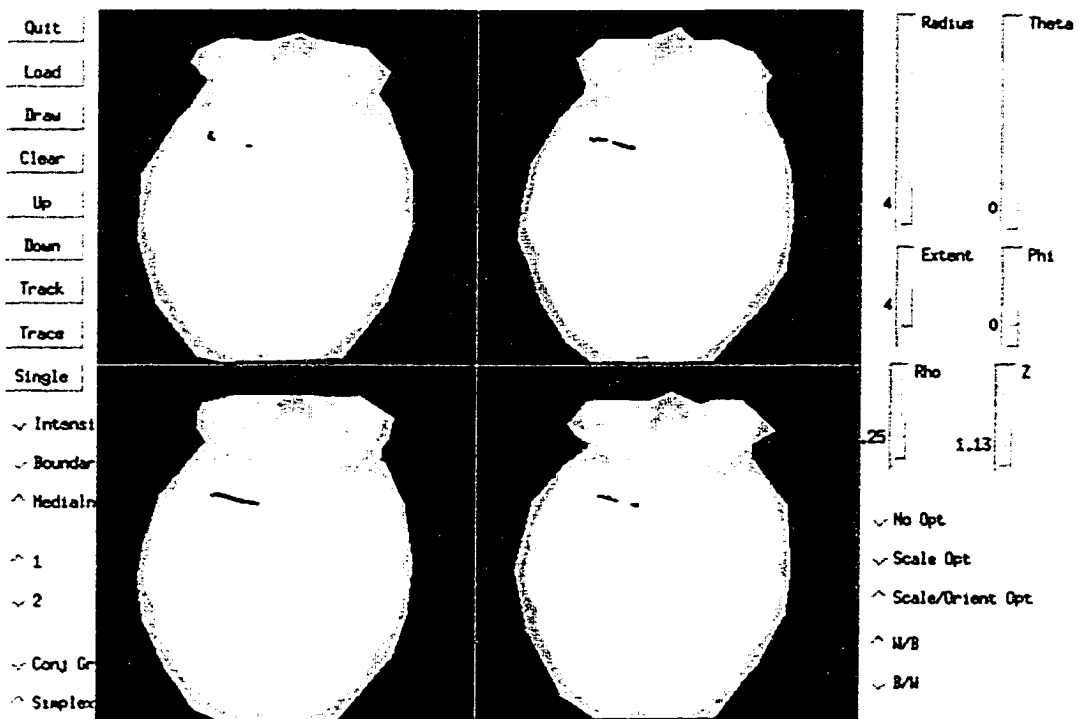


Figure 6.27: Image of cerebral vasculature (metaslice 9 - image slices 41-44) with object implied by the core superimposed in black

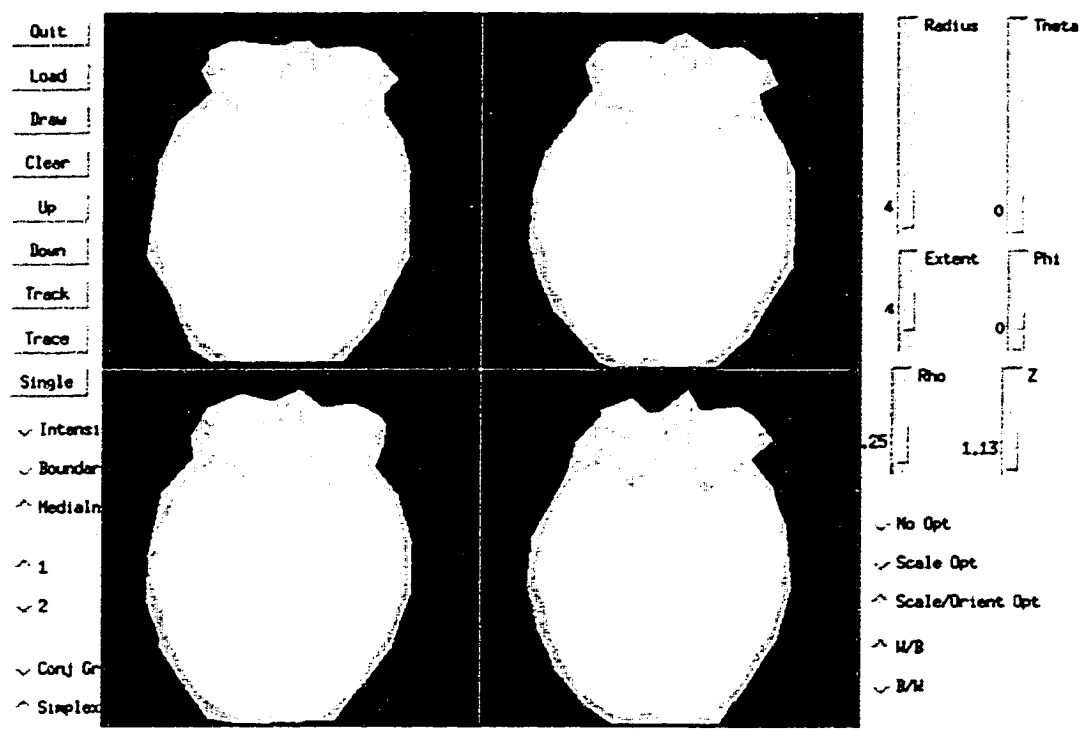


Figure 6.28: Image of cerebral vasculature (metaslice 10 - image slices 45-48)

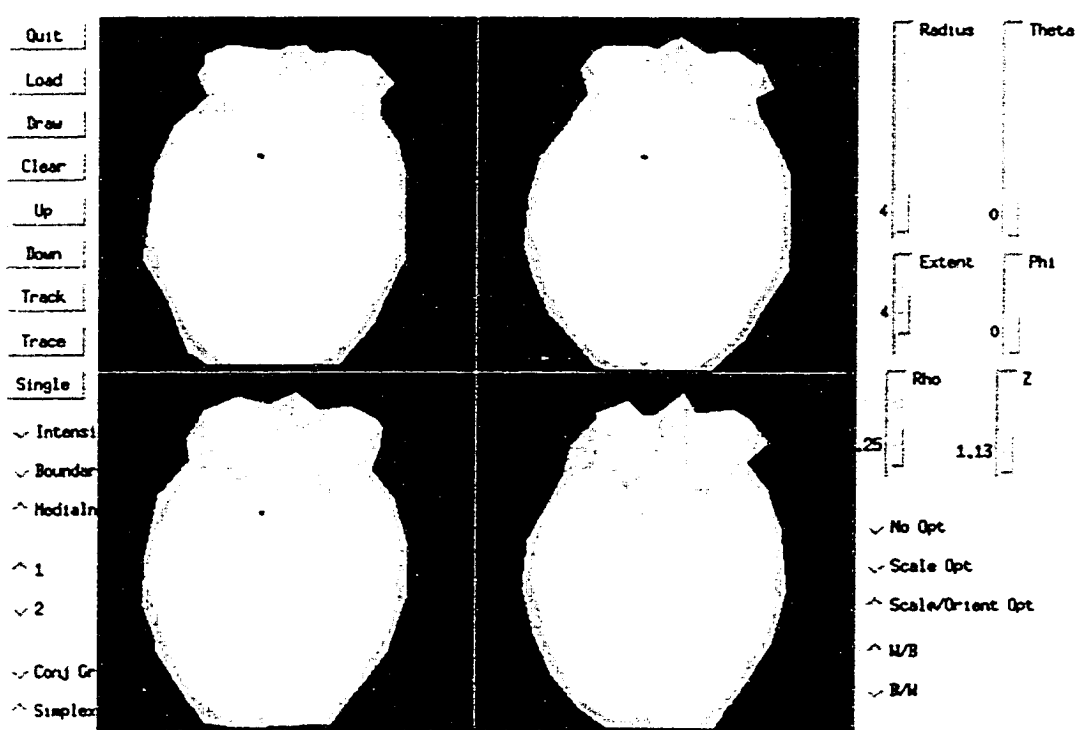


Figure 6.29: Image of cerebral vasculature (metaslice 10 - image slices 45-48) with object implied by the core superimposed in black

## Kidneys

The second medical image is of an abdomen (Figure 6.30 through Figure 6.47). The image is a CT taken from the top of the abdomen down, with slices parallel to the  $z$  plane. Each slice is 256 by 256 pixels; there are 41 slices. The figures I have included each contain four slices of the original image. The slices are arranged with the slice highest in the body in the upper left corner and are arranged left to right and then top to bottom with the lowest slice in the lower right corner. I have included only those slices of the original image which contain the core. The kidney is slightly more slab-like than tube-like, so I set the algorithm to find 2D cores. However, the distinction is slight, and I expect the core to be very narrow. It should run down the middle of the kidney, with radius proportion to the radius of the kidney and with an orientation along the minor axis of the elliptical cross section of the kidney. I used the 2-oriented, 3D Laplacian weighting function to calculate medialness. The results on the two kidneys are very different. The kidney on the left faced the problem of the indistinct boundary with the liver, and the core began to expand into the liver. I manually stopped the core traversal before the results of the liver obscured the kidney. In general, the poor contrast in the image caused problems for the core tracking, most especially in the left kidney. A further characteristic behavior is the extent of the core into the image slices above the kidney. This is a problem in core extraction encountered frequently in 2D studies. The upper end of the kidney did not stop the core traversal, further compounded by the confusion of organs near the top of the kidney. The kidney on the right displays very different behavior of the core. It stopped without my intervention, and while it describes the kidney well near the center of the organ, it also has problems near the top of the kidney, but this time stopping early. This also is due to the indistinct organ boundaries near the top of the core. Note that both cores did well in describing the kidney near the bottom end, where the kidney is more distinctly separated from the other image structures surrounding it.

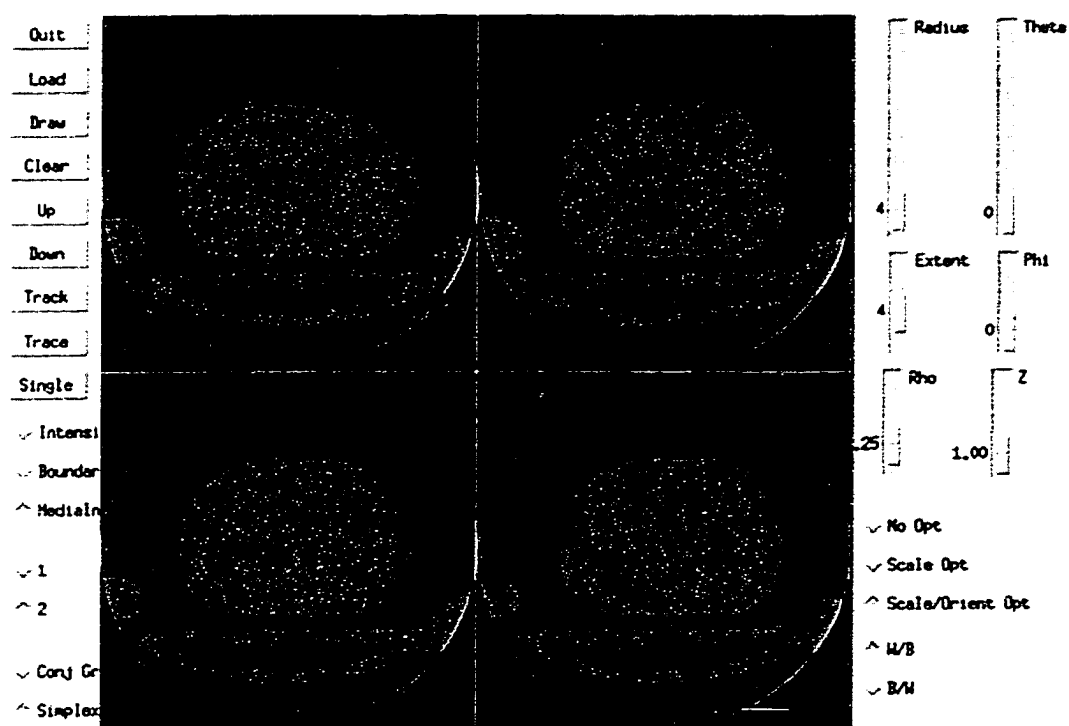


Figure 6.30: Image of abdomen around kidneys (metaslice 1 - image slices 1-4)

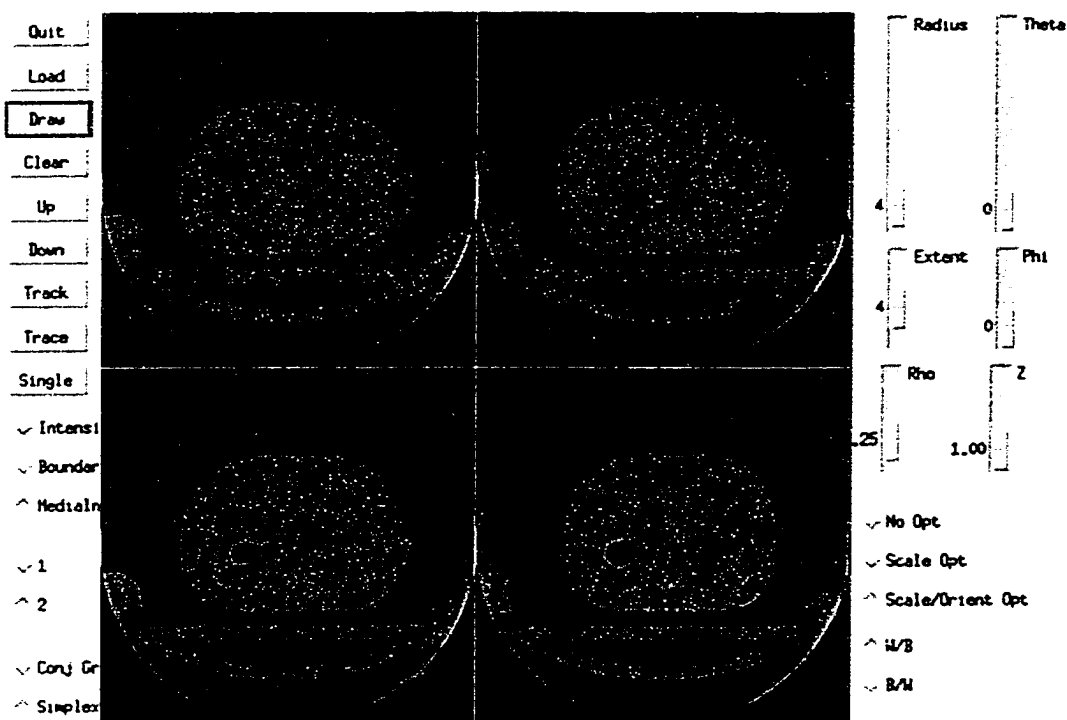


Figure 6.31: Image of abdomen around kidneys (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black

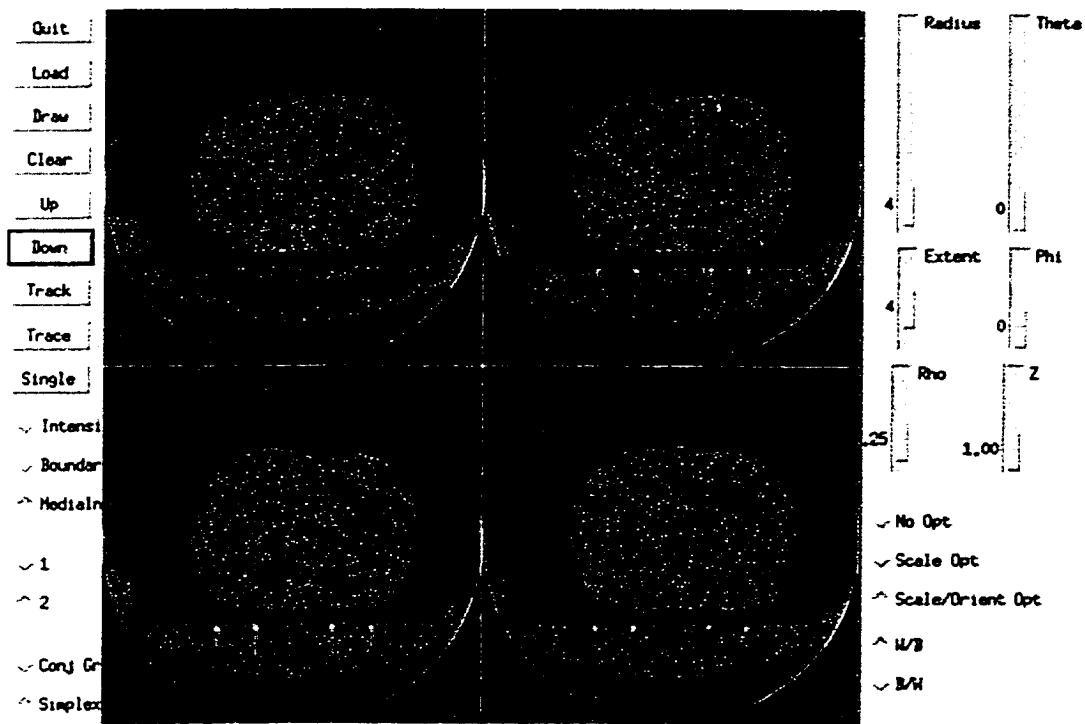


Figure 6.32: Image of abdomen around kidneys (metaslice 2 - image slices 5-8)

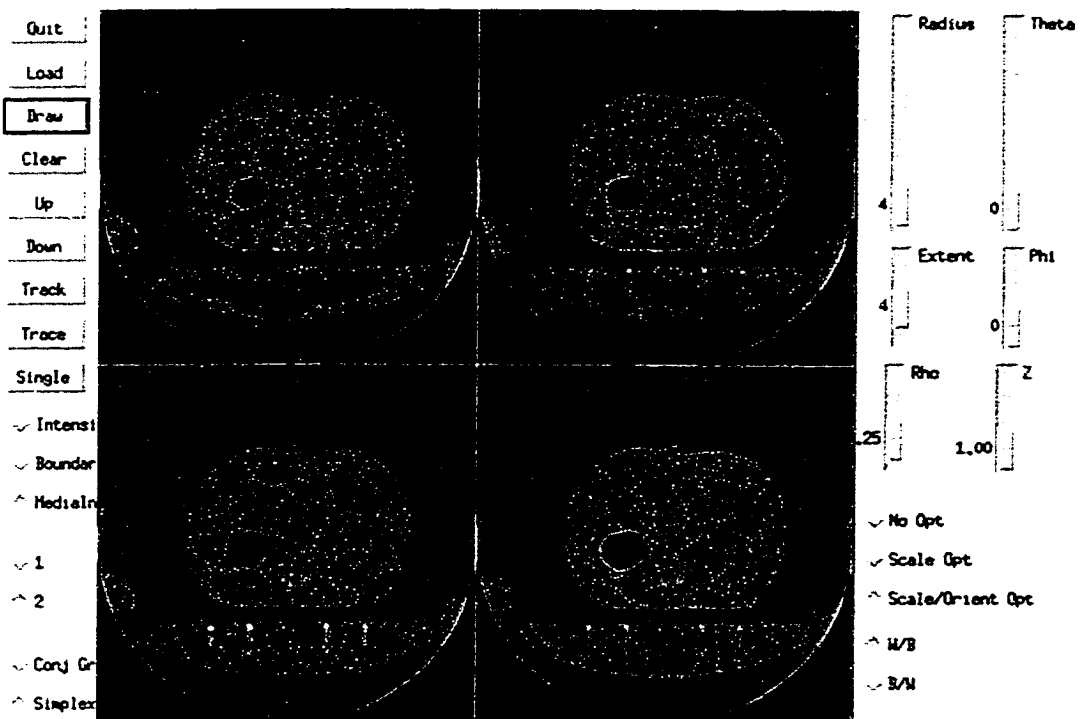


Figure 6.33: Image of abdomen around kidneys (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black

