5-2001

Development of a unified probabilistic framework for segmentation and recognition of semi-rigid objects in complex backgrounds via deformable shape models

Shaun Scott Gleason

To the Graduate Council:

I am submitting herewith a dissertation written by Shaun Scott Gleason entitled "Development of a unified probabilistic framework for segmentation and recognition of semi-rigid objects in complex backgrounds via deformable shape models." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Mongi A. Abidi, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Shaun S. Gleason entitled "Development of a Unified Probabilistic Framework for Segmentation and Recognition of Semi-Rigid Objects in Complex Backgrounds via Deformable Shape Models." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

M. A. Abidi

Dr. Mongi A. Abidi, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Interim Vice Provost and
Dean of the Graduate School

Development of a Unified Probabilistic Framework for Segmentation and Recognition of Semi-Rigid Objects in Complex Backgrounds via Deformable Shape Models

A Dissertation for

the Doctor of Philosophy Degree

The University of Tennessee, Knoxville

Shaun S. Gleason

May 2001

Dedication

This dissertation is dedicated to my wife, Caroline, and to my daughters, Palmer and Paige, who have supported me and have demonstrated great patience throughout the completion of this work.

Acknowledgements

I would like to first acknowledge the members of my committee, Dr. Abidi, Dr. Sari-Sarraf, Dr. Gregor, Dr. Karakashian, and Dr. Qi, for their advice during my research and the preparation of this document. In particular, I would like to thank Dr. Abidi for his guidance, encouragement, and generosity in providing me tools and facilities to complete this work. I also extend special thanks to Dr. Sari-Sarraf for his support and invaluable technical guidance from the very beginning of this research.

I acknowledge and thank Dr. Michael Paulus for his technical partnership and his vision that led to the development of the X-ray micro-CT scanner that inspired this research. I also thank John Branning, who was responsible for gathering much of the data presented in this dissertation.

Finally, thanks to the staff at Oak Ridge National Laboratory, particularly Ken Tobin, Jim Jansen, Dan McDonald, and Janet Swift, who supported me by providing a 2-year part-time sabbatical that was essential for the completion of this research.

Abstract

This dissertation presents the development, implementation, and application of a unified probabilistic shape and appearance model (PSAM) algorithm for boundary-based segmentation and recognition of semirigid objects on complex backgrounds. The algorithm requires that the boundary be represented by a set of landmark points (LPs). These LPs are iteratively adjusted to fit the boundary of a new object based on a priori information gathered from a training set. PSAM is derived from compound Bayesian decision theory, and the formulation is general enough that it can be used as a starting point to derive a variety of other probabilistic boundary-finding techniques. The motivation for developing PSAM arose from a need to segment and recognize semirigid anatomic structures within medical images that have faint and/or missing edge information.

The starting point for this research was the active shape model (ASM). ASM is described, along with some practical improvements that were made to the published algorithm. These practical improvements are demonstrated on synthetic and real data. ASM was tested on a a set of 2D medical images of kidneys within X-ray CT images of laboratory mice. Although ASM performance was improved because of the practical improvements, some remaining fundamental problems led to poor segmentation accuracy in many cases. These fundamental problems inspired the development of the PSAM algorithm.

PSAM contains three specific model components: (1) a global shape model (GSM), (2) a local shape model (LSM), and (3) a gray-level model (GLM). The GLM formulation is based on gradient gray-level profiles normal to the object boundary through each LP. All three of the PSAM components are optimized simultaneously when boundary searches

are performed within new images. PSAM is formulated so that the influence of each of these components on the final boundary position can be controlled by the system operator. This allows the same PSAM algorithm to be used in applications with predictable global shape and relatively poor object edge strength, as well as in other applications where global shape is unpredictable but object edges are prominent.

The new PSAM algorithm formulation provides confidence metrics for each of the three model components that give the operator feedback on the segmentation result. These confidence metrics indicate how well each PSAM component (GSM, LSM, and GLM) of the final boundary fits within the distribution of each component as derived from the training data. These confidence metrics can be monitored to alert an operator to any boundary results where one or more model components were found to be "out of bounds" relative to the training data. Furthermore, it was demonstrated that for some applications, the GLM confidence metric can be used as a predictor of segmentation accuracy.

The performance of the PSAM algorithm is summarized on both synthetic and real-world data. The results of three cases of real medical image data segmentations are presented. These cases include X-ray tomographic images of anatomic structures within laboratory mice. Specifically, the skull, the heart and lungs, and the kidneys are segmented using PSAM and ASM; and the results of the two algorithms are directly compared. In all cases the PSAM algorithm performed well and in fact, outperformed ASM by a substantial margin. It is shown that PSAM has a much larger degree of success than ASM on the most difficult segmentation cases. The PSAM performance is summarized, and a variety of future research topics are suggested that could lead to improved performance and broader applicability.

Table of Contents

1. Introduction 1
   1.1 The Motivation for This Research 1
   1.2 Segmentation Methods 3
      1.2.1 Top-down versus Bottom-up Methods 3
      1.2.2 Deformable Models 4
      1.2.3 Active Shape Models 5
   1.3 Active Shape Models: Strengths and Shortcomings 6
   1.4 Research: A New Probabilistic Deformable Model 8
      1.4.1 Practical Improvements to Existing ASM 9
   1.5 Document Organization 12
2. Background: Segmentation Methods 14
   2.1 The Significance and Difficulty of Segmentation 14
      2.1.1 Medical Image Analysis 16
      2.1.2 Segmentation and Object Recognition 16
   2.2 Bottom-up versus Top-down Methods for Scene Analysis 17
   2.3 Deformable Models for Image Analysis 18
      2.3.1 Low-, Medium-, and High-Level Models 20
      2.3.2 Research in Deformable Models 21
      2.3.3 Boundary versus Region-based Deformable Models 30
3. The Active Shape Model and Practical Improvements 32
   3.1 ASM Theory 32
      3.1.1 Training via Landmark Points (LPs) 33
      3.1.2 Training Set Alignment via Procrustes Analysis 36
      3.1.3 Creating ASM via Principal Component Analysis (PCA) 37
      3.1.4 Training the GLM 40
      3.1.5 Finding a New Set of LPs 41
      3.1.6 Fitting a Model Instance to a New Set of Points 43
      3.1.7 ASM Iterative, Multiresolution Search 45
   3.2 Strengths and Shortcomings of ASM 47
   3.3 An Improved ASM 50
      3.3.1 ASM Implementation 50
      3.3.2 Resolution-Dependent Shape-Model 50
      3.3.3 Improved GLM Objective Function 53
   3.4 Segmentation Results Using Enhanced ASM 57
   3.5 The Need for a New Statistical Shape Model 62
      3.5.1 Objective 1: Combining the SM and GLM Optimization Processes 62
      3.5.2 Objective 2: Employing a Continuous Penalty for Shape Variation 66
      3.5.3 Objective 3: Including Local Shape Characteristics in Optimization 67
      3.5.4 Objective 4: Developing a Segmentation Confidence Metric 68
4. Development of a Probabilistic Framework for Model-Based Segmentation 69
   4.1 A New Probabilistic Shape and Appearance Model 70
      4.1.1 Compound Bayesian Approach 71

4.1.2 Employing the Compound Bayesian Approach for PSAM 77
4.2 Objective Function Formulation 96
4.2.1 Global Shape Objective Function 97
4.2.2 Local Shape Objective Function 98
4.2.3 Gray-Level Profile Objective Function 99
4.2.4 Objective Function Formulation Summary 99
4.3 Objective Function Optimization 99
4.3.1 GSM Gradient Formulation 100
4.3.2 LSM Gradient Formulation 101
4.3.3 GLM Gradient Formulation 107
4.3.4 Optimization of the Objective Function 112
4.3.5 Multi-Resolution PSAM Search 113
4.3.6 Weight Parameters for PSAM Components 113
4.4 Confidence Metric Formulation 116
4.5 PSAM Development Summary 119
5. Experimental Results 121
5.1 Weight Parameter Experiment on Synthetic Data 121
5.2 PSAM Results on Synthetic Medical Data 127
5.3 PSAM Results on Medical Image Data 130
5.3.1 Mouse Skull and Trachea Segmentation 131
5.3.2 Mouse Heart and Lung Segmentation 137
5.3.3 Mouse Kidney and Spine Segmentation 144
5.3.4 Gaussian Distribution Goodness-of-Fit Tests 152
5.4 LSM Experiments 154
5.5 Experimental Results Summary 158
6. Conclusions and Future Work 160
6.1 PSAM Development and Performance 160
6.2 Future Research 163
Bibliography 166
Vita 175