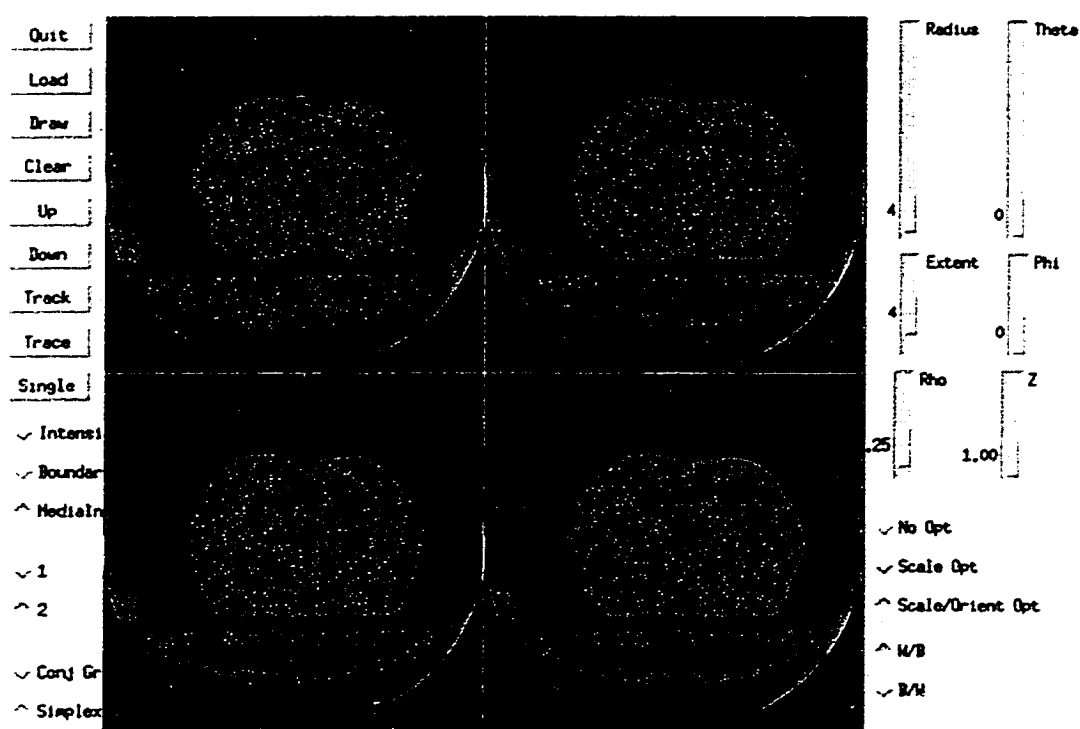


Figure 6.34: Image of abdomen around kidneys (metaslice 3 - image slices 9-12)

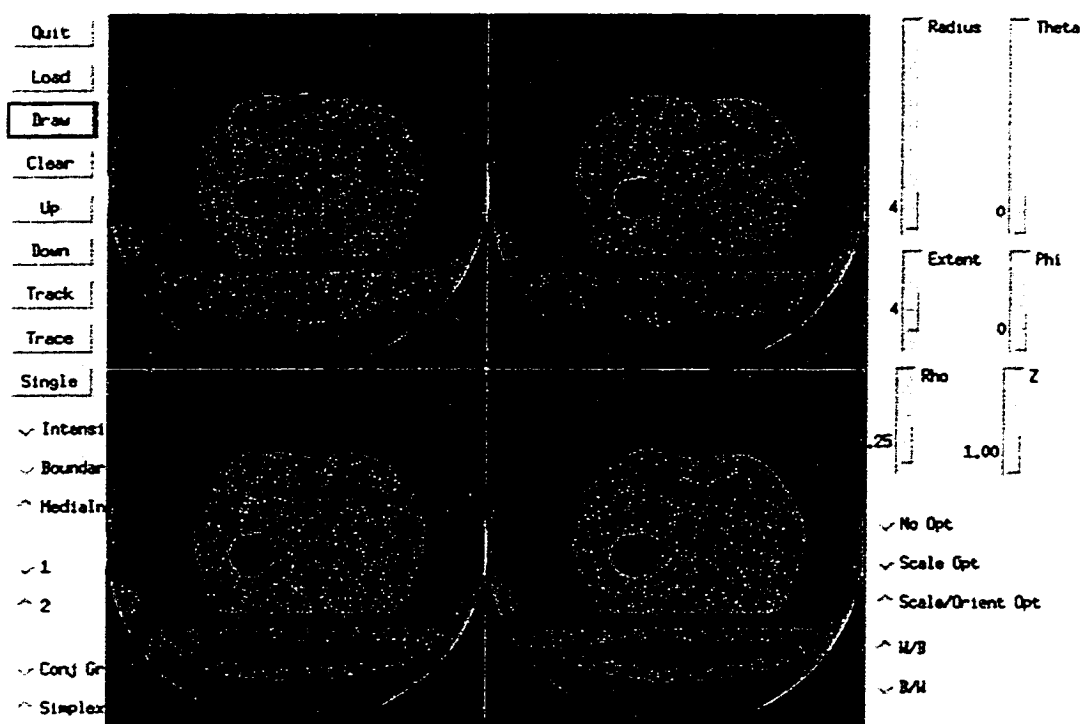


Figure 6.35: Image of abdomen around kidneys (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black

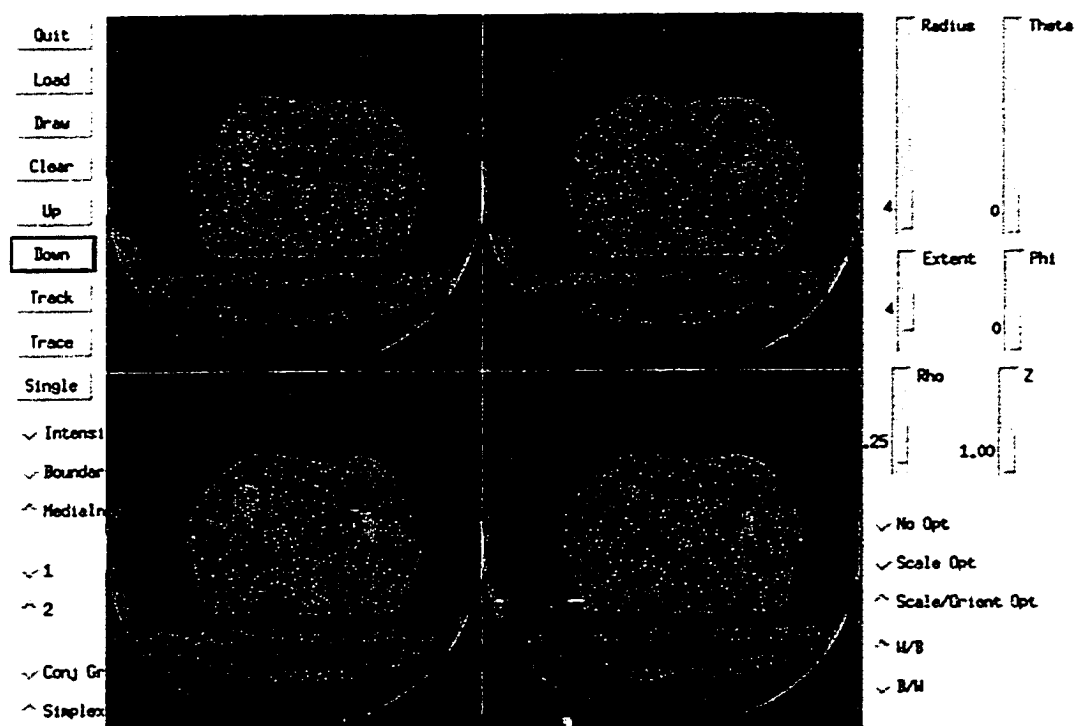


Figure 6.36: Image of abdomen around kidneys (metaslice 4 - image slices 13-16)

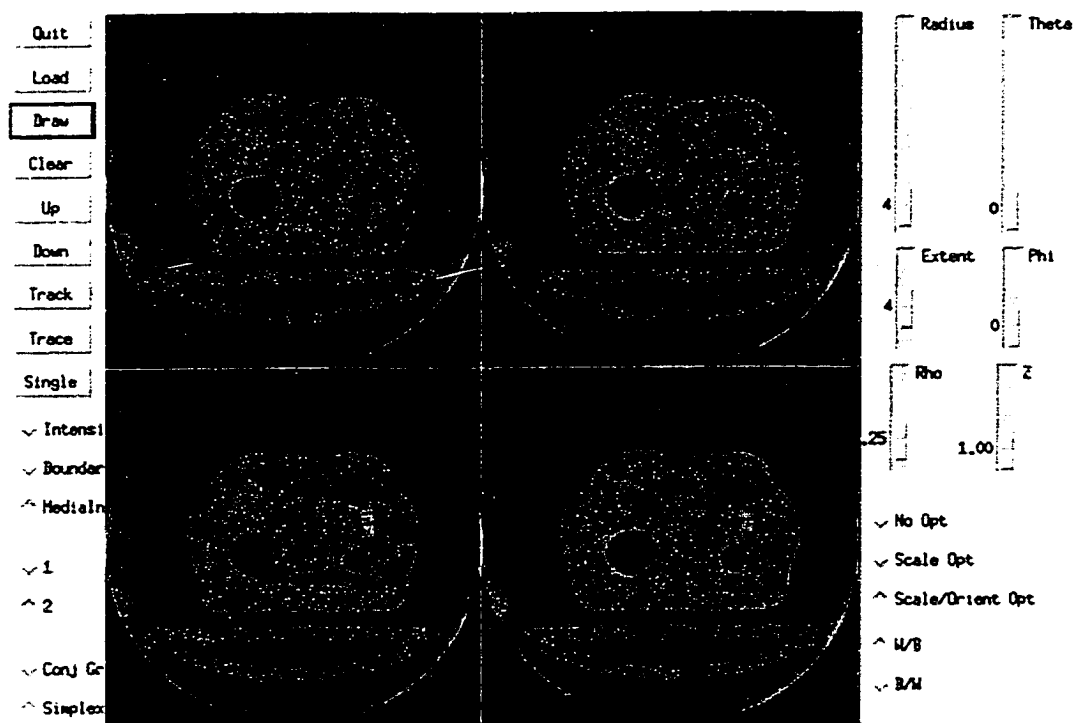


Figure 6.37: Image of abdomen around kidneys (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black



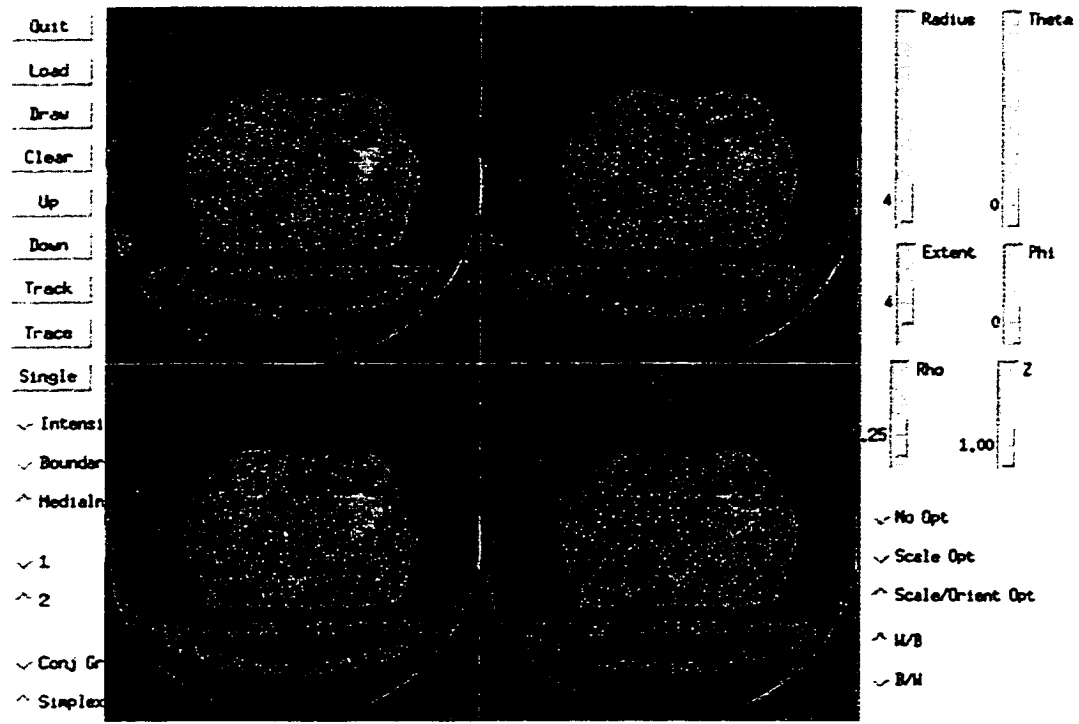


Figure 6.38: Image of abdomen around kidneys (metaslice 5 - image slices 17-20)

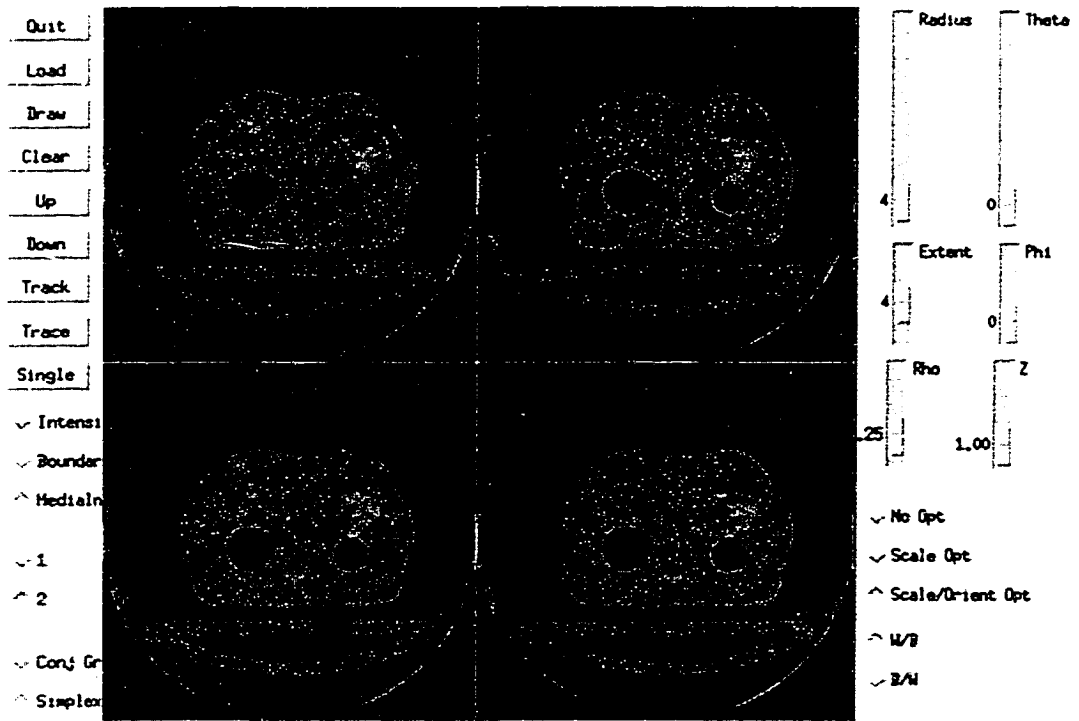


Figure 6.39: Image of abdomen around kidneys (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black

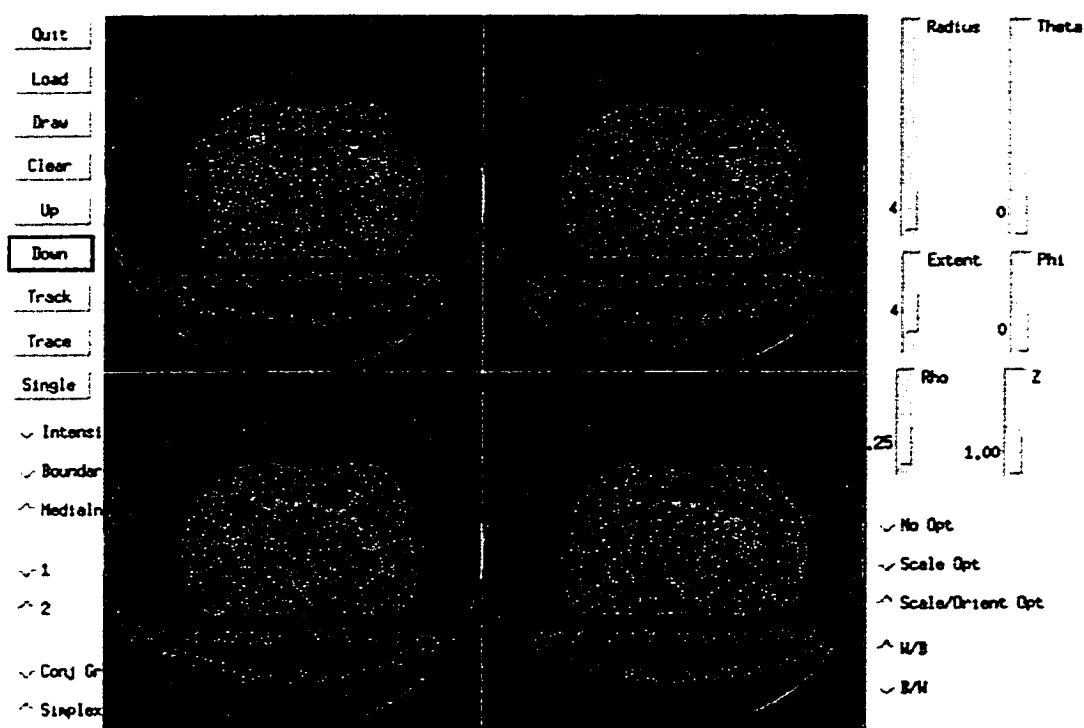


Figure 6.40: Image of abdomen around kidneys (metaslice 6 - image slices 21-24)

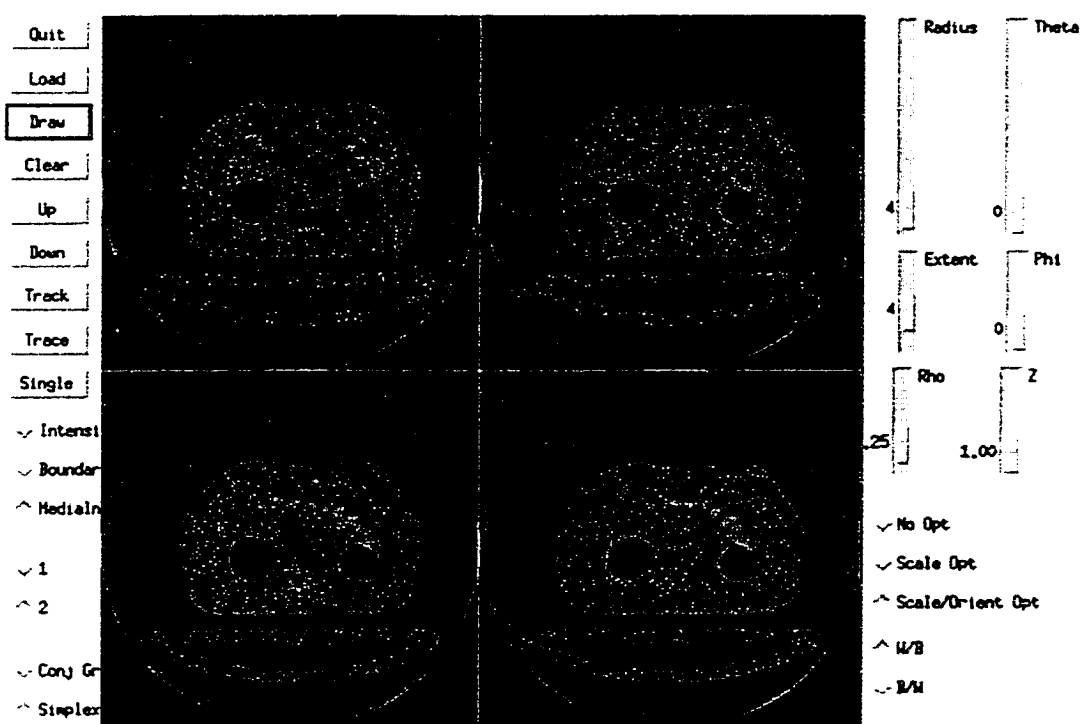


Figure 6.41: Image of abdomen around kidneys (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black