List of Figures

Fig. 2.1. Illustration of top-down vs bottom-up scene analysis techniques. 19
Fig. 3.1. Simplified ASM flow diagram. 34
Fig. 3.2. Example of a mouse kidney training image. 35
Fig. 3.3. An example showing a set of kidney and spine LPs: 38
Fig. 3.4. Images with overlaid normal profiles. 42
Fig. 3.5. Plot showing the mean shape vector. 46
Fig. 3.6. Result of using ASM to segment shapes. 48
Fig. 3.7. Example of multiresolution ASM search. 52
Fig. 3.8. ASM segmentation result showing inclusion of contrast agent. 56
Fig. 3.9. Example of ASM segmentation. 58
Fig. 3.10. ASM segmentation using the derivative and intensity GLM. 59
Fig. 3.11. ASM results on three mouse kidney CT images. 60
Fig. 3.12. ASM results on three mouse kidney CT images. 61
Fig. 3.13. Plot of the GLM objective function. 64
Fig. 3.14. Results of GLM fit before (a) and after (b) correction of LP 17. 65
Fig. 3.15. Poor ASM result due to ambiguity in GLM fit at individual LPs 66
Fig. 3.16. ASM segmentation of subject 13 after adjustment of LP 17 67
Fig. 4.1. Illustration of a shape boundary. 89
Fig. 5.1. Example of synthetic image used for weight parameter testing. 122
Fig. 5.2. Segmentation results on star image. 123
Fig. 5.3. New segmentation results on star image. 124
Fig. 5.4. Segmentation results on star image. 125
Fig. 5.5. Segmentation results on modified star. 126
Fig. 5.6. New segmentation results on modified star. 126
Fig. 5.7. Simulated kidney image with geometric distortion and noise. 128
Fig. 5.8. PSAM results with geometric distortion and additive noise. 129
Fig. 5.9. Sample mouse skull images from first training set. 132
Fig. 5.10. Transaxial slice through an X-ray CT volume of a mouse skull. 133
Fig. 5.11. Average boundary error for the skull images. 134
Fig. 5.12. Skull sample #10 segmentation: 135
Fig. 5.13. PSAM segmentation result for image 7. 136
Fig. 5.14. Plot of average boundary error vs GLM confidence for skull images. 137
Fig. 5.15. Example heart and lung images from second training set. 139
Fig. 5.16. Transaxial slice through an x-ray CT volume of a mouse chest. 140
Fig. 5.17. Average boundary error for the heart and lung images. 141
Fig. 5.18. Heart and lung segmentation result for image 3: 142
Fig. 5.19. Segmentation result using (a) PSAM and (b) ASM for image 8. 143
Fig. 5.20. Average error vs GLM confidence for heart and lung images. 144
Fig. 5.21. Mouse kidney images from third training set. 145
Fig. 5.22. Transaxial slice through an X-ray CT volume of a mouse abdomen. .147
Fig. 5.23. Total boundary error for the kidney and spine images. 148
Fig. 5.24. Kidney and spine segmentation result on image 1: 149

Fig. 5.25. Kidney and spine segmentation result on image 12: 150
Fig. 5.26. Average error vs GLM confidence for kidney and spine images. 151
Fig. 5.27. Probability distributions for multivariate Gaussian shape data. 153
Fig. 5.28. Effect of LSM influence on segmentation on mouse skull. 155
Fig. 5.29. Three examples from the ivy leaf data set. 156
Fig. 5.30. Ivy leaf example. 157
Fig. 5.31. Samples of PSAM segmentation on ivy leaf images. 157

List of Abbreviations and Symbols

Abbreviations:

A-D	Anderson-Darling
AAM	active appearance model
ASM	active shape model
CT	computed tomography
GLM	gray-level model
GSM	global shape model
K-S	Kolmogorv-Smirnov
LP	landmark point
LSM	local shape model
MAP	maximum a posteriori probability
MRI	magnetic resonance imaging
PCA	principal component analysis
PSAM	probabilistic shape and appearnace model
SM	shape model

Variables:

italicized letters (e.g. n, N, m, M	scalar variables
lower-case, italicized, bold letters (e.g. $\boldsymbol{v}$, $\boldsymbol{l}$, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{q}$, $\boldsymbol{r}$)	vectors
subscripted vector variable (e.g. v_i, l_j)	vector elements
upper-case, italicized, bold letters (e.g. $\boldsymbol{C}$, $\boldsymbol{S}$, $\boldsymbol{G}$	matrices
lower-case, subscripted matrix variable ($\boldsymbol{c}_i$, $\boldsymbol{s}_j$, $\boldsymbol{g}_k$)	matrix column vector

lower case, subscripted matrix variable (e.g. $c_{i,j}$, $s_{j,k}$)	matrix elements
i, j, and k are counting indices	
M is the number of training samples	
N is the number of LPs	
N_c is the number of critical LPs	
N_g is the number of gray-level samples	
t is the number of modes of variation	

Diacritical marks:

"bar" (e.g. $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{g}}$)	mean of set of vectors
"hat" (e.g. $\hat{\boldsymbol{v}}$, $\hat{\boldsymbol{g}}$)	vector aligned to set mean

Operators:

T-superscript (e.g. $\boldsymbol{v}^{\mathrm{T}}$, $\boldsymbol{G}^{\mathrm{T}}$)	transpose of vector or matrix

CHAPTER 1

INTRODUCTION

This document is a Ph.D. dissertation in the field of probabilistic, statistical-based deformable shape models for object segmentation and recognition in digital images. This introductory chapter first presents the motivation for the research. It then introduces the general field of deformable models for image analysis. It goes on to justify the choice of one deformable model-based analysis technique—the active shape model (ASM) [22]—as a starting point for the proposed research. It then summarizes the completed objectives of this research, which include the development of a new unified probabilistic framework for segmentation and recognition of semirigid objects in complex backgrounds via deformable shape models. Finally, the organization of the remainder of this document is outlined at the end of the chapter so that the reader can explore in more detail any aspect of the completed research.

1.1 The Motivation for This Research

The motivation for this work is the need for an algorithm that performs automatic segmentation and recognition of semirigid objects within complex backgrounds. A semirigid object is one that exhibits controlled shape variability across multiple instances of that object. In a statistical sense, controlled variability implies that the intraclass variation of object shape is less than the interclass shape variation. The algorithm must handle cases in which the object's boundary may be faint, obscured, or partially missing due to the com-

plexity of the background on which the object lies. A complex background can either obscure edges of the object and/or can introduce other confusing edge information that does not belong to the object of interest. The algorithm should also take full advantage of available a priori information regarding the appearance (e.g., shape and intensity characteristics) of the object. Finally, to ensure the practical utility of the algorithm in an automated application, it should generate some measure of confidence associated with the segmentation result. In summary, we seek a segmentation/recognition algorithm capable of handling

- semi-rigid objects
- complex backgrounds
- faint, obscured, or partially missing object boundaries
- open and/or closed boundaries
- available a priori information on object appearance
- a need for segmentation confidence metric

To parallel this description with an application example, consider the field of medical imaging, where anatomical structures may be somewhat predictable in terms of appearance (e.g., organ shape and location), but where variations are always encountered from subject to subject and within the same subject over time. Also, there are areas of the anat-

omy (e.g. the abdominal cavity) where the appearance, location, and background of structures are quite unpredictable, making the segmentation task even more difficult. The difficulties created by these variations in the organ and its background are, at least in part, why semi-rigid organ segmentation is not as well studied as rigid-organ (or rigidly enclosed organ, i.e., brain) segmentation. Also, in many cases of medical image analysis there is an abundance of a priori information available in the form of the patient's own historical records as well as imagery from a potentially large population of other patients that could be used for algorithm training. Finally, for automated medical image analysis to be used in a clinical setting, there needs to be a confidence measure generated by the algorithm to provide an opportunity for a clinician to intervene in the event of questionable cases or unexpected results.

1.2 Segmentation Methods

This section briefly describes the categorization of segmentation approaches into top-down and bottom-up categories. Deformable models are then introduced, and a statistical-based deformable model approach, ASM, is described as the starting methodology for the proposed work.