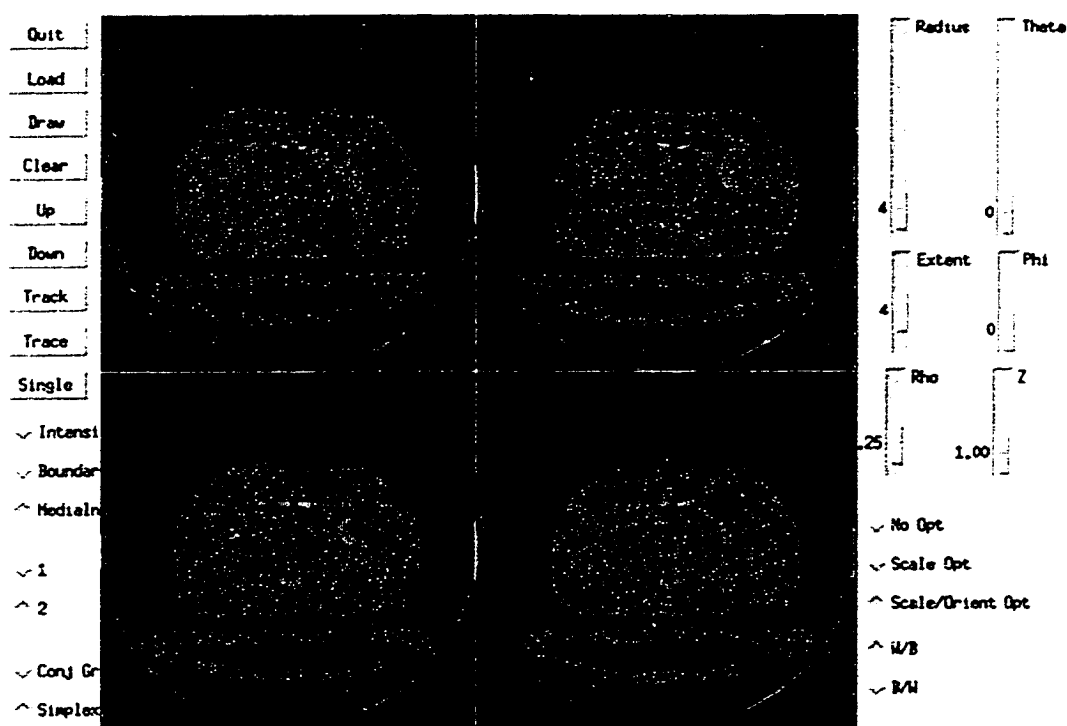


Figure 6.42: Image of abdomen around kidneys (metaslice 7 - image slices 25-28)

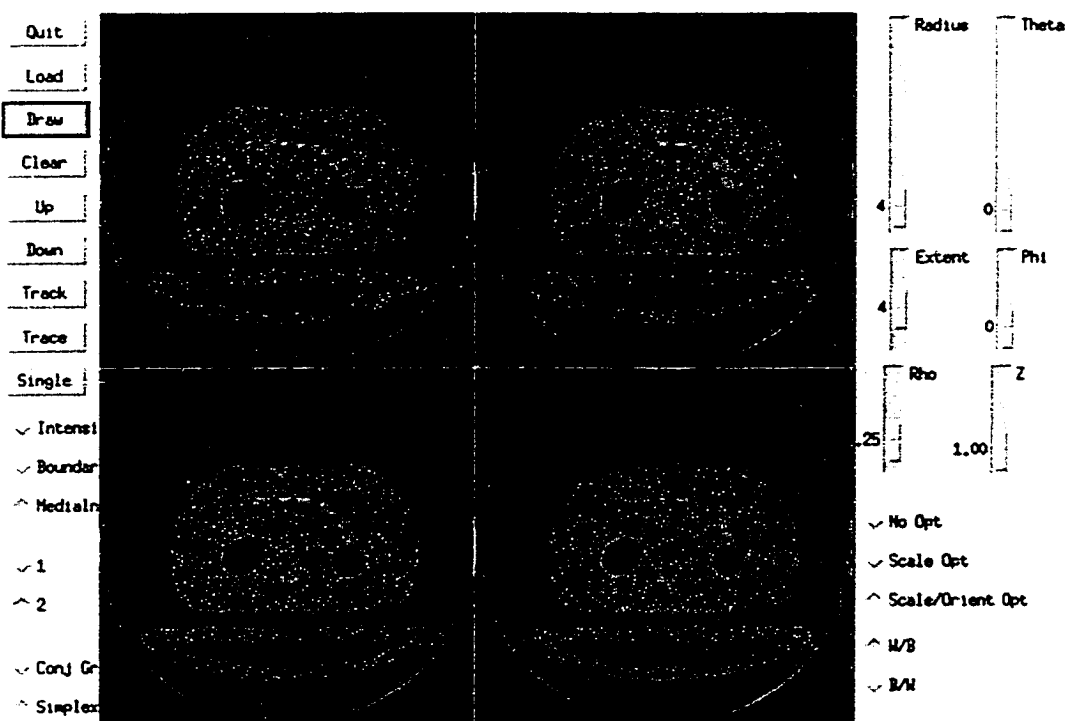


Figure 6.43: Image of abdomen around kidneys (metaslice 7 - image slices 25-28) with object implied by the core superimposed in black

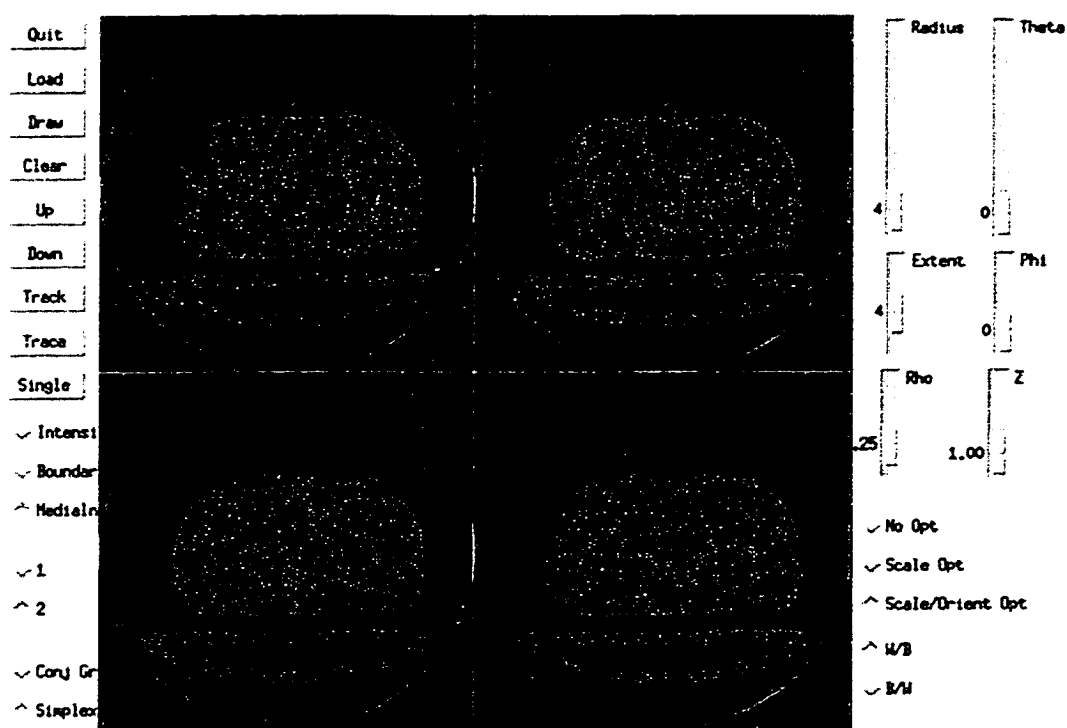


Figure 6.44: Image of abdomen around kidneys (metaslice 8 - image slices 29-32)

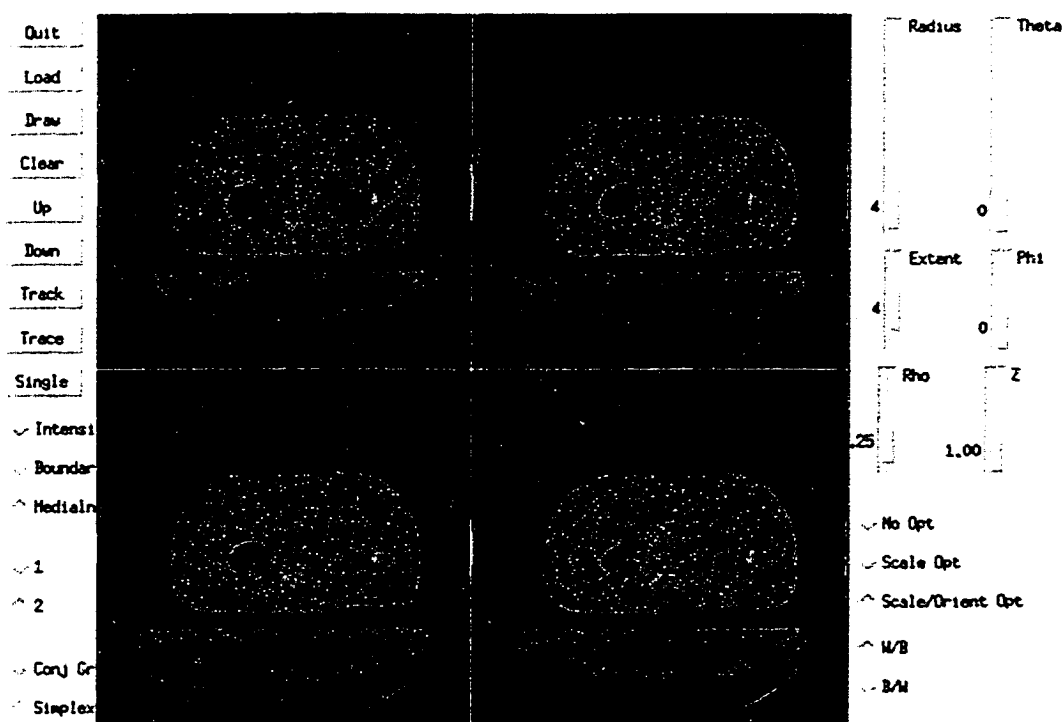


Figure 6.45: Image of abdomen around kidneys (metaslice 8 - image slices 29-32) with object implied by the core superimposed in black

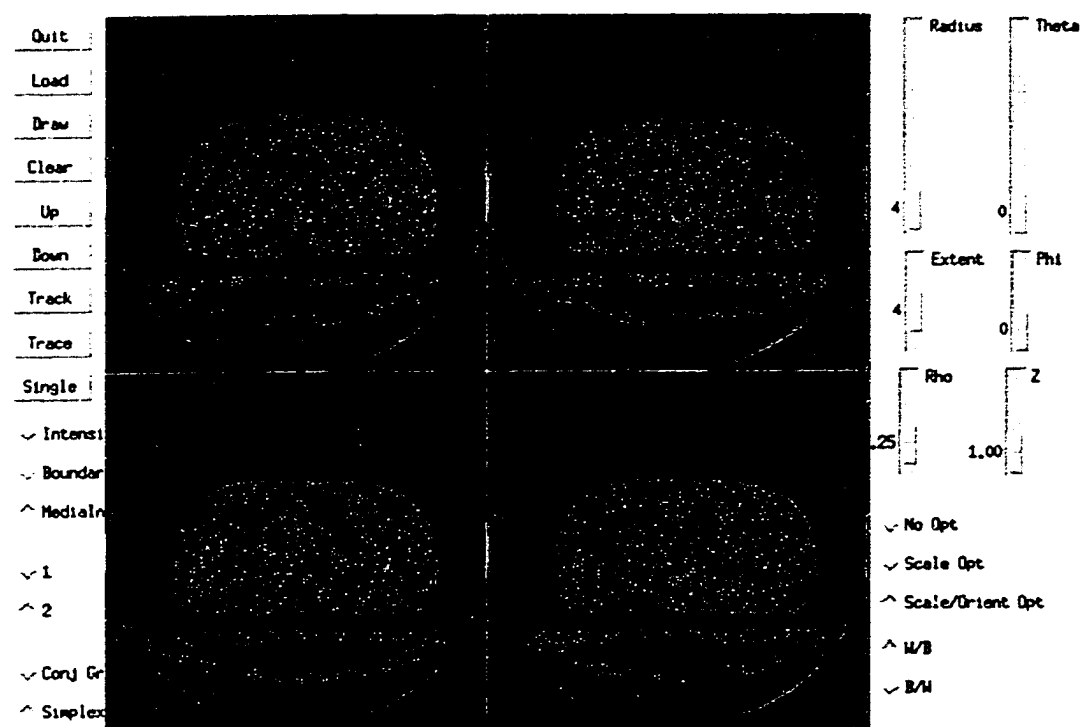


Figure 6.46: Image of abdomen around kidneys (metaslice 9 - image slices 33-36)

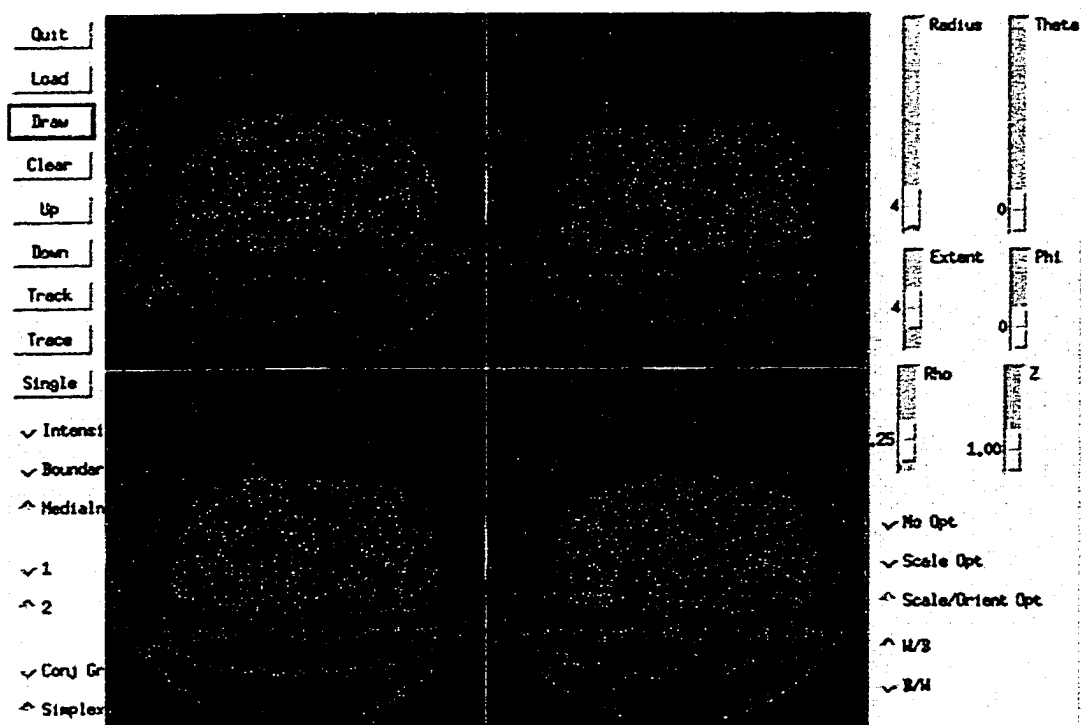


Figure 6.47: Image of abdomen around kidneys (metaslice 9 - image slices 33-36) with object implied by the core superimposed in black

## Skull

The third medical image is of a skull (Figures 6.48 through 6.67). The image is a CT scan taken from the top of the head down, with slices parallel to the  $z$  plane. Each slice is 256 by 256 pixels; there are 40 slices. Each pixel is 0.1449 mm by 0.1449 mm; the distance between slices is 0.4 mm. The figures I have included each contain four slices of the original image. The slices are arranged with the slice highest in the body in the upper left corner and are arranged left to right and then top to bottom with the lowest slice in the lower right corner. The skull is a slab-like structure, very much like the ellipsoidal shell used as a test case, so I used the 2-oriented, 3D Morse weighting function to calculate medialness and set Marching Ridges to extract 2D cores. I expect the core to run between the inner and outer side of the skull, be of nearly constant radius and have orientation normal to the surface of the skull. Because the marching ridges algorithm has trouble near the ends of objects and where small scale structures become complicated, I limited the search for core points to 4000 points; they took a little over an hour to generate. I performed such searches twice, once beginning on the left side of the skull and once beginning on the right side of the skull. The core follows the skull very well except where the skull becomes extremely thin and where its structure becomes much more complicated. The radius of the core points of the skull varies between 0.5 pixels and 1.6 pixels. This is an extremely small radius, and the variation is likely due to pixel noise. The orientation is mostly aligned with the  $yz$  plane, becoming more horizontal as the core approaches the top of the skull, as expected. The interslice distance of this image is also a problem for the core tracker: note that the top of the skull does not have a top and a bottom; the interslice distance is so great that only a single slice intersects the top of the skull. The core finder relies on opposite edges, so neither core extended to the top of the skull. The breaks in the core are due to the extreme narrowness of the skull at those points. At radii so small, the weighting functions applied to the image are only slightly larger than a pixel and do not provide reliable results for ridge tracking. The large black blob is what happens when the core tracking encountered complicated small scale structure and "got lost".

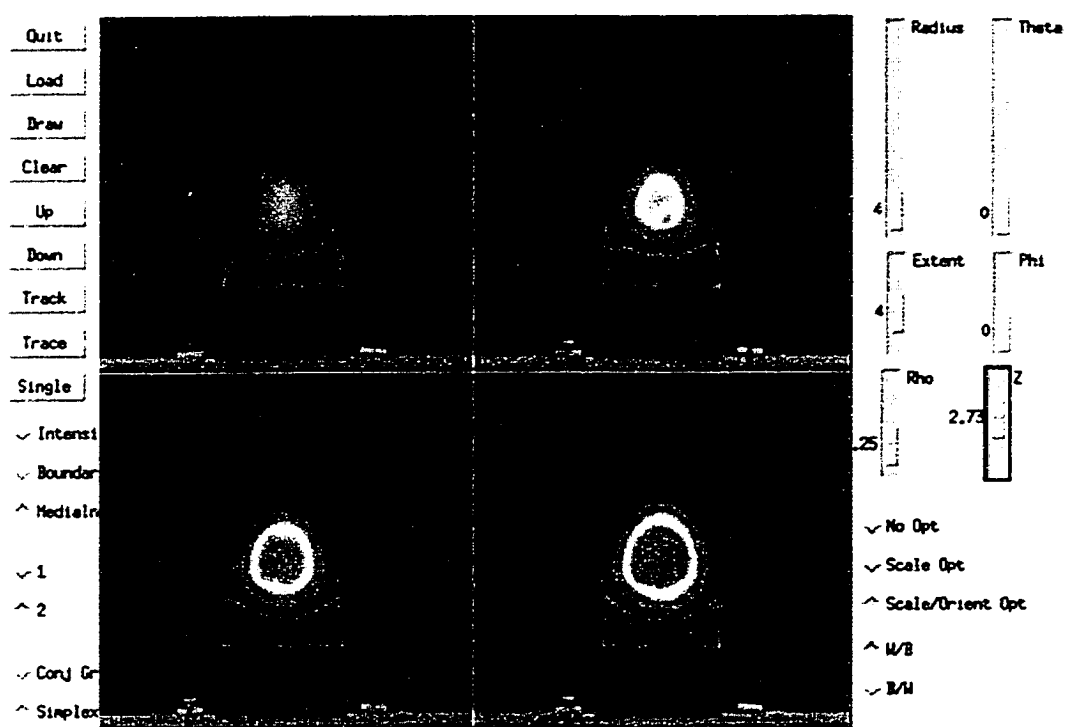


Figure 6.48: Image of skull (metaslice 1 - image slices 1-4)

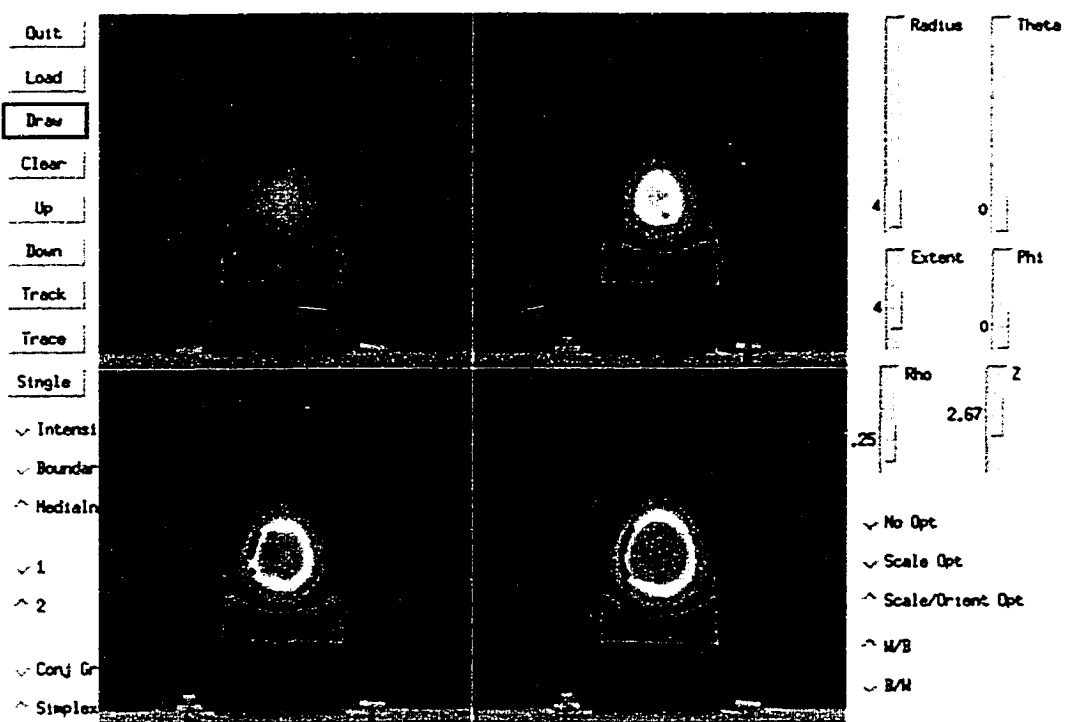


Figure 6.49: Image of skull (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black

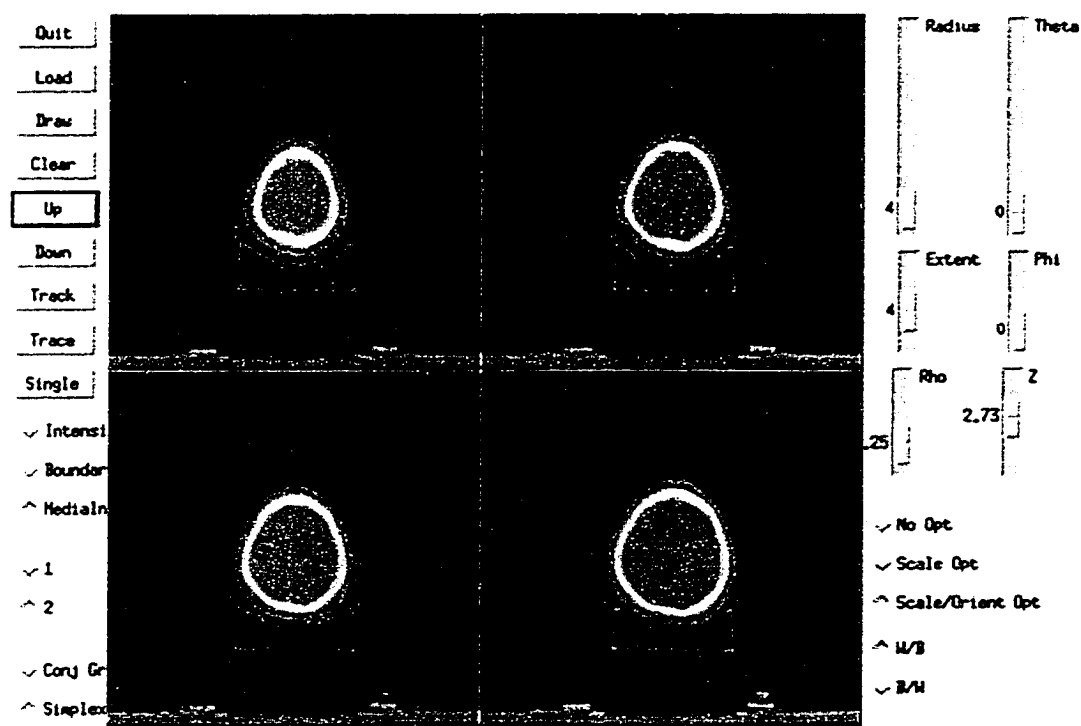


Figure 6.50: Image of skull (metaslice 2 - image slices 5-8)

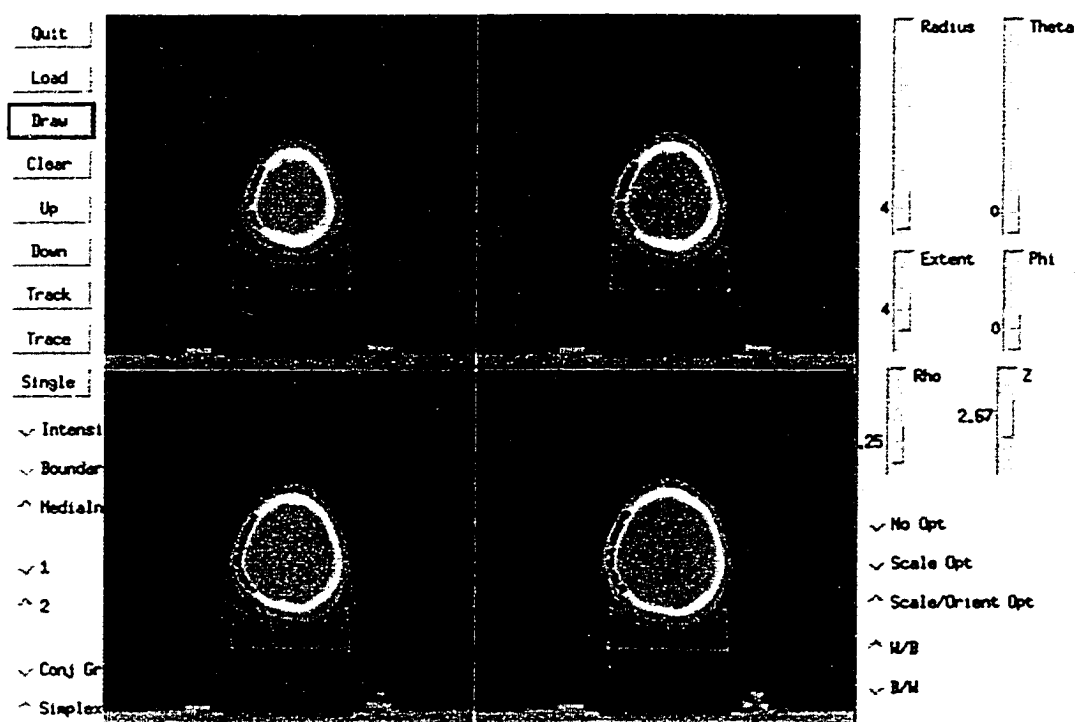


Figure 6.51: Image of skull (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black



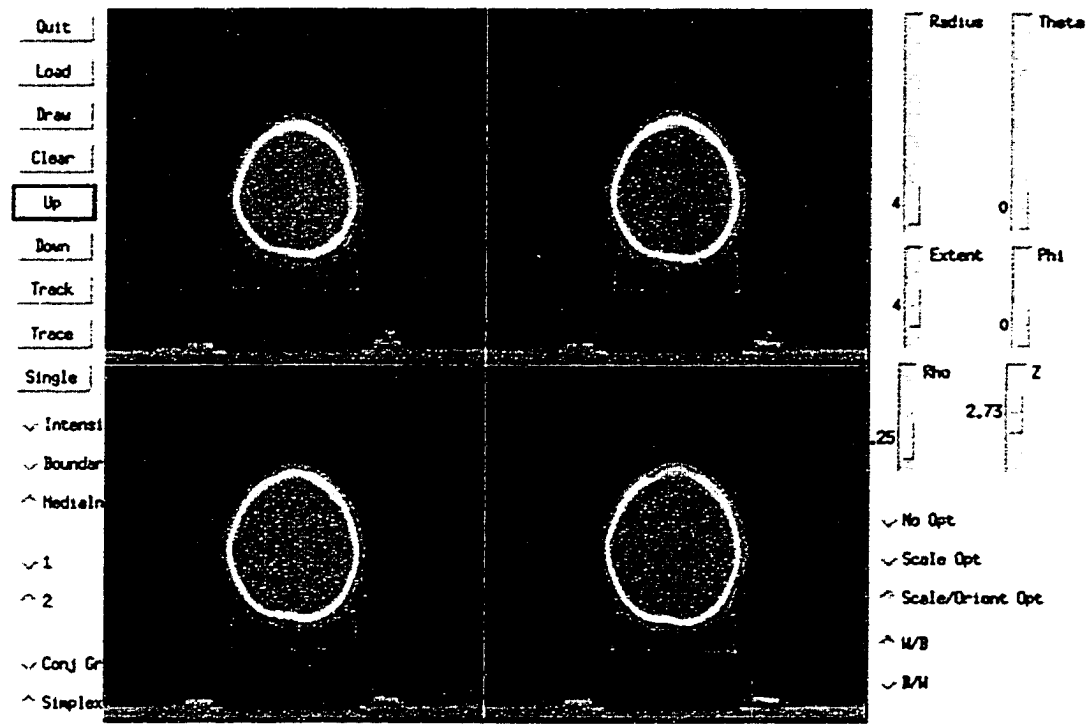


Figure 6.52: Image of skull (metaslice 3 - image slices 9-12)

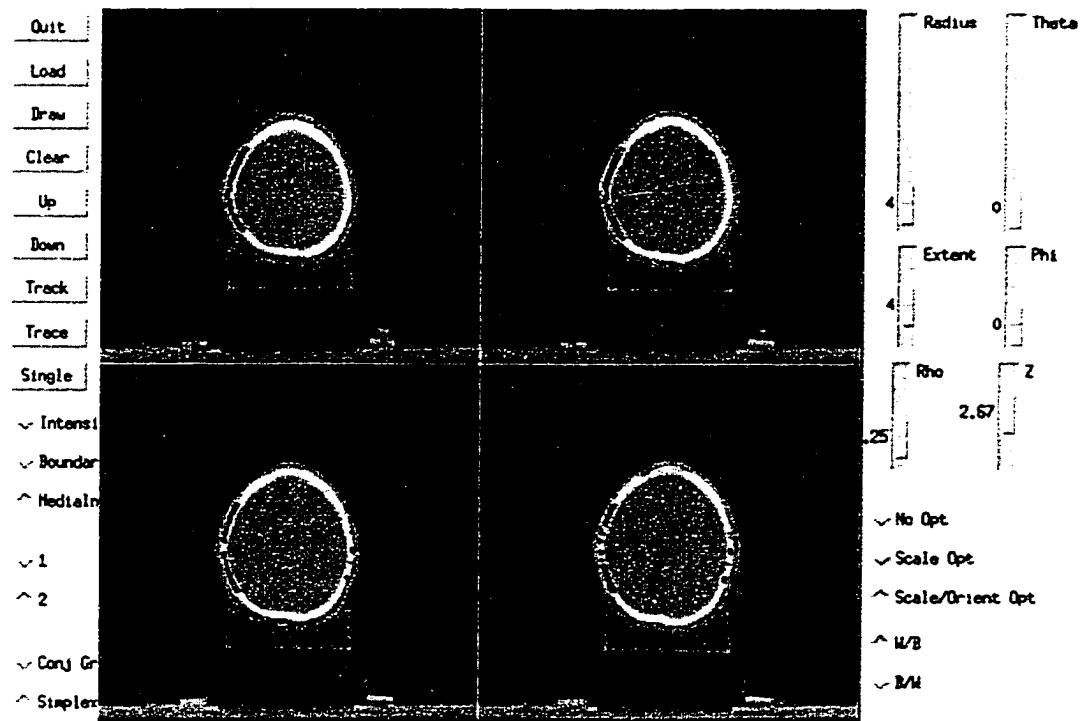


Figure 6.53: Image of skull (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black

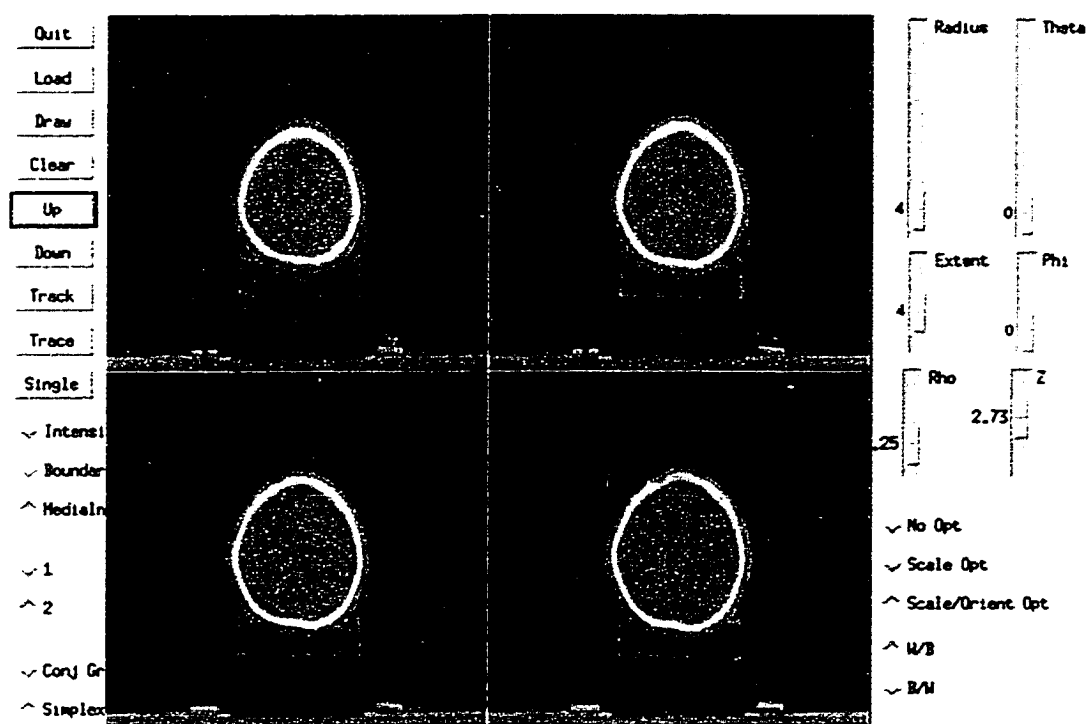


Figure 6.54: Image of skull (metaslice 4 - image slices 13-16)

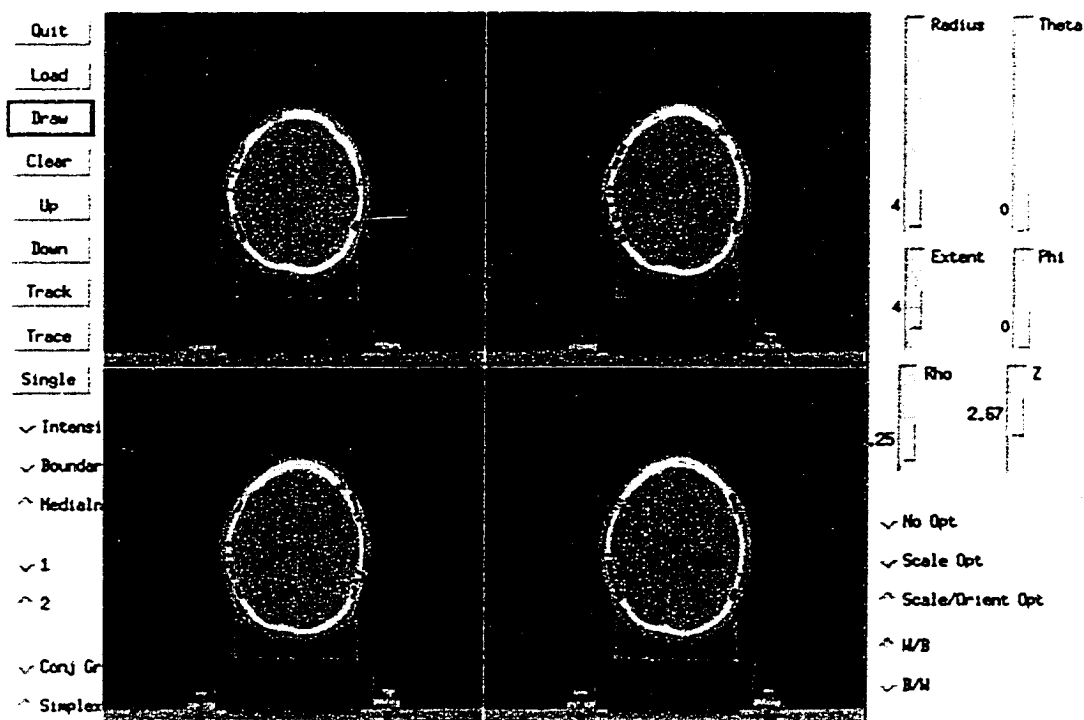


Figure 6.55: Image of skull (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black

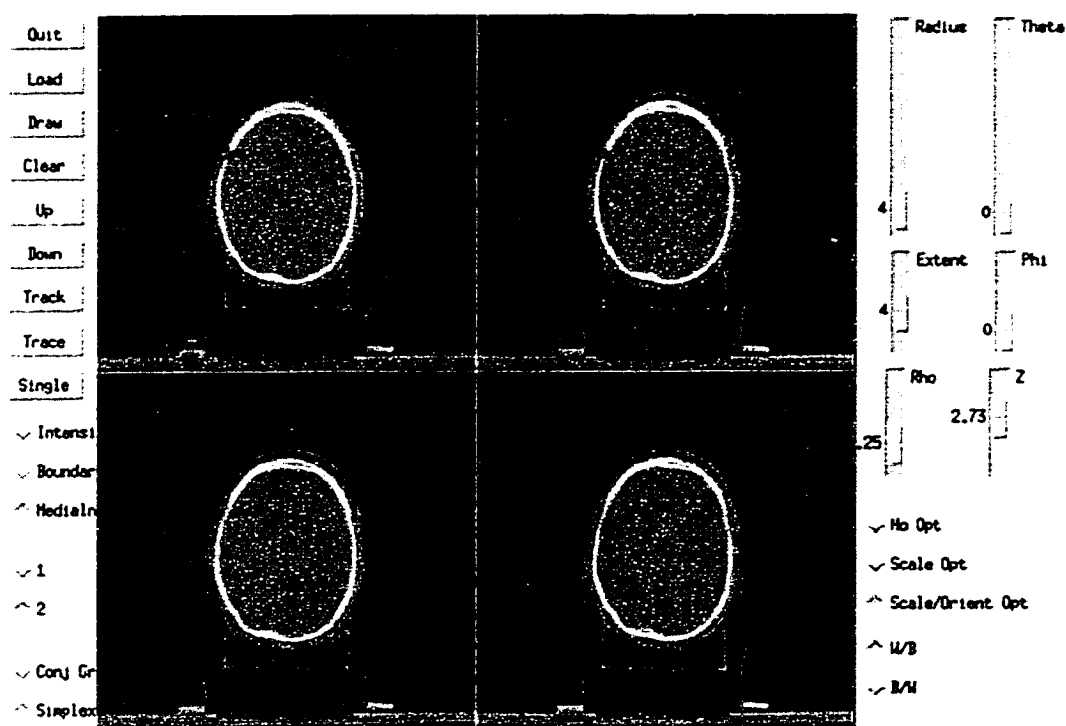


Figure 6.56: Image of skull (metaslice 5 - image slices 17-20)