1.2.1 Top-down versus Bottom-up Methods

Segmentation algorithms can generally be categorized into three classes: bottom-up, top-down, and hybrid approaches. Bottom-up segmentation techniques are data-driven and, hence, rely on little or no a priori information about the object of interest. The data within the image drives the algorithm to a solution. On the other hand, top-down approaches are model-based and rely heavily on a priori information to preconstruct an

object model. Because the top-down model is defined to locate specific objects, it actually combines the steps of segmentation and recognition into one. The data in the image, though critical to the top-down algorithm's success, play a smaller role in determining the form of the final solution. Some techniques are a hybrid of bottom-up and top-down approaches in that they rely heavily on both the image data and an object model when searching for a segmentation solution. Depending on the specificity of the model, these hybrid techniques may also combine the segmentation and recognition steps into a single one. It is this class of hybrid techniques that is most applicable to the semirigid object segmentation problem outlined previously. While the data must play an important role in defining the actual shape of the semirigid object, the preconstructed model needs to ensure that the resulting segmentation falls within acceptable shape limits. Some techniques within the class of segmentation algorithms known as deformable models fall into this hybrid category and are described next.

1.2.2 Deformable Models

Deformable models belong to a class of algorithms that typically use a boundary representation (2D deformable contour) or a surface representation (3D deformable surface) of an object for segmentation purposes. The deformable contour is typically initialized by placing a contour on the image and then iteratively adjusting its position and shape to best fit the object's boundary. The characteristics of the contour (e.g., local roughness, overall shape) depend on a set of parameters that is adjusted and tuned via an optimization process to best fit the information about the object as represented by the image data. These parameters that control the general form of the resultant contour are typically referred to

as *internal* constraints. The image data generate forces, referred to as *external* forces, that attract or repel the model. In their foundational work on active contours, or snakes, Kass et al. [57] formulated the internal constraints and external forces as internal and external energy terms in an objective function that, when minimized, corresponds to the best fit of the model to the image data.

1.2.3 Active Shape Models

Within the class of deformable models, ASM is one of the algorithms best-suited for applications where a priori information is available for incorporation into the segmentation process. ASM uses a training set of images that have been manually segmented via the placement of a collection of landmark points (LPs) along the boundary of an object. Information is extracted from this training data to create two distinct models, a shape model (SM) and a gray-level model (GLM), that together constitute the overall model in this hybrid segmentation approach. SM is created using the spatial location and the inter-relationship of LPs across the training images, while GLM is created using local gray-scale gradient characteristics of sampled profiles within the training images. ASM is a multi resolution search technique, so GLMs and SMs are created during training (and applied during testing) at each of the predefined image resolutions.

To put the optimization of ASM in the general context of deformable contours, described earlier, SM imposes the *internal* shape constraints on the final solution, and GLM is influenced by the *external* forces created by the image data. ASM is initialized by placing an "average" contour on a test image. The shape and position of this initial contour is then iteratively updated to optimize its fit to the underlying image information. As in

other deformable contour algorithms, the optimization of ASM achieves a compromise between the internal constraints imposed by SM and the external forces influencing GLM. An iterative scheme is employed whereby GLM finds a new set of LPs based on the best fit to the image data and passes that LP set to SM. SM then constrains the shape of those LPs to fall within the range of allowable shapes as defined by the training set and passes this constrained set of LPs back to GLM. This process repeats until the constrained set of LPs generated by SM stops changing. Because the SM and GLM are quite specific to a particular type of object, their application also combines the segmentation and recognition steps into one.

1.3 Active Shape Models: Strengths and Shortcomings

As is demonstrated by the initial segmentation results presented later in Sec. 3.2, ASM is an excellent starting point to solve the type of problem described earlier as the motivating application for the proposed research. ASM does make substantial use of a priori information through the creation of the GLM and SM components. The knowledge built into these components during the training process allows ASM to segment objects with faint and/or missing boundaries with remarkable accuracy. Another important advantage that the ASM approach has over other deformable model algorithms is its intrinsic ability to simultaneously handle multiple boundaries, both open and closed. Acknowledging the strengths of the ASM technique is important; however, there are some critical shortcomings of ASM that render it practically useless in the motivating application of automated semirigid object segmentation.

Recall that one of the characteristics of the motivating application was the possibility of objects in a complex background. Such a background can in many cases generate spurious edge information that will confound most segmentation techniques, including ASM. Because the GLM component matches gradient profile information, spurious edges can generate multiple possible matches and, hence, an erroneous boundary position. If the shape of this erroneous boundary happens to fall within the acceptable global shape constraints as indicated by the training set, SM has no chance of correcting the erroneous decision. This shortfall can be attributed to at least two problems with the ASM approach: (1) a reliance on only gradient information during GLM optimization and, more importantly, (2) the completely independent optimization of GLM and SM.

Another important characteristic of the segmentation algorithm needed for the motivating application is the ability to generate a confidence measure that can be used to set flags for manual intervention. If there is no reliable means for the algorithm to flag anomalous results, then the algorithm becomes useless in an automated application. These shortcomings of ASM demonstrate the need for a new statistical-based deformable shape model and are the foundation for the research objectives and tasks outlined in the next section.

Other researchers are currently exploring techniques that improve upon the foundation provided by ASM. Two of the most notable developments have been by Wang and Staib [96] [97], and Kervrann and Heitz [61]. Wang and Staib propose a Bayesian formulation for optimizing the ASM solution for a given image that results in a maximum a posteriori probability (MAP) objective function containing (1) a prior term based on the expected global shape characteristics and (2) a likelihood term that relies on a Canny edge-detected

image. Although this formulation allows optimization of both internal constraints (expected shape) and external forces (Canny edge information), it abandons the effective GLM objective function. The introduction of the more complex Canny edge-detector fits more cleanly into the analytical formulation but is difficult to justify from a performance standpoint. Kervrann and Heitz also propose a similar Bayesian formulation where the external forces are derived from an edge-detected image. Furthermore, these researchers introduce a post processing step where the optimized global shape is fine-tuned on a local scale using a first-order Markov model to promote boundary smoothness. My research differs from these approaches in that it is a new boundary-finding algorithm that (1) incorporates modeling of local *and* global shape characteristics, (2) simultaneously leverages the strengths of SM and GLM during contour deformation, and (3) generates a confidence metric that can be used to flag anomalous cases. The theoretical framework developed here is a unified probabilistic framework within which the existing approaches mentioned previously fit as special cases.

1.4 Research: A New Probabilistic Deformable Model

This section outlines the research opportunities in the field of shape-based deformable models and then describes the completed objectives that take advantage of these opportunities. The research can be broken into two main categories, the first of which is a practical improvement in the existing ASM technology. The second is the development of a fundamentally new probabilistic boundary-finding algorithm that relies on image-derived training data similar to that used by ASM. The advantages of this new algorithm are summarized at the end of the section.

1.4.1 Practical Improvements to Existing ASM

The following objectives were completed as part of this research. Further information on these completed objectives can be found in Chap. 3 and Chap. 4 as well as in references [43] and [44].

Custom Implementation of ASM

Before one can perform any detailed research into the effectiveness and shortcomings of the ASM technique, a custom implementation is required. This implementation process itself ensures a complete, detailed understanding of the technique and provides a tool by which one can experiment with every module and parameter of the algorithm.

The ASM tool implementation was completed using MATLAB. Furthermore, it was compared to a commercially available version of ASM and was found to perform better (qualitatively and quantitatively) on sets of synthetic and real-world images. This tool became the foundation for the boundary-based segmentation technique outlined in this document.

Performance Enhancement of ASM

There were opportunities to make a significantly positive impact on the published ASM algorithm's performance by making three practical improvements. These three improvements, described below, involved both the SM and GLM components.

1. Create Resolution-Dependent Shape Model: ASM is a multiresolution search technique. Typically, the shape constraints imposed by SM are identical at every resolution, and this can lead to problems, especially during the low-resolution image search. The large pixels used in low-resolution images allow ASM to cover a larger search area, but

naturally lead to more significant variations in LP position. For this reason, an important objective was to create a shape constraint that was a function of the current search resolution.

Resolution dependence was incorporated into SM to provide more stringent shape constraints at lower image resolutions and more shape flexibility at higher resolutions. This has led to much-improved segmentation results over the standard, resolution-independent shape constraints (see Fig. 3.7).

2. Improve Numerical Stability of GLM: The published form of the GLM objective function can lead to a numerical instability during GLM optimization. This occurs when the gray-level gradient profiles for a given LP are relatively consistent across all of the images within the training set. This leads to a situation in which a training set with little or no variation in object edge characteristics causes the GLM objective function to become indeterminate. The objective was to reformulate the GLM objective function to more elegantly handle this potentially common circumstance.

Principal component analysis (PCA) was performed on the gray-level gradient profiles for each LP across the training images. The PCA is used to construct a transformation matrix that, when applied to new profiles, in effect removes those profile elements that have little variation across the training set. The new GLM objective function acts on these transformed profiles and therefore does not encounter numerical instabilities.

3. Incorporate Absolute Intensity into GLM: ASM uses image gradient profiles as a basis for optimizing the position of individual LPs. It became clear through the initial testing of ASM that some benefit could be derived by including absolute intensity informa-

tion in GLM. There are a variety of segmentation tasks in which the absolute intensity of the object can be characterized. The objective here was to incorporate this information into the GLM search process.

An additional term based on absolute intensity profiles was added to the GLM objective function. Recall that one of the motivations for this work was to handle segmentation in complex, unpredictable backgrounds. For this reason the absolute intensity profiles included here are sampled only within the interior of the object. This has also led to improved segmentation results (see Fig. 3.9 and Fig. 3.10).

Development of a New Probabilistic Deformable Shape Model

The ASM approach can be roughly broken down into two main concepts that make it a unique tool for object segmentation/recognition applications. First is the incorporation of a priori information in the form of training data that is acquired and used to develop shape and gray-level models. This concept of ASM has been demonstrated to be quite effective. The second, equally important ASM concept is the optimization process that is employed to deform these shape models to fit objects in new images. The noted shortcomings of the ASM optimization process motivated the development of a new optimization scheme and, hence, a new probabilistic deformable shape model. The following three subobjectives encompass the overall objective of developing this new deformable shape model.

1. Reformulate the Shape-Model Deformation Scheme: The published ASM approach separately optimizes the SM and the GLM components of the ASM. GLM searches the image gradient profiles and updates the LP positions with no regard as to how well they will fit the global constraints imposed by SM. Similarly, SM fits a new (model-generated)

set of LPs to the GLM-suggested LPs with no regard as to how this new set matches the gradient profile information contained in the GLM. Clearly, this is not the best approach for optimizing the ASM boundary. The probabilistic reformulation improved upon the ASM approach by using the gray-level and global shape information captured in the training set in a more robust and efficient way that simultaneously considers the effects of both during the deformation process.

2. Incorporate Local Shape Characteristics. In addition, local shape characteristics were incorporated into the formulation to improve the optimization of critical LP position movements. Modeling local shape characteristics for more accurate local LP updates is a unique approach that has not been pursued in any of the most recent ASM research literature.

3. Develop a Segmentation Confidence Metric: The lack of interaction between the GLM and SM creates another unfortunate situation in that no confidence measure is available to approximate how well the ASM segmentation algorithm has performed on any given image. In some fully automated applications, this renders the ASM technique useless. The reformulation of the optimization scheme provided a means to generate a performance confidence metric that makes possible the use of this new probabilistic deformable shape model in a fully automated application.

1.5 Document Organization

A thorough background of the published deformable model research is presented in Chap. 2. This background information includes the foundational research in the field, var-

ious algorithmic approaches that have grown out of this foundation, and a collection of applications in which deformable model algorithms have been used.

Chap. 3 presents the theory behind the development and application of ASM. The strengths and shortcomings of ASM are also outlined. Chap. 3 proceeds to describe the practical improvements made to the ASM algorithm and validates these improvements through various examples. It concludes by describing the fundamental shortcomings of ASM that limit its applicability.

Chap. 4 presents the theoretical development of the new probabilistic shape and appearance boundary-finding algorithm. This includes the compound Bayesian formulation of the objective function, followed by the derivation of the objective function, its gradient, and a description of the optimization of that function.

Experimental results for a variety of synthetic and real data are presented in Chap. 5, followed by conclusions and suggestions for future work in Chap. 6.

CHAPTER 2

BACKGROUND: SEGMENTATION METHODS

This section opens with a general discussion of the segmentation problem, then presents a convenient categorization of segmentation techniques. This categorization leads to a discussion of deformable models, followed by a detailed literature review of deformable model research.

2.1 The Significance and Difficulty of Segmentation

Image segmentation in computer vision applications can generally be described as the process of differentiating the object of interest in an image or a volume from its background. Computer vision researchers generally agree that segmentation is one of the most challenging and important tasks in the field of computer vision. Historically, many computer vision problems have been roughly broken down into a three-step process: (1) segmentation, (2) feature extraction and analysis, and (3) recognition (or classification). Segmentation, being the first step in the process in this serialized approach, has direct impact on all tasks that follow. If an object is not accurately segmented from its background, the subsequent tasks of feature extraction and classification may become meaningless or, even worse, inaccurate. Hence, the segmentation task is critical to the success of the entire analysis. To compound matters further, it is often the most difficult task of the three.

The difficulty of the segmentation task depends on many factors, including the predictability of the object of interest. If the object can be described *uniquely* within an image using features like intensity, texture, and shape, then the likelihood of successful segmentation is generally high. Two main problems can dramatically increase the difficulty of segmentation, and hence decrease the potential for success: (1) inconsistencies in the appearance of the object from image to image, and (2) background clutter and noise in the image. An object whose appearance varies wildly will confound a technique that makes narrowly defined assumptions about object appearance. In addition, background clutter and noise in an image may both obscure the object of interest and may confuse the segmentation technique by "distracting" the algorithm away from the object of interest.

All segmentation algorithms are constructed so as to incorporate some assumptions about the appearance or structure of an object. These assumptions may be very simple—e.g., the object has detectable boundaries or a particular internal intensity/texture—or they may be quite complex such as in the case of highly parameterized, model-based segmentation algorithms. For any given segmentation technique, these underlying assumptions drive the algorithm to a solution. If these assumptions are invalid for a given image, the algorithm is doomed to fail. As an example, thresholding, one of the simplest forms of segmentation, makes the assumption that the object of interest will have a unique range of intensity values within the image. If this assumption is violated, thresholding the image will result in a poor segmentation. Some of the most challenging segmentation tasks involve objects that have significant, but controlled variations in appearance. These tasks require a segmentation algorithm that makes specific assumptions about the object of

interest but at the same time has the flexibility to accommodate all of the acceptable object variations.

2.1.1 Medical Image Analysis

Medical image analysis is a good example of a field in which the appearance of objects, although somewhat predictable, may vary widely. The assumptions built into a segmentation algorithm must be able to accommodate this variation. Healthy organs will vary in shape, size, and position from subject to subject as well as within the same subject at different times. This is especially true of organs within the abdomen of the body such as the stomach, liver, kidneys, and intestines. Human beings have remarkable ability to segment an object from the background in an image, even when the object is partially obscured or has an inconsistent appearance. Although the exact sequence of events that occur within the human brain during such tasks is unknown, it is clear that a person's ability to perform a task improves with practice. This is due, at least in part, to the ability of the brain to recall knowledge from previous experiences and then apply that knowledge to subsequent tasks. If an untrained person looks at 100 labeled images of healthy kidneys, for example, that person will learn the acceptable variations of kidney shape, size, location, intensity, etc., and will likely be able to quickly identify healthy kidneys in subsequent images. It is precisely this type of a priori information that many developers of advanced segmentation techniques attempt to encapsulate into their algorithms.

2.1.2 Segmentation and Object Recognition

With a segmentation algorithm that incorporates significant amounts of a priori information, there is a point at which this algorithm can be considered both a segmentation *and*

an object recognition technique. If, for example, strict circular shape constraints are placed on the analysis result so that only objects that are circular are segmented within the image, then the algorithm accomplishes two tasks. First, it separates the object of interest from the background (segmentation), and second, it automatically, by default, identifies it as a circle (recognition). Both of these tasks together fall into the broader computer vision task of scene analysis. This leads directly to a discussion of one way to categorize scene analysis techniques.

2.2 Bottom-up versus Top-down Methods for Scene Analysis

One useful way to categorize the variety of different scene analysis techniques is by considering the main driving force behind the algorithm that leads to the final result. Some algorithms rely more on the data in the image to influence the solution, while other algorithms make up-front assumptions about the analysis problem that strongly influence the final form of the solution. In addition, a third class of algorithms use a compromise between the data and the model. These three classes of algorithms are sometimes referred to as (1) *bottom-up*, (2) *top-down*, and (3) *hybrid* analysis techniques [17].

At one end of the spectrum, bottom-up analysis algorithms are completely data-driven in that they rely entirely on the data within the target image to come to a conclusion. In purely data-driven strategies, no other sources of information are used as guides and/or constraints to arrive at a solution. Bottom-up approaches have the advantages of being flexible (e.g., they are applicable to a variety of object shapes and sizes) and requiring little or no a priori information to perform their task. If, however, the data are corrupted by noise, clutter, occlusions, etc., then bottom-up strategies will not perform well because

they have no other source of information to rely upon. Examples of data-driven analysis strategies are adaptive thresholding and region-growing. Generally, bottom-up approaches are simpler in theory and implementation than their counterparts in the top-down analysis category. Bottom-up techniques cannot be relied upon for object recognition because post-processing of the result is necessary to identify the object.