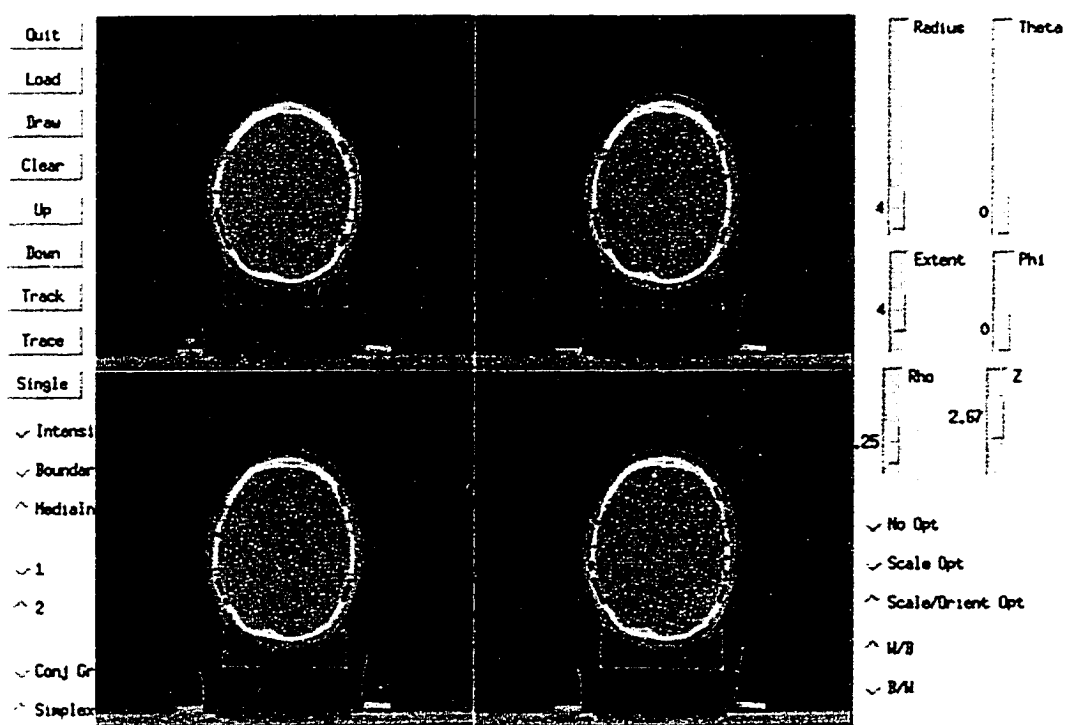


Figure 6.57: Image of skull (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black

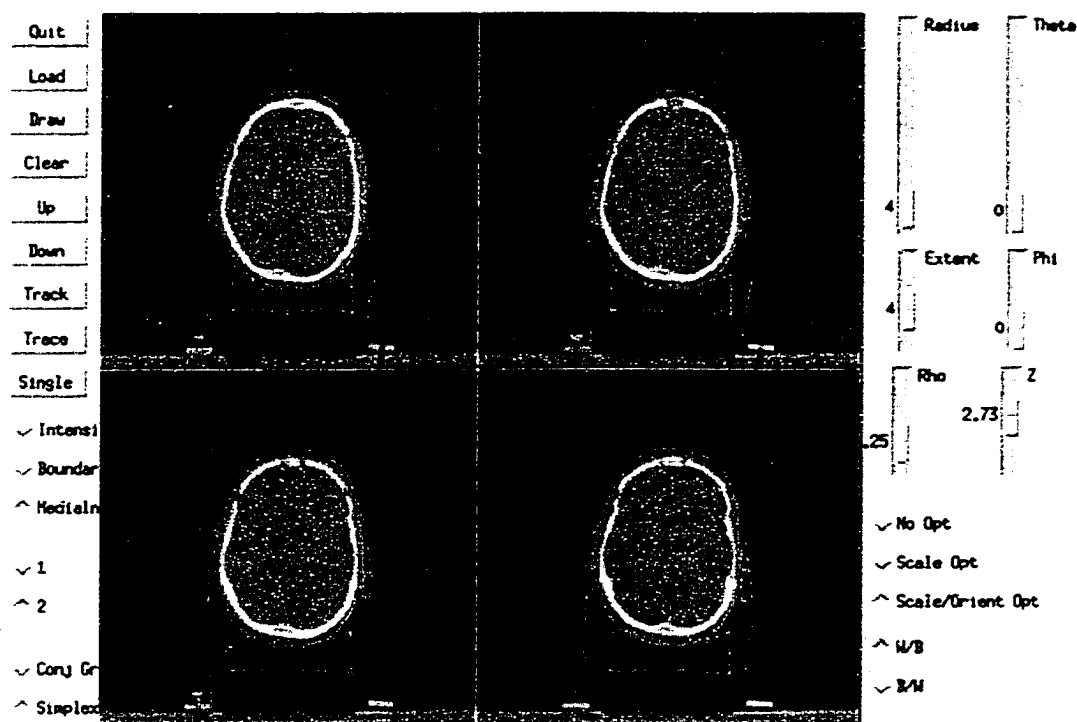


Figure 6.58: Image of skull (metaslice 6 - image slices 21-24)

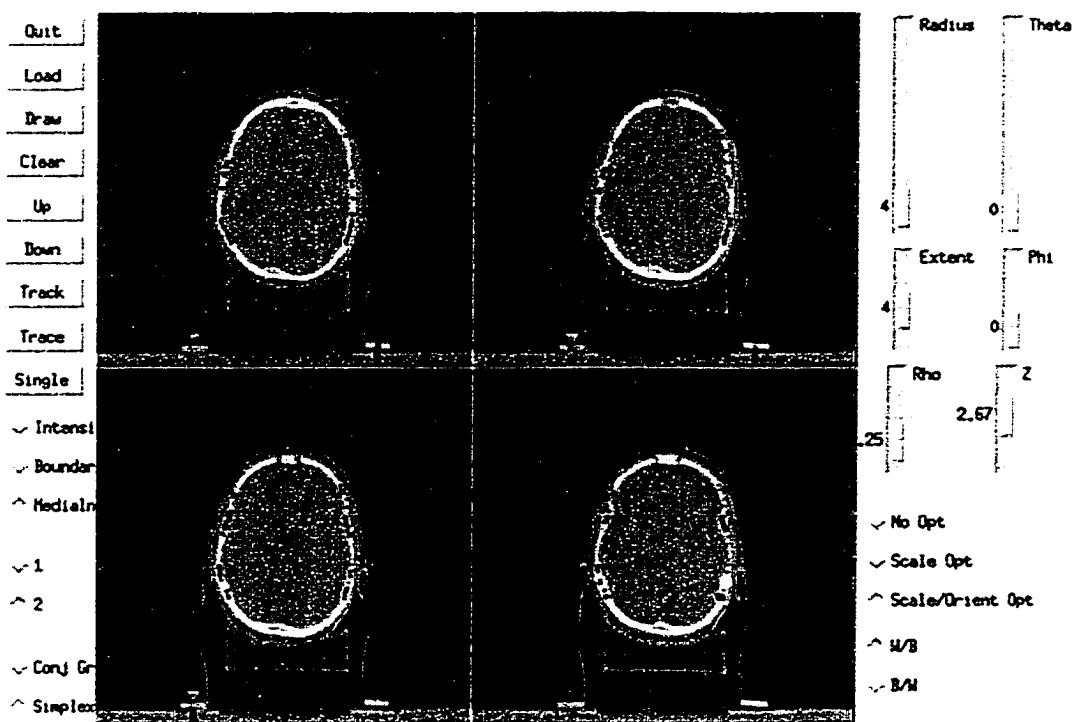


Figure 6.59: Image of skull (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black

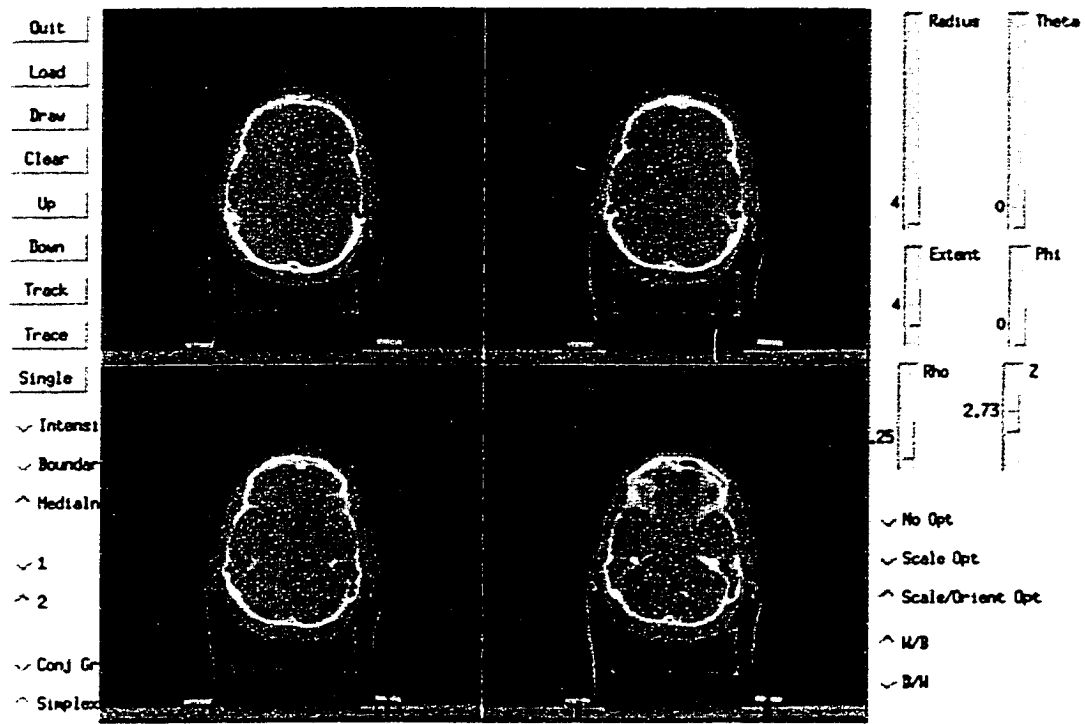


Figure 6.60: Image of skull (metaslice 7 - image slices 25-28)

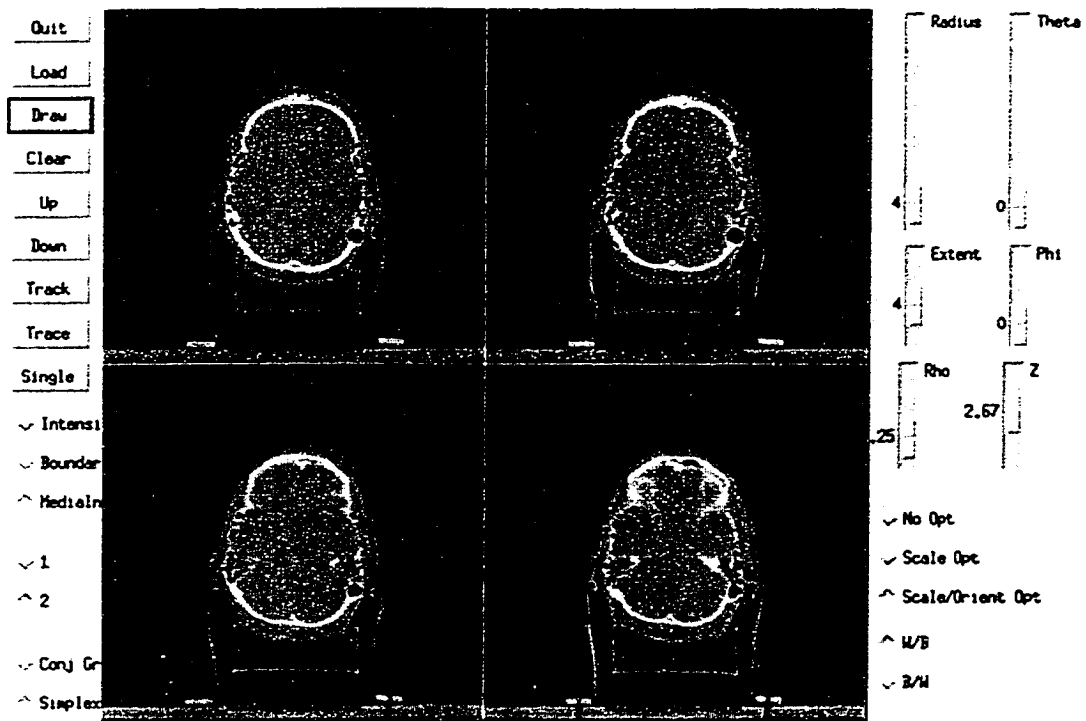


Figure 6.61: Image of skull (metaslice 7 - image slices 25-28) with object implied by the core superimposed in black

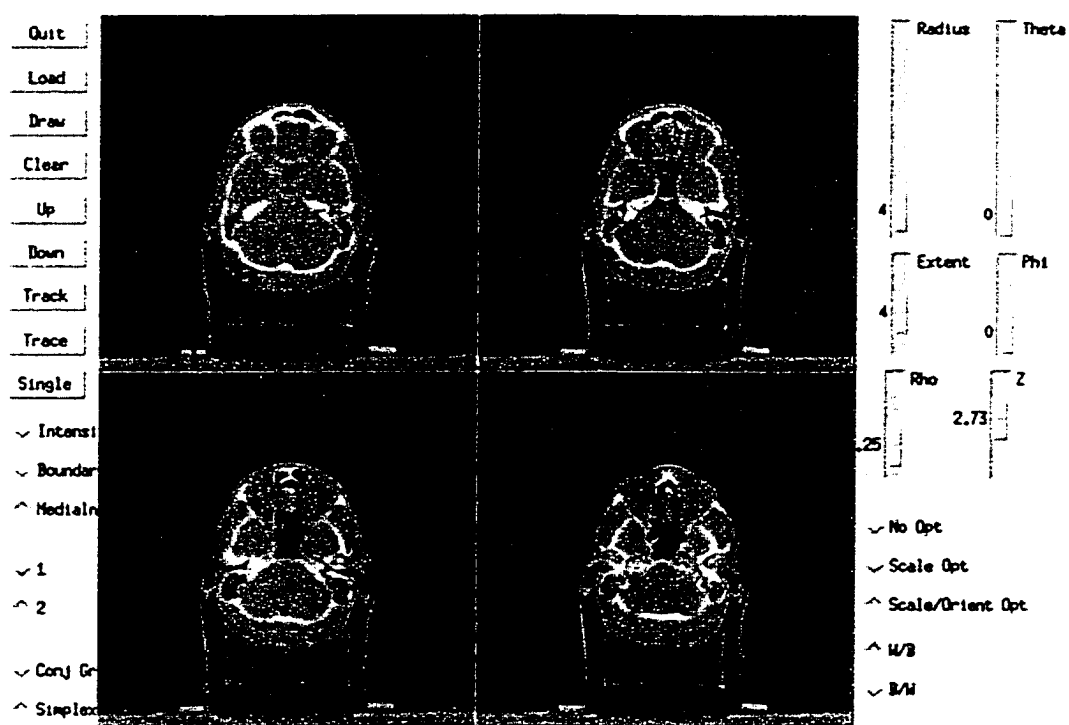


Figure 6.62: Image of skull (metaslice 8 - image slices 29-32)

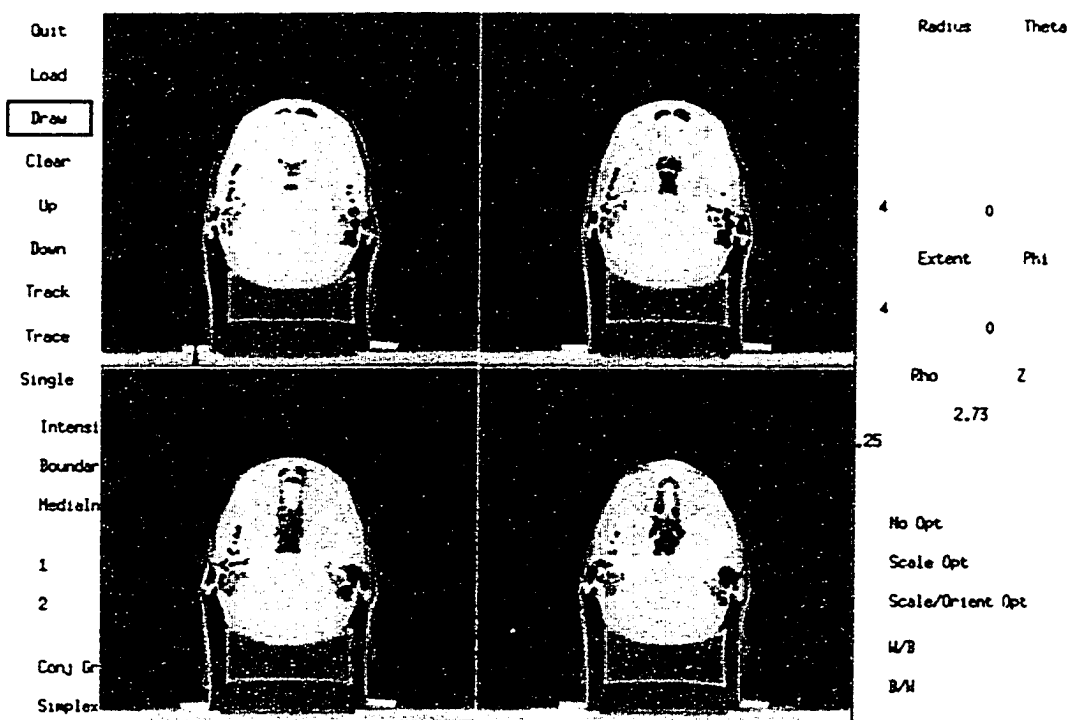


Figure 6.63: Image of skull (metaslice 8 - image slices 29-32) with object implied by the core superimposed in black