At the other end of the spectrum are top-down analysis algorithms that start with very specific assumptions about the characteristics of the target image, the object of interest, and, hence, the final solution. These assumptions are based upon a priori information about the particular analysis problem at hand. These assumptions might include expected object characteristics (e.g., shape, color, texture) as well as background characteristics (e.g., consistency, homogeneity, intensity). These assumptions are used in many cases to preconstruct some type of image model with characteristics and constraints that correspond to the up-front assumptions. In top-down strategies, the data are examined to determine how to perturb the model in an attempt to fit it to the object(s) of interest.

Very few scene analysis algorithms are purely bottom-up or top-down. Most fall on a continuum of hybrid approaches between these two ends of the spectrum. The next section reviews a class of analysis techniques called *deformable models* made up of a varying hybrid of bottom-up and top-down approaches.

2.3 Deformable Models for Image Analysis

Deformable model-based analysis techniques belong to a class of algorithms that use a priori information to preconstruct a model of the object of interest. Deformable model techniques are actually composed of a significant mix of top-down and bottom-up strate-

gies and, hence, fall into the hybrid category mentioned in the previous section. Fig. 2.1 illustrates the relationship between deformable models and the top-down/bottom-up scene analysis techniques. Most deformable models accomplish both tasks of object segmentation and object recognition simultaneously. This is a result of the a priori information that is used to constrain the deformable model and, hence, the final solution. The algorithm will segment only the objects allowed by the preconstructed model. This is a tremendous advantage of these techniques because it prevents the propagation of error throughout the computer vision system that follows the traditional serialized approach of segmentation, feature extraction, and recognition (or classification). In the context of deformable model discussions, when the term *segmentation* is used in this document, it will imply object recognition as well.

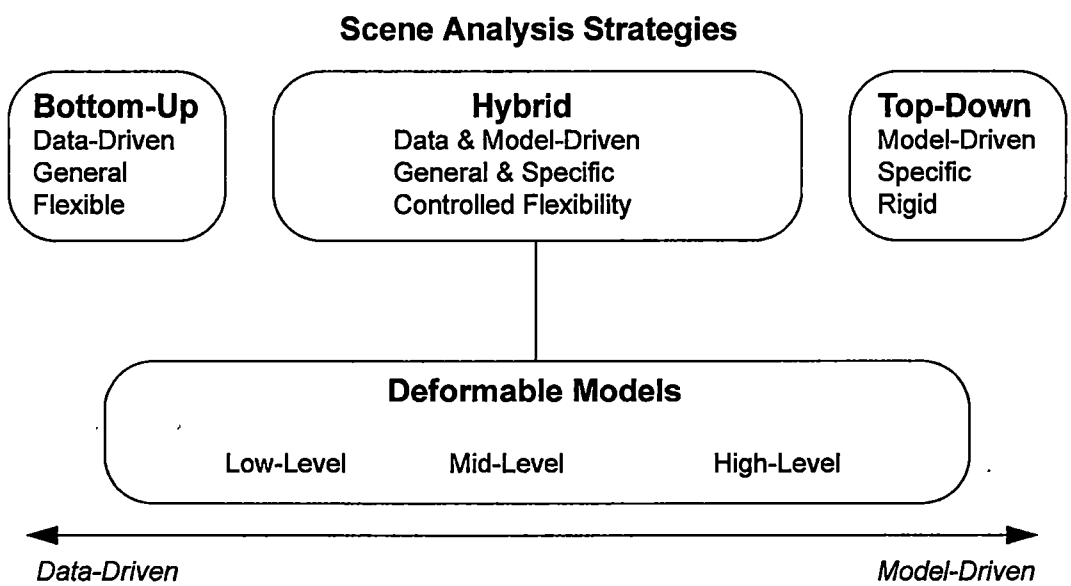


Fig. 2.1 Illustration of top-down vs bottom-up scene analysis techniques. Note that deformable models fall under the "hybrid' category of techniques that rely on a combination of data-driven and model-driven strategies.

In most cases deformable model approaches use a model that is a representation of either the boundary of an object in a 2D image or the surface of an object in a 3D volume. The shape of the model depends on a set of parameters that are adjusted and tuned via an optimization process to be compatible with the information about the object as defined by the current image data. These predetermined parameters control the general form of the resultant model and are typically referred to as *internal* constraints (or internal energy). The image data typically generate forces, referred to as *external* constraints (or external energy), that somehow attract or repel the model to a state that optimizes the combination of internal (model-driven) and external (data-driven) forces. The goal of a deformable model-based algorithm is to find a good compromise between these internal and external forces.

The remainder of this section introduces three classes of deformable models, provides background on past and current research in the field, and finally addresses the topic of what types of data within an image drive a deformable model to a particular segmentation solution. For additional reviews of deformable model research consult references [56] and [71].

2.3.1 Low-, Medium-, and High-Level Models

The deformable model may range from the simple to the complex depending on the type and amount of a priori information available about the object. Deformable models can be loosely categorized into low-level, mid-level, and high-level classes depending on the level of control that the model imposes on the segmentation result. As illustrated in Fig. 2.1 low-level models are used in more bottom-up segmentation strategies in that the

information in the current image has the most significant control over the result. Alternatively, high-level models are used in more top-down segmentation strategies. The preconstructed model has the most significant control over the segmentation result, and the image data only accounts for slight perturbations in the model. Mid-level models are used in several deformable model segmentation algorithms where there is varying trade-off between the amount of data-driven versus model-driven control.

2.3.2 Research in Deformable Models

This section presents the significant highlights in deformable model research over the past 12 years. While not comprehensive, this summary does cover the major subject areas and highlights many of the important contributors to the field.

Low-Level Models

Although deformable model research dates back to 1973 ([99], [100]) the original research on piece-wise splines by Kass, Witkin, and Terzopoulos [57] is considered by most to be the foundational work in the field of deformable models. Better known as "snakes," or active contours, these deformable models are considered low-level because they are constrained only by local smoothness parameters and have no overall shape constraints. These researchers coined the internal and external energy concepts that are still widely used in current algorithm development strategies for deformable models. Prior to this Gritton and Parrish introduced a form of snake called a bead chain [49], but the opti-

mization of this chain was not as sophisticated as that by Kass and colleagues. Their energy function is modeled by internal and external energy forces as follows:

$$E_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 E_{int}(v(s)) + E_{ext}(v(s))ds, \qquad 2.1$$

where

$$E_{ext}(v(s)) = E_{image}(v(s)) + E_{con}(v(s)), \qquad 2.2$$

$v(s)$ is the parametrized contour $[x(s), y(s)]$, E_{int} represents the internal energy of the spline due to bending, E_{image} represents forces imposed by the target image, and E_{con} represents external constraint forces that the user may impose. For snakes, the internal energy is related to the smoothness of the active contour, and the external energy is based on edge information in the target image.

Because snakes are more of a bottom-up, data-driven concept, they are very useful and flexible in images with "clean" data (noncluttered images with well-defined objects and continuous borders). They were designed to be interactive and require good initial guesses to converge to the correct solution. The more corrupt and unpredictable the data, the better the initial guess must be for success. They tend to become trapped in local minima, especially when the image is cluttered with spurious edges. If the global shape of the object is known a priori, then there is not a convenient way to include this knowledge in the snake deformation process.

The internal and external energy concepts for snakes are important to almost every other deformable model approach. The internal energy term can be represented by

$$E_{int} = \frac{[\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2]}{2}, \qquad 2.3$$

where $v_s(s)$ and $v_{ss}(s)$ are the first- and second-order derivatives of v. From this energy equation one can see that the values of α and β control the first- and second-order smoothness of the spline $v(s)$.

If we assume the external energy force is composed only of image forces with no additional user-imposed constraints ($E_{con} = 0$), then the only contribution is by E_{image}. In fact, in most snake implementations, the image is the only contributor to the external energy terms. In practice, a variety of image-derived forces are used, but the most common form of image energy is given by edge content

$$E_{ext}(x, y) = E_{edge}(x, y) = -|\nabla I(x, y)|^2, \qquad 2.4$$

where $I(x, y)$ is the image of interest.

Snakes and their derivative works have been successfully applied to a wide variety of segmentation problems. Some examples are mouth detection [57], object tracking ([4], [7], [8], [29], [42], [53], [65], [93]), image retrieval [27], medical image segmentation ([35], [72]), and more specifically, brain image segmentation [3] and ultrasound imagery [64].

Much work has been done to improve upon this original active contour model. Tagare [91] employs an orthogonal curve constraint for better control of snake motion and, hence, better controls the global shape of the object. This strategy also reduces the search space

so that optimization can be achieved more quickly. Terzopolous and Metaxas [92] take this idea a step further and employ a superquadric ellipsoid model on a global scale along with membrane splines at the local level. The result is a model with flexibility similar to snakes to capture detail, but with an overall global shape constraint. Chandran and Potty [15] use dynamic programming to improve the snake-fitting optimization process in terms of speed and accuracy. Other notable alternative formulations of the snake energy can be found in references [2], [37], [40], [76], and [101]. In addition, snakes have been generalized to 3D in references [9], [19], [20], [21], [74], [92], and [94]. Although some promising results are reported, a comprehensive analysis of their effectiveness on 3D segmentation problems has yet to be reported.

The classical active contours are topologically limited in that they cannot split, merge, fold over on themselves, etc., so work has been done by McInerney and Terzopoulos [72] and by DeCarlo and Metaxas [26] to develop a topologically adaptable snake. In this approach, a decomposition grid is iteratively re-parameterized to allow topological changes of the contours. O'Donnell, M. Dubuisson-Jolly, and Gupta [77] have adapted snakes for application to branching cylindrical structures. Theoretically, the development of snakes is not conducive to topological variation, and published approaches are "add-on" adaptations that attempt to overcome this limitation.

The deformable models known as geodesic snakes lend themselves better to handling topological changes. This category of low-level approaches to boundary-based image segmentation (Casselles et al. [10], [11], [12]; Malladi, Sethian, and Vemuri [68]; and Sapiro [84]) evolved out of the field of differential geometry, where curve evolution and geometric flow theory are well-characterized mathematically but had not been adapted and

applied to the field of image segmentation until recently. As described by Whitaker [98], these 2D geodesic snakes can be characterized as a level set (contour) of an image (or an isosurface of a volume) and therefore can take on any shape allowed by the discrete pixel (voxel) grid in the image. A geodesic snake can split and merge to allow detection of multiple objects simultaneously. Also, these level sets can be extended directly to 3D [58] because the math supports this without any additional theoretical development.

Geodesic snakes are very flexible in terms of overall shape and are good at capturing the details of complex objects but, like other more bottom-up approaches, are less accurate in the presence of noise and image clutter. Also, incorporation of a priori information (e.g. global shape constraints) to limit variability has not been actively researched and/or reported. Most implementations are data-driven, with some simple smoothness constraints. Because geodesic snakes are non-parameterized, they have topological flexibility, but the results are also more difficult to interpret. Although one is likely to know the general form of the objects delineated by geodesic snakes based on the current application, their nonparameterized form requires additional postsegmentation analysis to complete the task of object recognition.

A detailed description of geodesic snakes is beyond the scope of this work, but Niessen, Romeny, and Viergever [75] present a good summary of the research in this field along with a comparison of implicit models (geodesic snakes) versus explicit models (classical snakes) as applied to medical imagery. Application of geodesic snakes to brain anatomy analysis [6] and bone segmentation [67] have been reported in the literature.

Mid-Level Models

Many deformable model algorithms rely heavily on both the model and the image data to converge to a segmentation solution. These models are used in algorithms that would be considered a substantial mix of bottom-up and top-down approaches. These mid-level models tend to be more generally applicable to various segmentation problems. They lack the flexibility and ability to capture the minute details that the low-level models do, but they are more flexible than the well-constrained, high-level models. All of the well-known mid-level model approaches allow the user to somehow define the degree of flexibility the model will possess. This model flexibility may be characterized by either a user-created training set of manually segmented objects or a collection of user-selectable model parameters that must be valued based on a priori knowledge of object characteristics.

Staib and Duncan [87] proposed a model parameterized by an orthonormal basis of sinusoids. The parameters of the model are Fourier coefficients that provide a convenient and theoretically elegant way to control boundary smoothness. Including higher-order coefficients in the model allows for higher-frequency components and, hence, increased local boundary roughness. Allowable variations of these coefficients are determined via analysis of a training set and are used to constrain the final state of the contour. A couple of limitations noted about this approach are that sinusoidal basis functions are inefficient in approximating discontinuities such as corners, and they also lack the straightforward relationship between Fourier parameters and object shape. A Fourier representation has also been used for segmenting magnetic resonance imaging (MRI) images and volumes in [90].

Pentland and Sclaroff ([79], [85]) use a finite-element model of a base shape (e.g., an ellipsoid) and mathematically derive the modes of shape vibration where low-order modes control global shape and high-order modes control local shape variations. Although a training set is not needed for this method, it does require the user to impose knowledge about the application by setting model parameters. Depending on the application, the user must choose the number of vibration modes allowed during the deformation process. These modes must be determined and set manually before application of this technique can be successful. For example, to model the human head the authors use the first 30 modes of vibration.

Discrete, or point-based, shape representations are commonly used in shape models because of their convenience and their simple, yet effective ability to characterize shapes. Points are easy to keep track of, and shape measurements can be made simply and quickly using a point representation. It was not until the work of Cootes et al. ([22], [28]) that a mid-level deformable model approach was introduced that made extensive use of training data in the form of user-defined LPs that delineate significant shape features. The authors coined the term "active shape model" (ASM) to describe these contours. ASM is shape-constrained by the measured shape variations within the training set and, at the same time, are attracted to edges in the target image. An optimization strategy is used to balance the shape constraints imposed by the model and the attraction force generated by object edges within the target image. The optimization strategy used in ASM is a brute-force iterative scheme.

ASM has been successfully applied to face recognition, knee cartilage segmentation [22], human vertebra segmentation [86], heart ultrasound imagery [52], and brain MRI

segmentation [51]. Extension of ASMs to 3D is straightforward but currently impractical due to the difficulty of generating a sufficient 3D training set. Given that ASMs are part of the foundation for the work proposed here, a more detailed account of their advantages and disadvantages and the theory behind them is given in Chap. 3.

As mentioned in the Introduction, only two substantial adaptations of the ASM approach have been reported in the literature. First, Kervrann and Heitz [59][60][61] present an ASM segmentation approach that models global shape characteristics combined with Markov modeling on the local scale that promotes boundary smoothness. A joint Bayesian estimation function is created and optimized, leading to a MAP solution. The focus of Kervrann and Heitz's work is on motion-tracking applications. The second ASM adaptation, by Wang and Staib [96] [97], also uses the ASM approach for segmentation but performs the ASM fitting procedure using a MAP approach similar to the theory behind the Fourier boundary model representation [87] discussed previously. Another difference between Wang and Staib's approach and ASM is that Wang and Staib include the pose parameters (translation, scale, and rotation) as part of the parameter set that is optimized. In the approach of Cootes et al., pose parameters are calculated separately from the shape parameters. Both of these ASM adaptations rely solely on edge information in the target image as the external attraction force. Duta and Sonka ([33], [34]) have also adapted the original ASM approach and have slightly modified the optimization scheme. Fritsch et al. [39] also use a statistical model of shape, but extend the boundary representation from a collection of LPs to slightly more complex geometric representations. Finally, noting that ASM is a statistical approach, Grenander [46] and Grenander and Keenan [47] have published useful information on the role of statistical approaches for image understanding.

High-Level Models

High-level models (also referred to as deformable templates) incorporate stringent constraints on the final shape of the object, both globally and locally. There is little flexibility in how the data can influence the model parameters. High-level models can be extremely useful in applications where one knows precisely what object is to be segmented, and the shape characteristics of that object are well understood. As one would expect, since the high-level algorithms rely heavily on model characteristics and rely less upon the actual data, they are claimed to work well in the presence of noise, clutter, and occlusions. An example of a high-level technique not typically grouped in the category of deformable models is the generalized Hough transform [5], [32]. In fact, the generalized Hough transform is commonly used in other low-, mid-, or high-level deformable model approaches to initialize the first instance of the model (e.g., in [62]).

Examples of high-level models used in segmentation algorithms are parameterized analytic functions (e.g., polynomials) whose coefficients are allowed to vary within imposed constraints [54]. Yuille, Hallinan, and Cohen [102] used these parameterized templates to address the problem of face recognition. Mirhosseini, Yan, and Lam [73] also employed a deformable template approach for mouth boundary detection. Deformable templates have also been used for image restoration [1], robot vision [16], character recognition [55], general occluded object detection [69], and vehicle segmentation [30]. What each of these approaches has in common is that the template is strongly based on the shape of the object of interest, and they are therefore very application specific. New templates must be developed for new applications. Because these templates are highly constrained, their optimization is relatively straightforward and very fast.