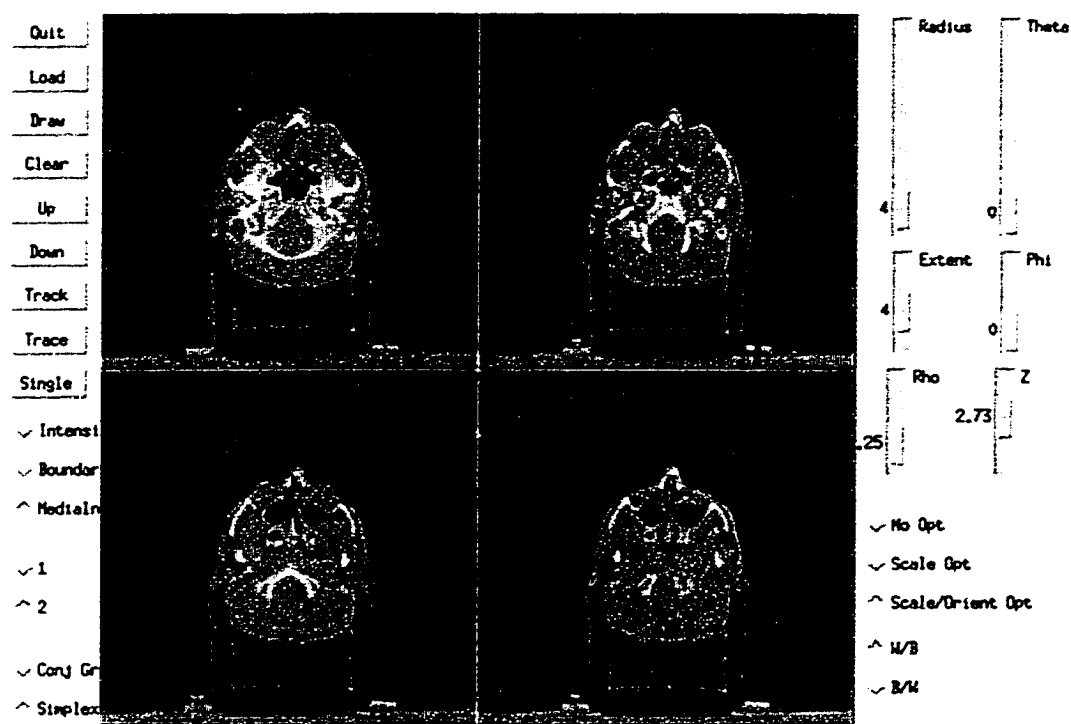


Figure 6.64: Image of skull (metaslice 9 - image slices 33-36)

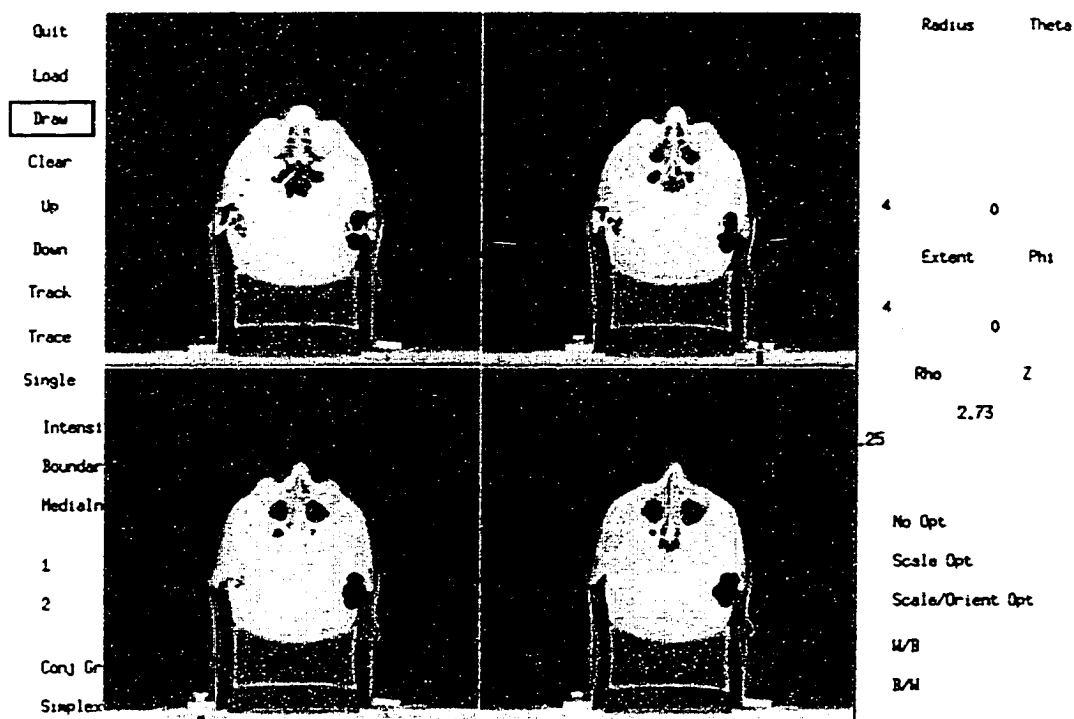


Figure 6.65: Image of skull (metaslice 9 - image slices 33-36) with object implied by the core superimposed in black

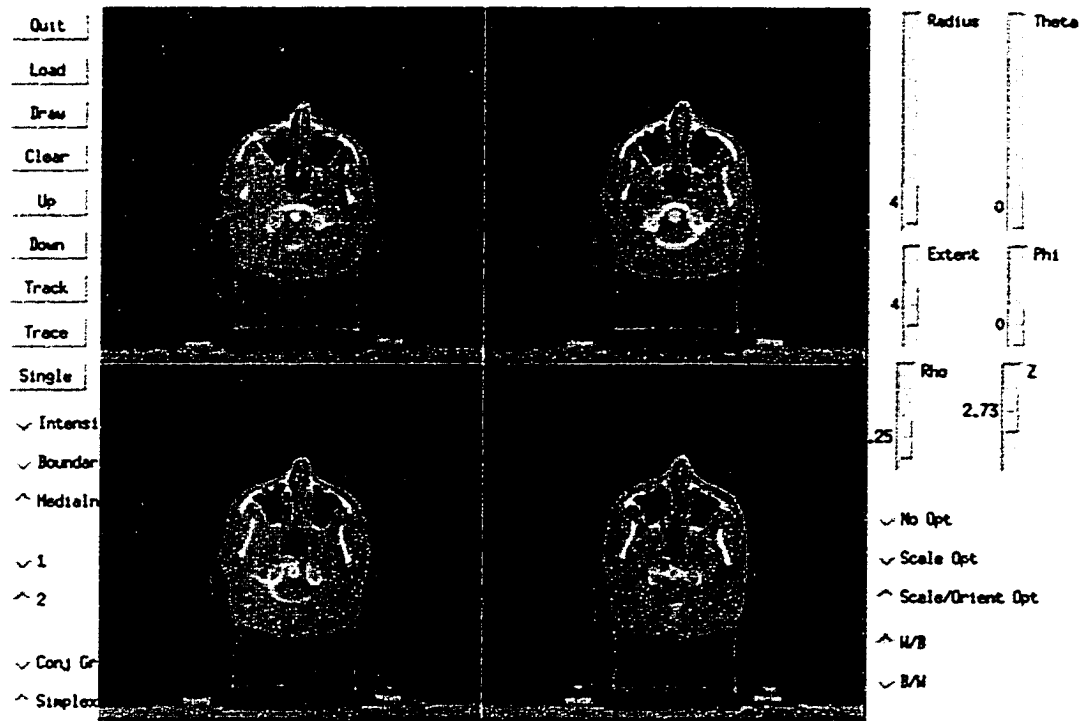


Figure 6.66: Image of skull (metaslice 10 - image slices 37-40)

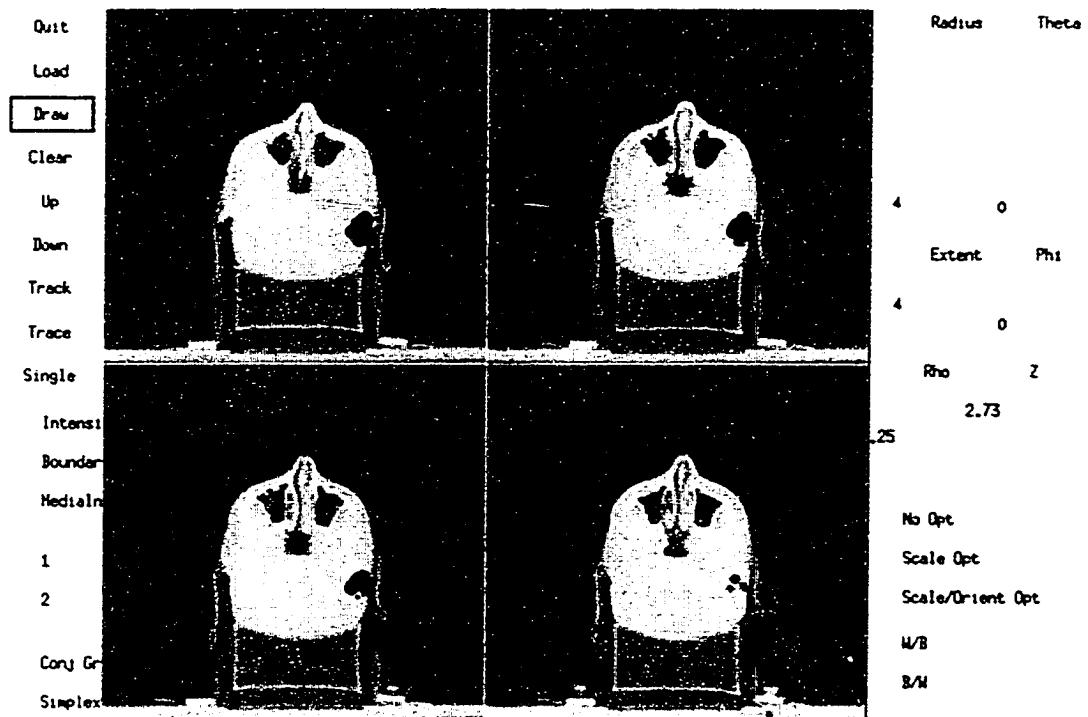


Figure 6.67: Image of skull (metaslice 10 - image slices 37-40) with object implied by the core superimposed in black



## Brain Ventricle

The third medical image is of a brain (Figures 6.68 through 6.79). The image is an MRI taken from the top of the head down, with slices parallel to the  $z$  plane. Each slice is 256 by 256 pixels; there are 50 slices. Each pixel is 0.859 mm by 0.859 mm; the distance between slices is 1.4 mm. The figures I have included each contain four slices of the original image, with decreasing  $z$  left to right and then top to bottom. The ventricle looks remarkably like a Klingon Bird of Prey with the bridge split in half. For those without Star Trek knowledge, the ventricle is a dark, C-shaped slab with a pair of spindly legs. I extracted cores from the upper, C-shaped slab, so used the 2-oriented, 3D Morse weighting function to calculate medialness and set Marching Ridges to extract 2D cores. I expect the core to run between the inner and outer sides of the ventricle and be of nearly constant radius. I extracted cores from both the left and right ventricle, expecting that the core would have trouble with the relatively low contrast of the image and endstopping. The results show that the core tracking did very well in certain parts of the ventricle, but not in others where it extended beyond the boundaries of the ventricle and became lost. Consistent with expectations, Marching Ridge has trouble with the boundary of the ventricle for two major reasons: 1) the oriented Morse kernel does not register ends of object well, and 2) the gradually decreasing radius has the effect of creating weighting functions that are so small as to be unstable. Consistent with expectations, the core extended past the end of the ventricle. Once the ridge tracking has extended beyond the limits of the ventricle, the noise and complexity of the rest of the image make the behavior of the ridge tracking essentially unpredictable.

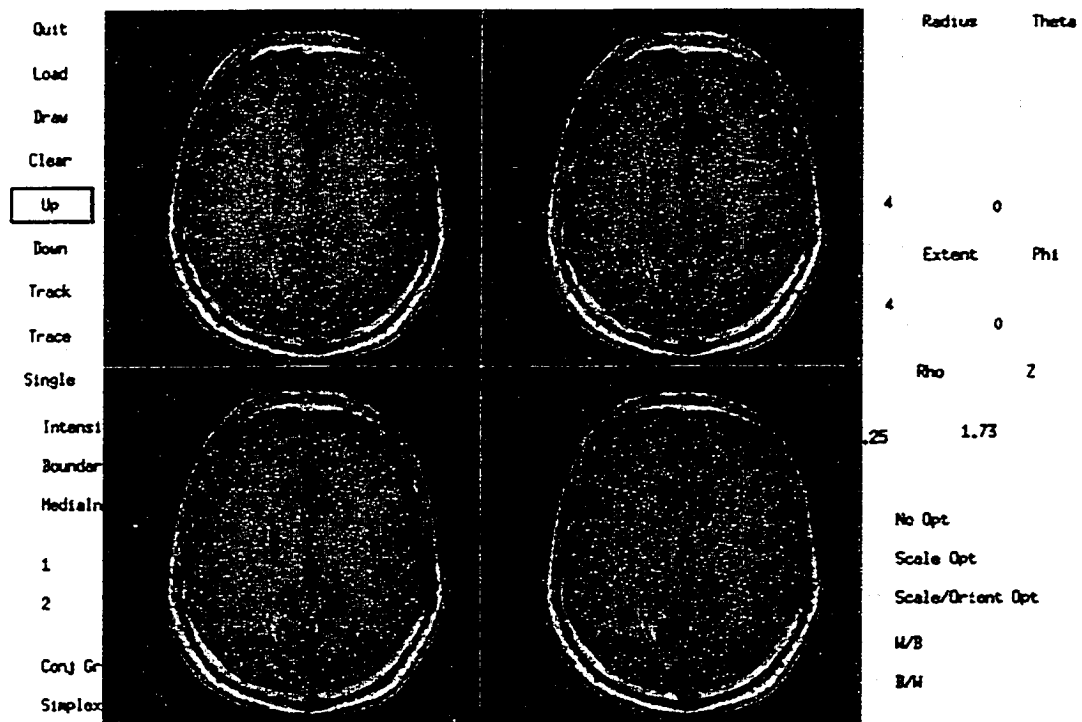


Figure 6.68: Image of ventricles (metaslice 1 - image slices 1-4)

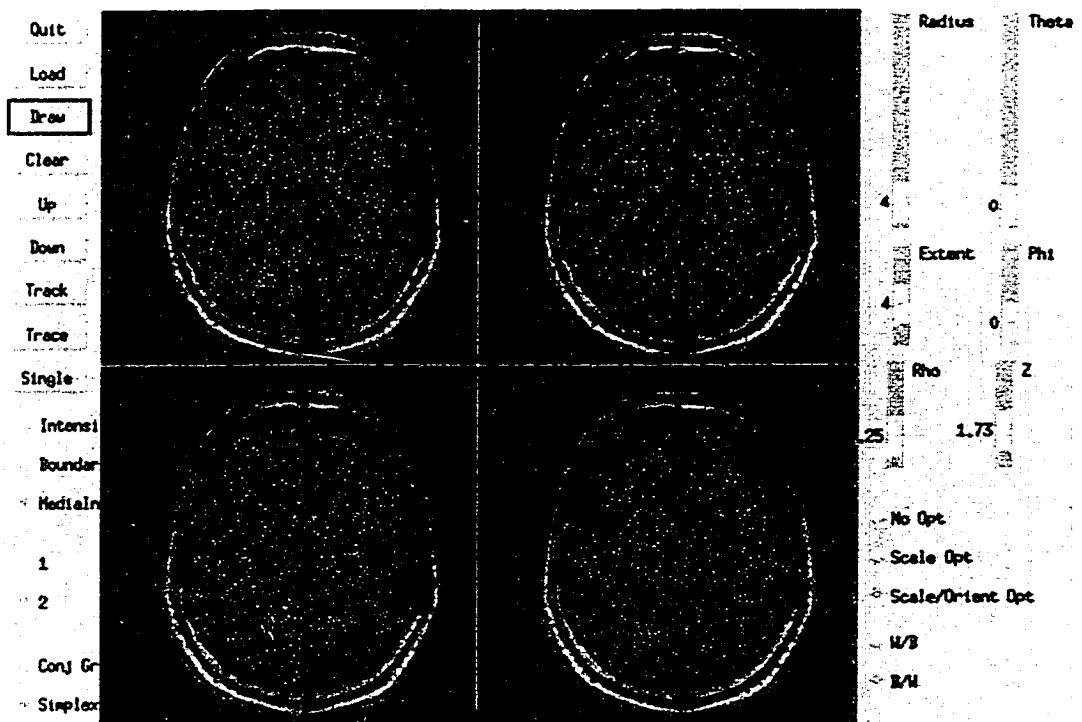


Figure 6.69: Image of ventricles (metaslice 1 - image slices 1-4) with object implied by the core superimposed in black

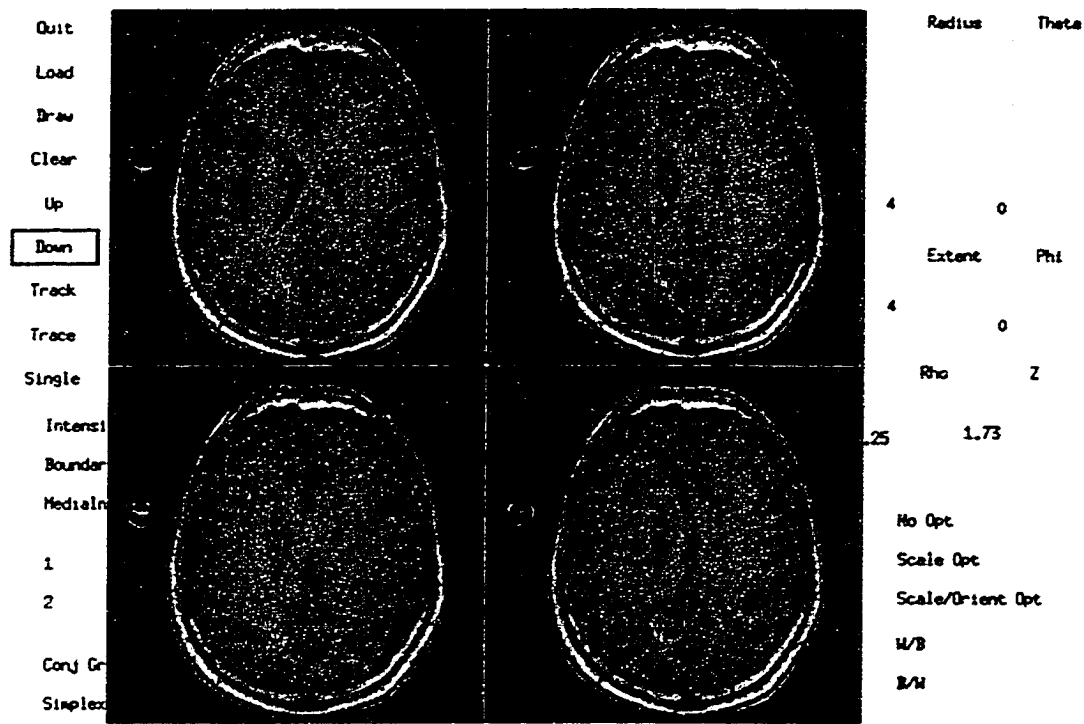


Figure 6.70: Image of ventricles (metaslice 2 - image slices 5-8)