2.3.3 Boundary versus Region-based Deformable Models

To this point, not much attention has been given here as to specifically what data within the image drive deformable models to a particular solution. In the vast majority of cases, deformable model algorithms are developed and implemented so that the external attraction force is generated by edge information within the target image. Although edge information is an essential component for many segmentation problems, relying solely on edge information will likely lead to problems in at least the following three situations:

- **Blurred and/or faint edges** will lead to a weak attraction force for the deformable contour.
- **Occluded or partially missing edges** can be troublesome in that parts of the deformable contour will lock in on the strong edge information while the rest of the contour may to "wander" aimlessly in search of the missing edges. This can distort the final result.
- **Spurious edges** generated by clutter within an image will attract the deformable contour away from the desired edge.

These situations motivate the inclusion of additional external energy forces that can help attract a deformable model to the desired solution.

Surprisingly, the research into this notion of incorporating new external energy forces has been quite limited compared to purely boundary-based solutions. Poon et al. [80] and Ronfard [82] have extended the active contour model to allow inclusion of additional object features that can contribute to the discriminant function. Region-based information

is used to generate an energy term that is included in the energy objective function given by Eqn. 2.1 [25]. Chakraborty, Staib, and Duncan [13] have extended the Fourier-parameter, MAP approach in reference [87] to include regional intensity information within the object. Gauch, Pien, and Shah [41] present a two-phase approach of region-based segmentation to initialize a deformable boundary, followed by optimization of that boundary using traditional active contour methods. Zhu and Yuille [104] have also integrated region information by combining aspects of both region-growing and active contour models with a balloon inflation force. In the field of ASM, Cootes, Edwards, and Taylor [23] have modified their original approach to include gray-level information sampled from the interior of the target object. Shape and gray-level intensity parameters are all grouped together, and a principal component analysis (PCA) is performed on data extracted from a training set to identify the most significant modes of variation. The resulting model is called an active appearance model (AAM). AAMs are optimized in a manner similar to the ASM optimization approach. This technique has been recently applied to face recognition [63] and general object localization [103].

CHAPTER 3

THE ACTIVE SHAPE MODEL AND PRACTICAL IMPROVEMENTS

Recall (from Sec. 1.1) that the motivation for this work is the need for an algorithm that performs automatic segmentation and recognition of semirigid objects within complex backgrounds. ASM has been identified as a good starting point for this research, so this chapter outlines the theory behind its development. It then presents some of the strengths and shortcomings of the approach, followed by a description of several practical improvements that were designed and implemented. The chapter ends with some examples and a discussion of the need for fundamental changes in the existing ASM boundary-finding algorithm.

3.1 ASM Theory

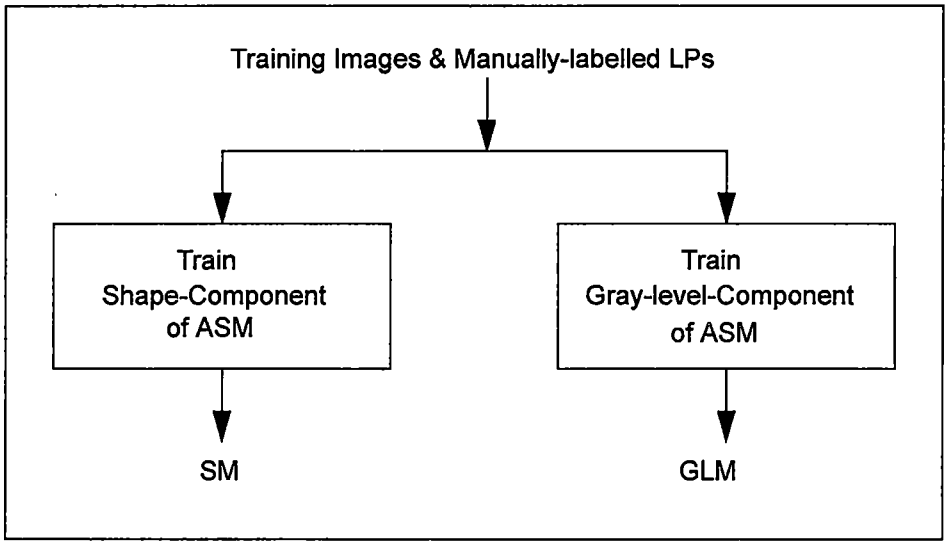
It was not until the work of Cootes et al. [22] that a deformable model approach was introduced that made extensive use of training data in the form of user-defined LPs that delineate significant shape features. These authors coined the term ASM, or "smart snakes," to describe these contours because of the a priori knowledge that is built into them via a training procedure. ASM is shape-constrained by the measured shape variations within the training set and, at the same time, is attracted to edges in the target image. An optimization strategy is used to balance the shape constraints imposed by the model and the attraction force generated by object edges within the target image.

A simplified view of the ASM training and optimization process is shown in Fig. 3.1 to illustrate the main components of the algorithm. The remainder of this section breaks the theory down into more detail. The training process consists of two separate tasks: (1) training the shape model (SM) and (2) training the gray-level model (GLM). Together, SM and GLM constitute the complete ASM. Both of these training processes require a set of training images with manually labelled LPs that define the boundary of the object(s) to be segmented. SM constrains the final shape of the resulting ASM boundaries to look like those represented by the training set. Comparing this technique to the original work in active contours by Kass, Witkin, and Terzopoulos [57], we see that SM represents the internal forces. Optimizing the fit of SM to a test image is comparable to minimizing the internal energy term in Eq. 2.1. The GLM is used to update the position of the ASM boundary under the influence of the data within the image. Optimizing the GLM can be compared to minimizing the external energy term of Eq. 2.1. Note that fitting the ASM to a new image is an iterative process in which the GLM and SM are *independently* applied to the current set of LPs. This independence of GLM and SM is important to note because it leads to some problems with the ASM technique that can be solved by improving the level of communication between these two models. This is an important motivation for the proposed research to be presented later in Sec. 4.1, but it is important to first outline the current ASM theory.

3.1.1 Training via Landmark Points (LPs)

The training process for the model consists of gathering a substantial collection of sample images and manually placing LPs in the image that define the location of impor-

(a) ASM Training Procedure



(b) ASM Optimization Procedure

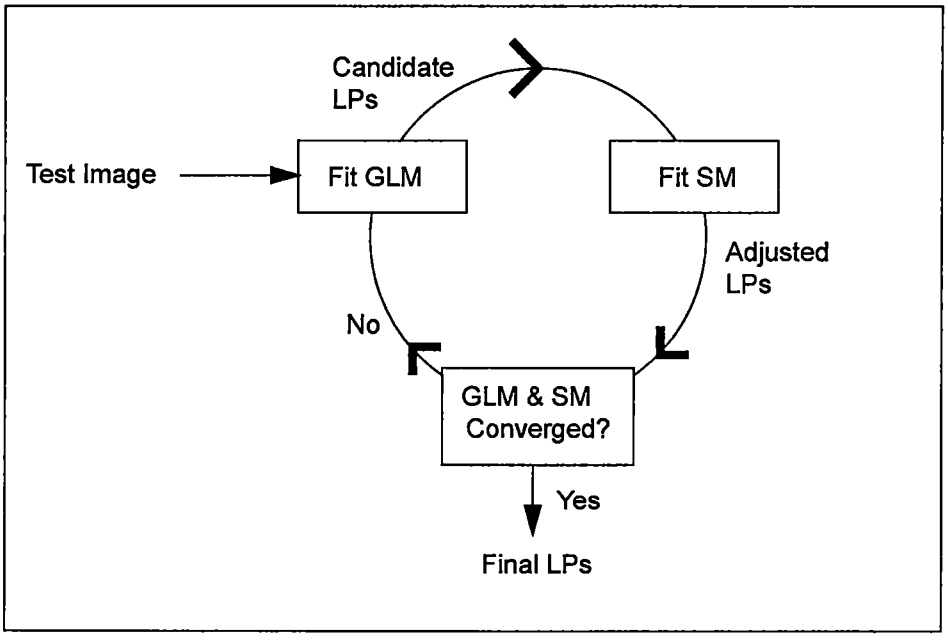


Fig. 3.1 Simplified ASM flow diagram. Shown are (a) the ASM training procedure and (b) the ASM optimization procedure.

tant and/or unique points of the object of interest. Appropriate choices for these LPs are typically along the boundary of the object of interest and should capture unique features of the boundary such as corners, points, and high-contrast areas. It is also appropriate to mark additional, equally spaced points along the border between these unique features. The image in Fig. 3.2 shows a sample training image labelled with LPs.