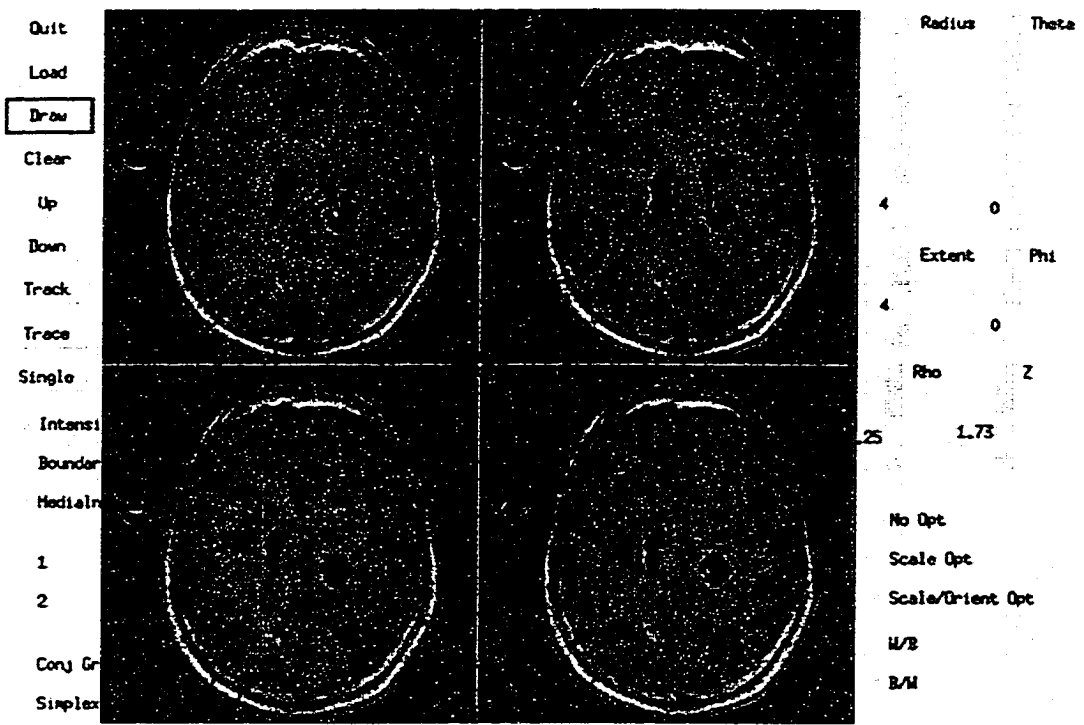


Figure 6.71: Image of ventricles (metaslice 2 - image slices 5-8) with object implied by the core superimposed in black

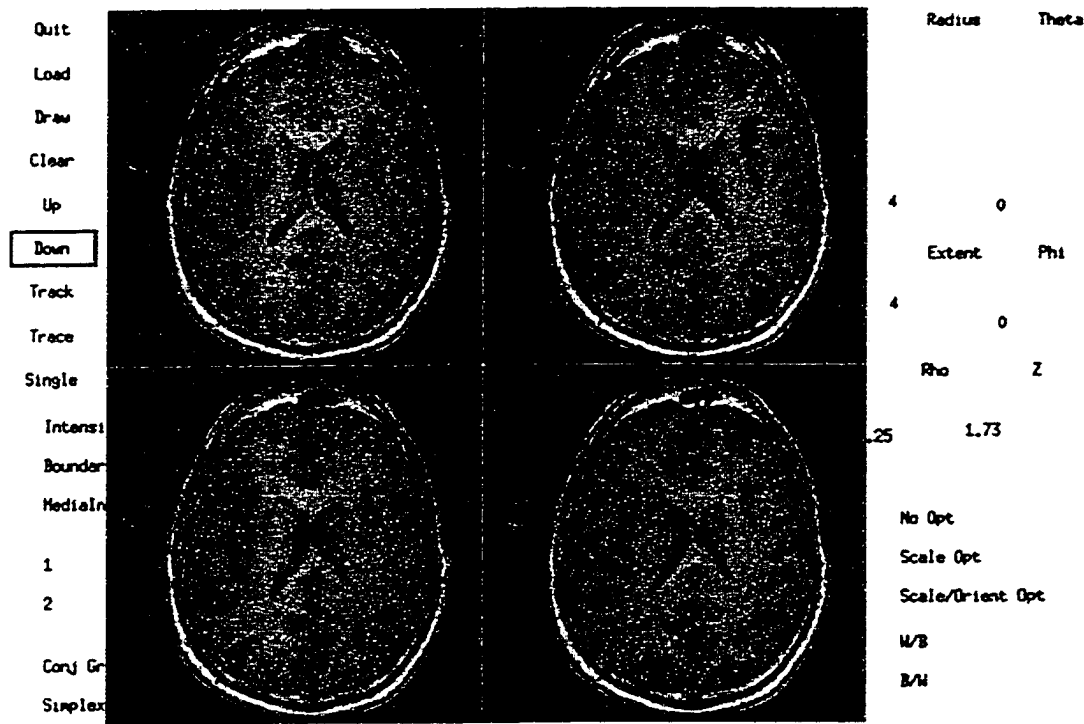


Figure 6.72: Image of ventricles (metaslice 3 - image slices 9-12)

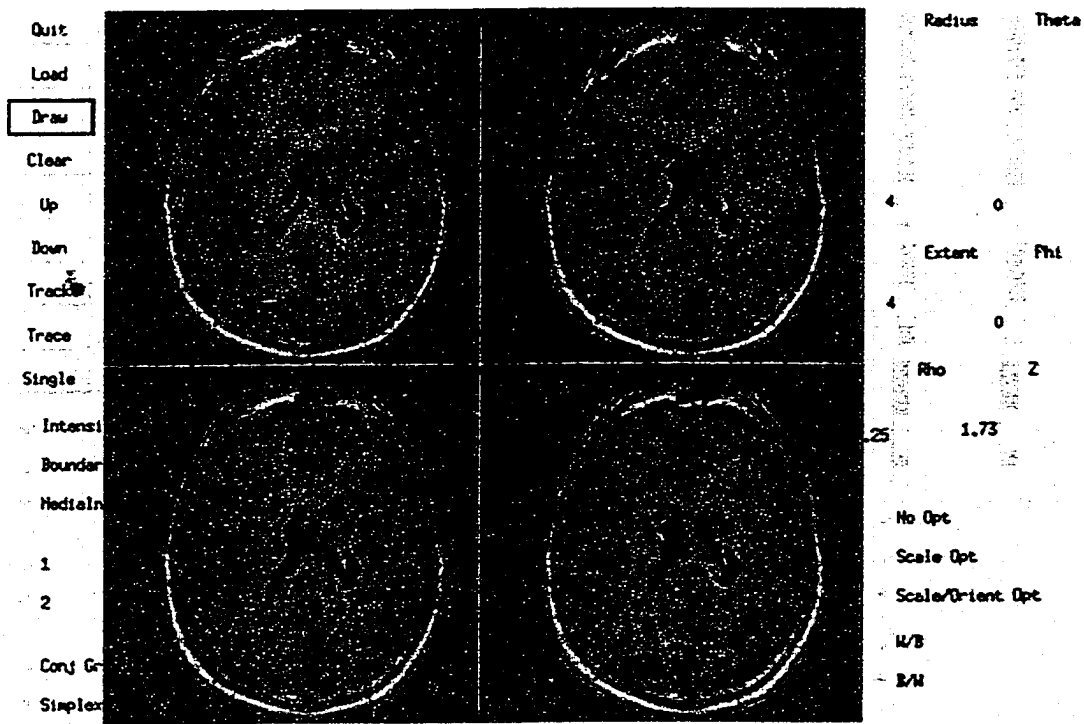


Figure 6.73: Image of ventricles (metaslice 3 - image slices 9-12) with object implied by the core superimposed in black

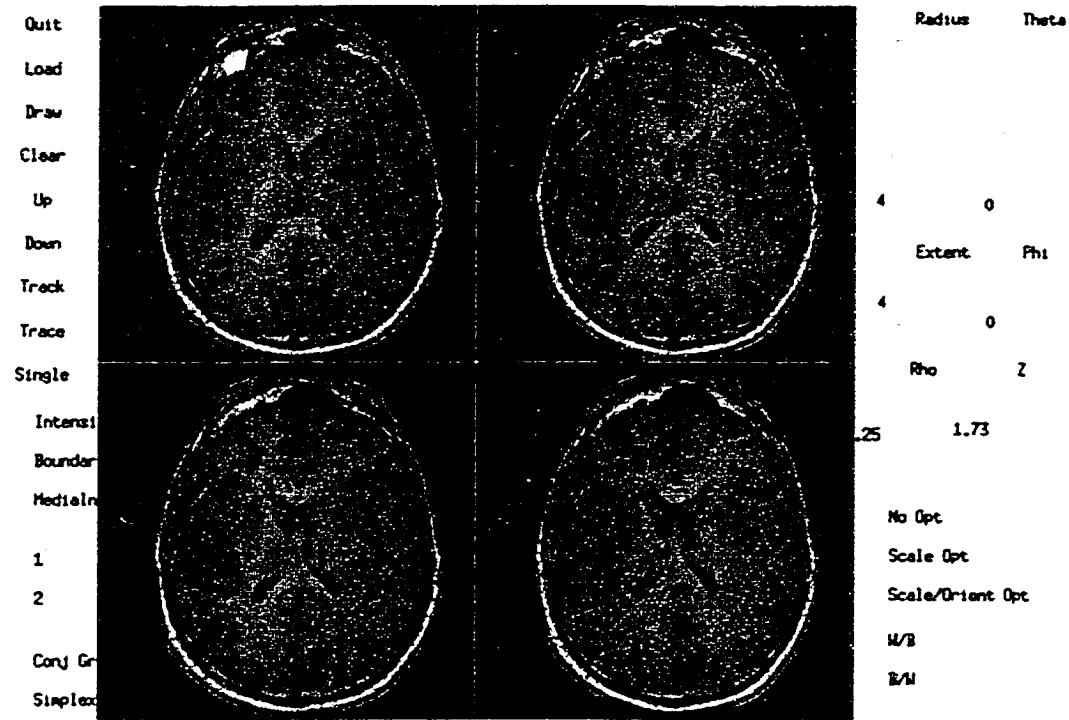


Figure 6.74: Image of ventricles (metaslice 4 - image slices 13-16)

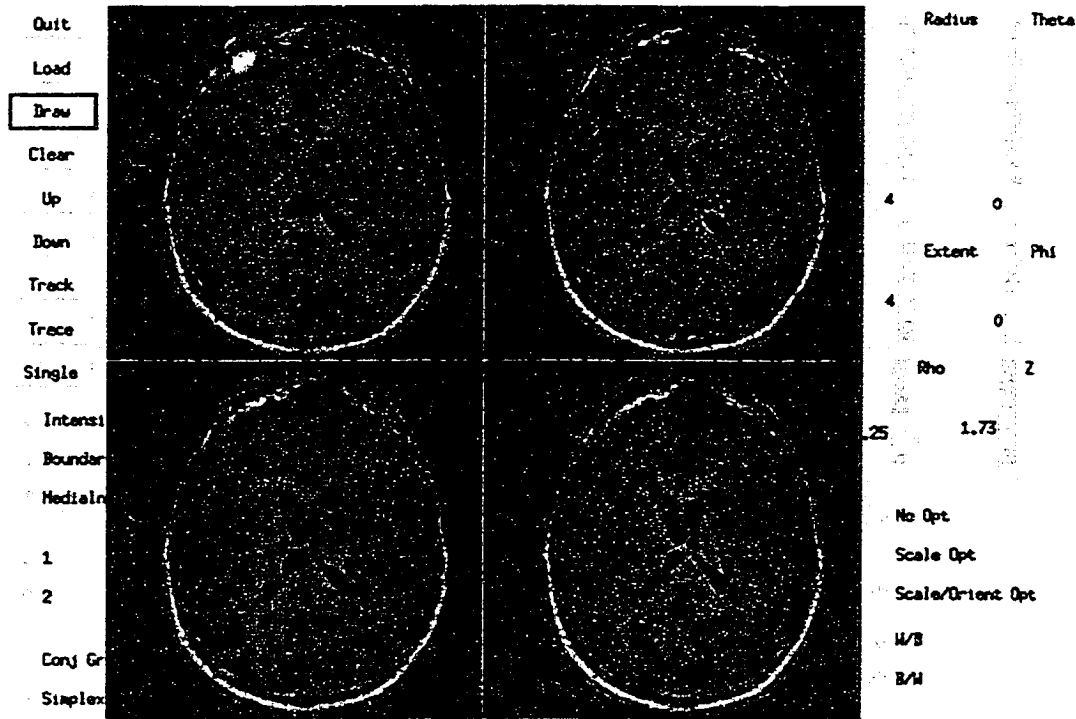


Figure 6.75: Image of ventricles (metaslice 4 - image slices 13-16) with object implied by the core superimposed in black

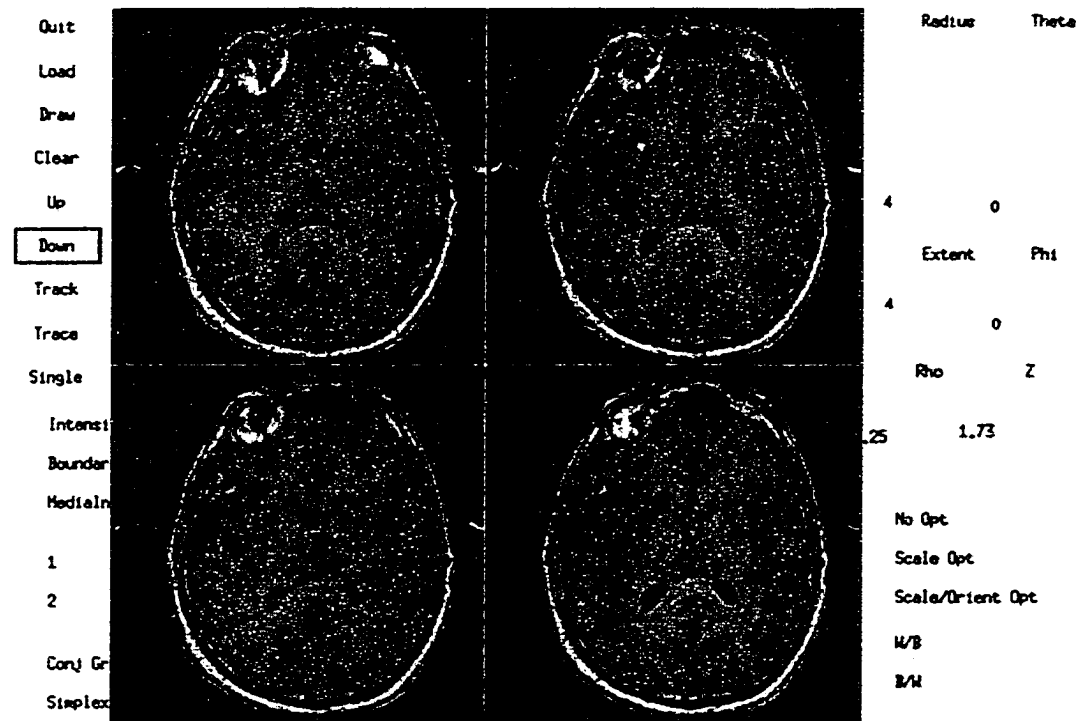


Figure 6.76: Image of ventricles (metaslice 5 - image slices 17-20)

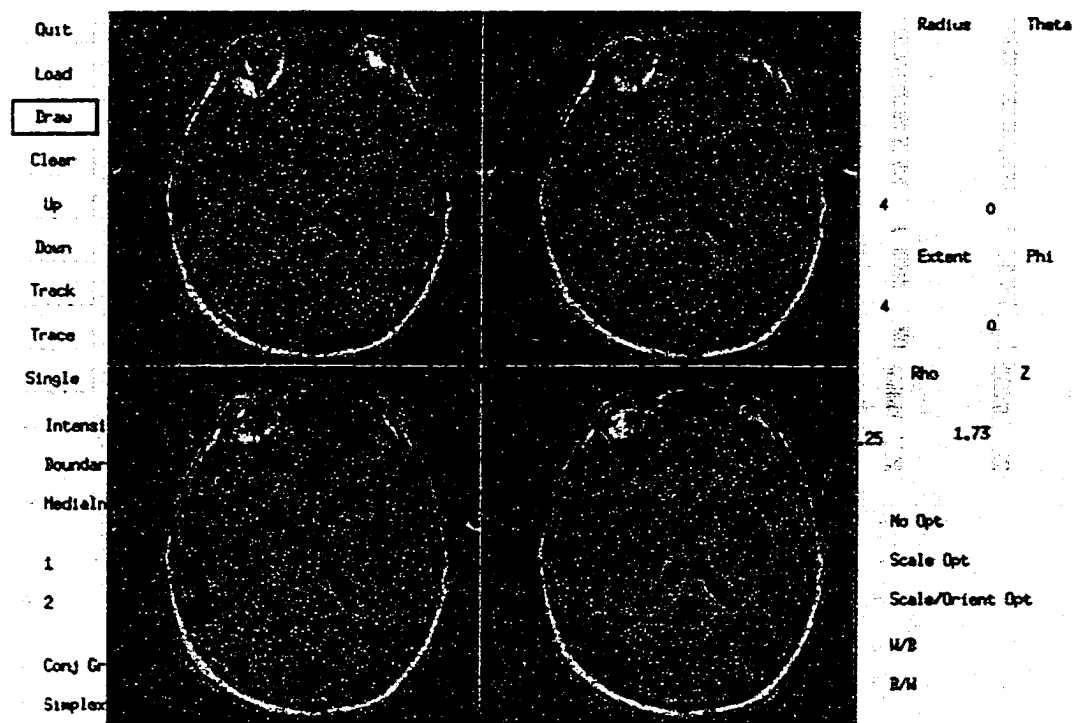


Figure 6.77: Image of ventricles (metaslice 5 - image slices 17-20) with object implied by the core superimposed in black

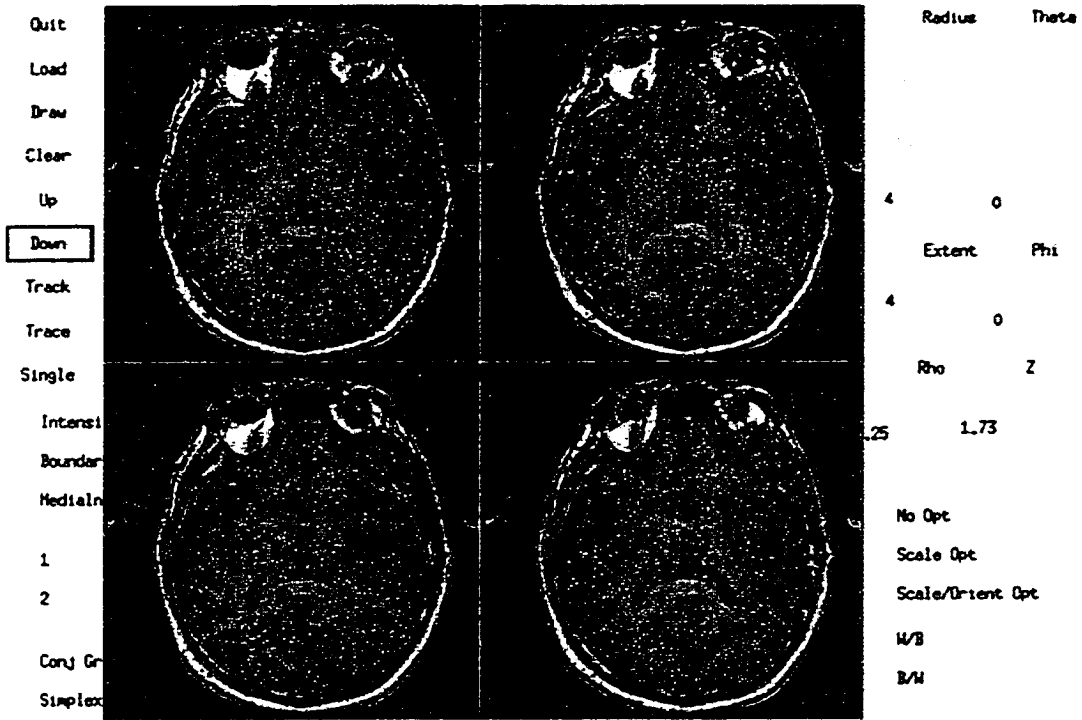


Figure 6.78: Image of ventricles (metaslice 6 - image slices 21-24)

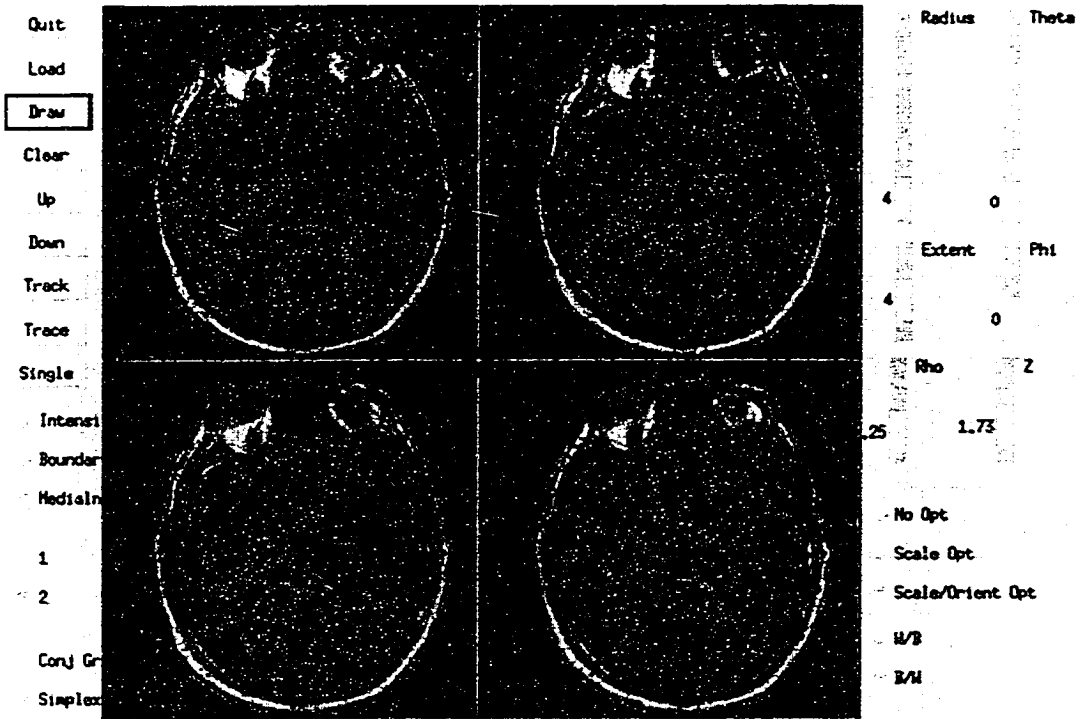


Figure 6.79: Image of ventricles (metaslice 6 - image slices 21-24) with object implied by the core superimposed in black

Overall, the core extraction performed as expected. The results on the test objects indicate that the core extraction is accurately calculating positions, widths and orientations of ridges of medialness that are known to be present. The marching extends over the entire core, not stopping prematurely. It could be much faster, and future work will include faster optimizers, more efficient medialness calculations, and porting to the faster machines now available.

The results on the medical images, while not as “clean” as the results of the test objects, were predictable in their behavior. Three behaviors of cores predicted by the mathematics of medialness and ridges were observed:

- the marching algorithm passes through the ends of objects,
- ridges do not branch and may stop at places where an object branches, and
- low contrast boundaries between objects are ignored.

Each of these behaviors can be seen in the results of the medical images. (In fact, the first is seen even in the core of the flared tube.) Figure 6.73 shows what happens at the ends of objects: the core marches past the end of the object and begins wandering in the rest of the image. While the ventricle is well modeled where its sides are parallel, at the lower tip of the right ventricle, the core has gone through, and interacting with noise in the image, started wandering through the image. Figure 6.15 shows what happens at a branch: the core of a blood vessel is coming up from lower in the brain in the lower two slices, and then stops, not appearing in the upper two slices. The vessel begins to branch in the lower left slice and does not continue into the branch of the upper slices. Figure 6.37 shows what happens when two objects do not have a high contrast boundary: the core marches into the second object. The kidney on the left is quite close to the liver, and the boundary between them is very indistinct. The core, initiated in the kidney, has begun to register in the liver.

In addition to the properties of cores predictable from theory, other behaviors derive from the Cartesian grid of sample points and Marching Ridges is susceptible to the following pixel-based constraints:

- a lower limit on the size of objects and
- an upper limit on the curvature of the core.

Both of these constraints are evident in the medical images. Figures 6.55 and 6.57 show how the core does not extend into very thin sections of the skull, although slightly thicker parts are well identified. Figure 6.49 show the effect of curvature: the core does not extend up into the top of the skull. Here the curvature is with respect to  $z$  distance, because the interslice distance is 2.76 times the intraslice distance between sample points. Thus, the skull curves extremely sharply with respect to  $z$ , and the marching fails to follow the core.

Future work will investigate how best to deal with these behaviors and overcome these constraints. Blum-like medialness may provide a mechanism for naturally recognizing the ends of objects. Connectors may provide a way of dealing with branches, by following connectors through the branch until ridges again emerge away from the branch.



Supersampling may provide a way of dealing with the pixelization artifacts. Adaptive medialness that changes its behavior as radius changes may also provide a method of dealing with this problem.

In general, the cores look promising as starting points for the medial analysis of object shape in medical images. Besides the needed improvements just described, there is other work to be done. For example, the visualization provided in these images is very basic. The orientation of the core is not shown in the images, and the representation of the core is limited to the original slices of the image. A system of viewing cores that allowed arbitrary slices or 3D visualizations would be more useful and easier to intuitively understand, as would one that allowed a visualization of core radius and orientation.

Further, this dissertation concentrates only on medical images. Other imaging modalities, such as seismic, visible light, and infrared may also benefit from medial shape analysis and thus prove a fertile ground for the application of this research.

### 6.4.3 Practical Considerations

There are three main practical issues concerning the work in this dissertation:

- speed,
- extensions, and
- incorporation into other systems.

The issue of speed has been raised a number of times already in the dissertation. The choice of medialness can have a huge impact on the speed of the algorithm, depending on the footprint of the weighting function. The Morse and Blum-like medialness measurements are fastest for finding 2D cores, while the Laplacian is fastest for 1D cores. The choice of which dimension reduction strategy to use with optimal parameter ridges was determined in part by speed. There are other options for speeding up the algorithm. The first is a software solution, involving resampling the image and medialness weighting function based on the scale parameter of the medialness measurement. This has two potential benefits: 1) the footprint of the weighting function is constant, regardless of the radius of the core and 2) the number of different medialness weighting functions is reduced as aperture becomes constant in terms of pixel widths. The former will naturally drastically speed up the calculation of larger radius cores while the latter would allow the precalculation of a small number of weighting functions, removing those calculations from deeply nested loops. The second option is a hardware option: the design of circuits for the specific application of medialness weighting functions. Digital signal processing chips are fairly common in specific 1D applications, and it is not unreasonable to expect that the same will happen for 2D and 3D applications.

The work in this dissertation has been designed to be easily extended. The components of the algorithm have been separated in source code as they are in this dissertation: the calculation of medialness is contained in separate source files from the calculation of optimal parameter ridges, which is kept in separate source files from the marching algorithms. This has allowed me to relatively easily add medialness functions as they are

defined, change the method of finding ridges, and extend the Marching Ridges algorithm from 2D to 3D. Clary [8] is in the process of extending the code to include endness calculations in the ridge calculation. He has reported relatively little trouble in 2D. Other extensions could include the calculation of connector sets, Blum-like medialness, and 4D data sets.

This separation of source that makes this work easily extended also benefits its inclusion in other systems. A number of researchers have taken the medialness functions included here and applied them to their own algorithms successfully. Aylward and I were able to modify a medialness function and apply it to his ridge finding system in his work on finding ridges of blood vessels. Both Fritsch [17] and Low use the oriented Laplacian medialness measurement in their work on deformable shape models in 3D.

## 6.5 Conclusion

The contributions of this dissertation provide a better understanding of the medial shape analysis of objects in 3D medical images. They support the thesis that oriented medialness and optimal parameter height ridges employed in a marching ridges algorithm will effectively extract cores of 2D and 3D greyscale images. This thesis, in turn, provides a platform from which to launch even deeper studies into the medial analysis of shape.

# Bibliography

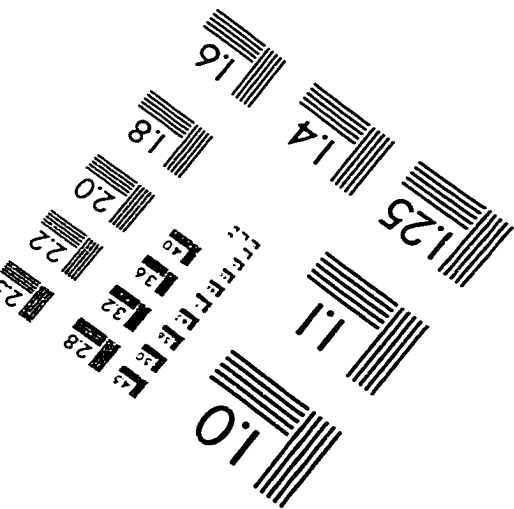
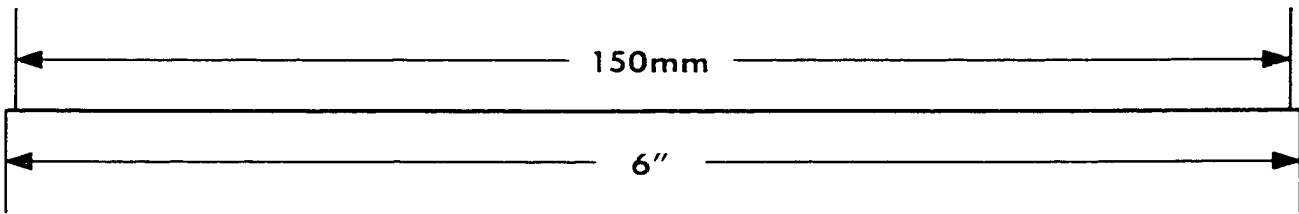
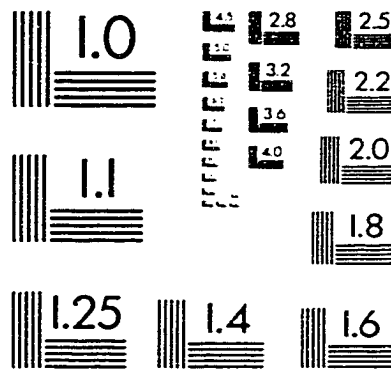
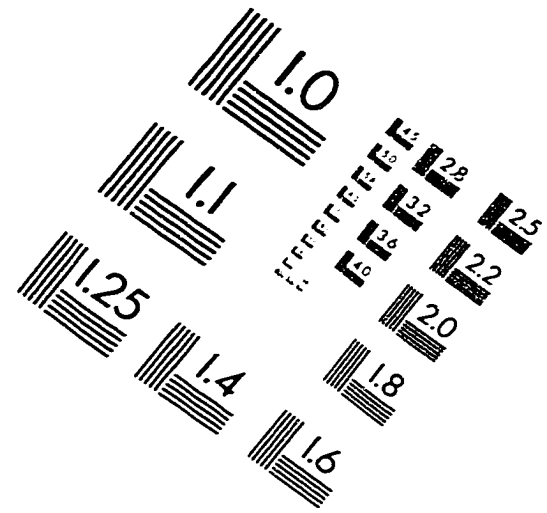
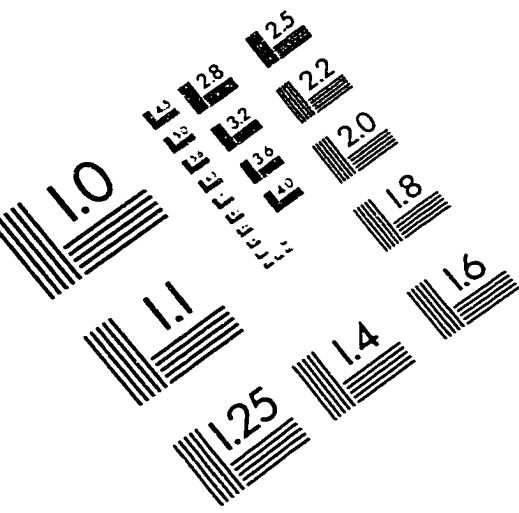
- [1] S. Aylward, S. Pizer, E. Bullitt, and D. Eberly. Intensity ridge and widths for tubular object segmentation and description. In *Proceedings of the workshop on Mathematical Methods in Biomedical Image Analysis*, pages 131–138. IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence, June 1996.
- [2] J. Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, pages 341–355, 1988.
- [3] H. Blum and R. N. Nagel. Shape descriptions using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.
- [4] Fred L. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, 1991.
- [5] E. Bullitt, A. Liu, and S. Pizer. Methods of displaying intracerebral vascular anatomy. *American Journal of Neuroradiology*, 18:417–420, 1997.
- [6] C. A. Burbeck and S. M. Pizer. Object representation by cores: Identifying and representing primitive spatial regions. *Vision Research*, 35:13:1917–1930, 1995.
- [7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.
- [8] G. Clary, 1999. Personal communication.
- [9] Gregory J. Clary, Stephen M. Pizer, Daniel S. Fritsch, and J. Randolph Perry. Left ventricular wall motion tracking via deformable shape loci. In H. U. Lemke, M. W. Vannier, and K. Inamura, editors, *Computer Assisted Radiology and Surgery*, pages 271–276, 1997.
- [10] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for location structures in medical images. In H. H. Barrett and A. F. Gmitro, editors, *Information Processing in Medical Imaging*, volume 687 of *Lecture Notes in Computer Science*, pages 33–47. Springer-Verlag, 1993.
- [11] J. L. Crowley and A. C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):156–170, March 1984.

- [12] J. N. Damon. Properties of ridges and cores for two dimensional images. Preliminary version.
- [13] D. Eberly. *Geometric Methods for Analysis of Ridges in N-Dimensional Images*. PhD thesis, The University of North Carolina at Chapel Hill, January 1994.
- [14] D. Eberly. *Ridges in Image and Data Analysis*. Computational Imaging and Vision Series. Kluwer Academic Publishers, Dordrecht, NL, 1996.
- [15] M. Fidrich. Iso-surface extraction in 4d with applications related to scale space. Technical Report 2833, Institut National de Recherche en Informatique et en Automatique, March 1996.
- [16] L. Florack, B. ter Haar Romeny, J. Koenderink, and M. Viergever. On scale and differential structure of images. In *Image & Vision Computing*, volume 10, pages 376–388. Kluwer Academic Publishers, 1992.
- [17] D. Fritsch. 1999. Personal communication.
- [18] D. S. Fritsch. *Registration of Radiotherapy Images using Multiscale Medial Descriptions of Image Structure*. PhD thesis, The University of North Carolina at Chapel Hill, 1993.
- [19] D. S. Fritsch, E. L. Chaney, A. Boxwala, M. J. McAuliffe, S. Raghavan, A. Thall, and J. R. D. Earnhart. Core-based portal image registration for automatic radiotherapy treatment verification. *International Journal of Radiation, Oncology, Biology, Physics. (special issue on Conformal Therapy)*, 33(5):1287–1300, 1995.
- [20] D. S. Fritsch, D. Eberly, S. M. Pizer, and M. J. McAuliffe. Simulated cores and their application in medical imaging. In *Proceedings of the International Conference on Image Processing in Medical Imaging*, pages 365–368, 1995.
- [21] D. S. Fritsch, S. M. Pizer, B. S. Morse, D. H. Eberly, and A. Liu. The multiscale medial axis and its applications in image registration. *Pattern Recognition Letters*, 15:445–452, May 1994.
- [22] J. Furst, S. Pizer, and D. Eberly. Marching cores: a method for extracting cores from 3d medical images. In *Proceedings of the workshop on Mathematical Methods in Biomedical Image Analysis*, pages 124–130. IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence, June 1996.
- [23] Jacob D. Furst, Robert S. Keller, Jason E. Miller, and Stephen M. Pizer. Image loci are ridges in geometric spaces. In Bart ter Haar Romeny, Luc Florack, Jan Koenderink, and Max Viergever, editors, *Scale Space Theory in Computer Vision*, Lecture Notes in Computer Science, pages 176–187. Springer-Verlag, 1997.
- [24] R. M. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22:28–38, 1983.
- [25] S. N. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. On topological deep-structure segmentation. In *Proc. Intern. Conf. on Image Processing*, pages 863–866. Santa Barbara, California, October 26–29 1997.

- [26] R. Katz. *Perception-Based Shape Simplification*. PhD thesis, University of North Carolina at Chapel Hill, In preparation.
- [27] R. Keller. *Generic Transitions of Relative Critical Sets in Families of Smooth Functions with Applications to Medical Imaging*. PhD thesis, University of North Carolina at Chapel Hill, In preparation.
- [28] J. J. Koenderink. *Solid Shape*. The MIT Press, second printing edition, 1990.
- [29] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. Technical Report ISRN KTH/NA/P-96/06-SE, KTH (Royal Institute of Technology), 1996.
- [30] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.
- [31] K. C. Low. A figural deformation approach to segmentation, 1997. Internal Report: UNC-CH Department of Computer Science.
- [32] J. B. A. Maintz, P. A. van den Elsen, and M. A. Viergever. Evaluation of ridge seeking operators for multimodality medical image matching. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 18(4):353–364, April 1996.
- [33] M. McAuliffe, D. Eberly, D. Fritsch, E. Chaney, and S. Pizer. Scale-space boundary evolution initialized by cores, 1995.
- [34] J. Miller. *Relative Critical Sets in  $\mathbb{R}^n$  and Applications to Image Analysis*. PhD thesis, University of North Carolina at Chapel Hill, 1998.
- [35] B. S. Morse, S. M. Pizer, and A. Liu. Multiscale medial analysis of medical images. *Proceedings on Information Processing in Medical Images*, 687:112–131, 1993.
- [36] L. R. Nackman. Curvature relations in three-dimensional symmetric axes. *Computer Graphics and Image Processing*, 20:43–57, 1982.
- [37] S. M. Pizer and C. A. Burbeck. Robust object shape representation, 1996. Poster presentation at ARVO96.
- [38] S. M. Pizer, D. Eberly, B. S. Morse, and D. S. Fritsch. Zoom-invariant vision of figural shape: The mathematics of cores. *Computer Vision and Image Understanding*, 69:55–71, 1998.
- [39] S. M. Pizer, D. H. Eberly, B. S. Morse, and D. S. Fritsch. Zoom invariant vision of figural shape: the mathematics of cores. Technical Report TR96-004, University of North Carolina, 1996. Accepted for publication in *Computer Vision and Image Understanding*.
- [40] D. Puff. *Human vs. Model Performance for Medical Image Estimation Tasks*. PhD thesis, The University of North Carolina at Chapel Hill, 1995.
- [41] X. Qu and X. Li. A 3d surface tracking algorithm. *Computer Vision and Image Understanding*, 64(1):147–156, July 1996.
- [42] M. E. Shenton, C. G. Wible, and R. W. McCarley. A review of magnetic resonance imaging studies of brain abnormalities in schizophrenia. In K. Ranga Rama Krishnan

- and P. Murali Doraiswamy, editors. *Brain Imaging in Clinical Psychiatry*, pages 297-380. Mercel Dekker, Inc, New York, 1997.
- [43] G. Stetten and S. Pizer. Extracting shape properties via populations of medial primitives. Technical Report TR98-008. University of North Carolina, 1998.
  - [44] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig. Segmentation of 3d objects from mri volume data using constrained elastic deformations of flexible fourier surface models. In *First International Conference, CVRMed '95. Proceedings*, pages 495-505. April 2-6 1995.
  - [45] J. Thirion and A. Gourdon. The 3d marching lines algorithm and its application to crest lines extraction. Technical Report 1672, Institut National de Recherche en Informatique et en Automatique, May 1992.
  - [46] Terry S. Yoo. *Image Geometry Through Multiscale Statistics*. PhD thesis, University of North Carolina at Chapel Hill, 1996.

# IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc.  
1653 East Main Street  
Rochester, NY 14609 USA  
Phone: 716/482-0300  
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