For a 2D application, the N LPs for the i^{th} training image can be written as a vector, $\boldsymbol{p}_{train}(i)$, of length $2N$:

$$\boldsymbol{p}_{train}(i) = (x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N)^{\mathrm{T}} \tag{3.1}$$

One of these vectors is created for each image in the training set to form a set of training vectors. There must be correspondence between the LPs from one training image to

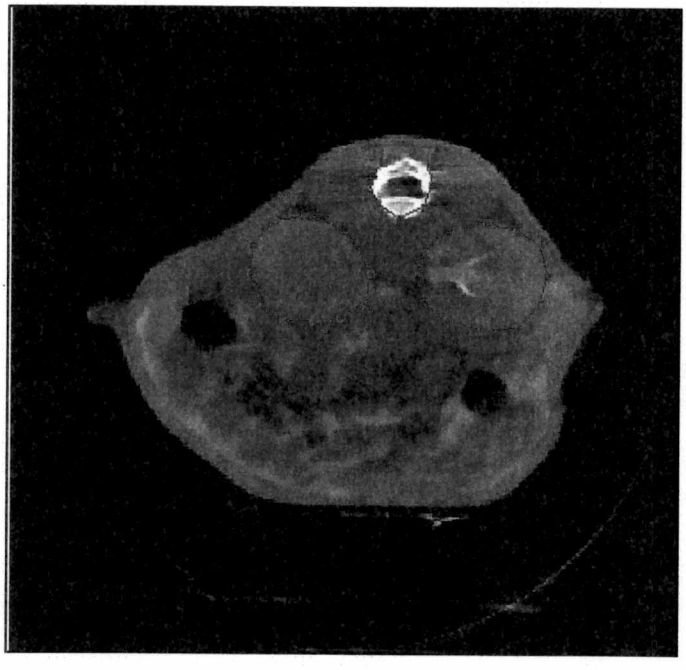


Fig. 3.2 Example of a mouse kidney training image. Pictured is an X-ray CT cross-section image with the manually placed landmark points (LPs) along the kidneys and spine borders.

another. That is, a given landmark point, (x_i, y_i), should be placed on the same image feature across all images in the training set. This is important for the next step in the training process: alignment of the set of LP vectors within the training set.

3.1.2 Training Set Alignment via Procrustes Analysis

Before a statistical analysis can be performed on a set of training vectors, they must be aligned to a common coordinate frame. Cootes and associates [22] accomplish this using an iterative process with the key step being Procrustes analysis [45]. Procrustes analysis is a technique used to find the pose parameters (translation, scale, and rotation) that align two sets of corresponding LPs. This technique first translates one set of LPs such that its center of gravity is aligned with that of the first set. Then the same set of LPs is rotated and scaled using parameters that minimize the sum of squared distances between corresponding LPs in each set. Cootes et al. embed this alignment procedure within an iterative framework as follows:

1. Align the centers of gravity by translating the center of gravity for each LP training set to the origin.
2. Choose one training sample (e.g., the first one) as initial mean shape, $\bar{\boldsymbol{p}} = \boldsymbol{p}_{train_i}$, and normalize it so that $|\bar{\boldsymbol{p}}| = 1$.
3. Call this current estimate of the mean $\bar{\boldsymbol{p}}_{def}$ to define the default orientation.
4. Align all training samples, $\boldsymbol{p}_{train_i}$, to the current mean, $\bar{\boldsymbol{p}}$, for all $i = 1 \ldots M$ (M = number of samples in training set). The aligned set of training samples is defined as $\hat{\boldsymbol{p}}_{train_i}$.

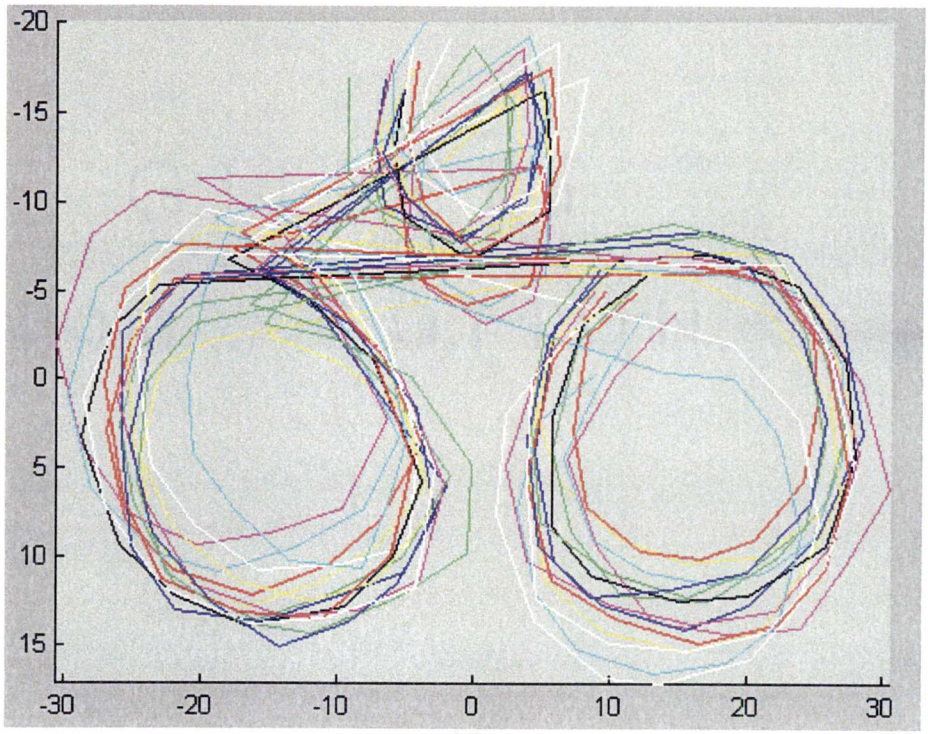
Minimize sum of point-to-point distances to the mean, D, via Procrustes analysis.

5. Reestimate $\bar{\boldsymbol{p}}$ from the aligned training samples.
6. Align $\bar{\boldsymbol{p}}$ with $\bar{\boldsymbol{p}}_{def}$ and normalize.
7. Check for convergence. If convergence has not occurred (i.e., $\bar{\boldsymbol{p}}$ continues to change), return to step 4.

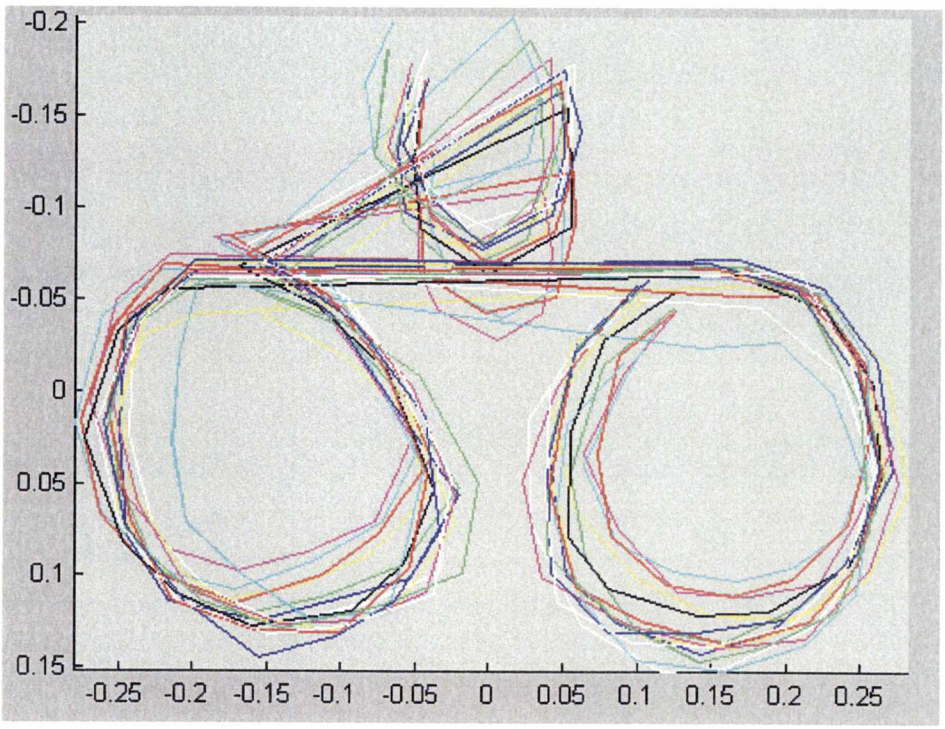
This procedure typically converges after a few iterations. The aligned shapes can then be analyzed statistically to determine the significant modes of variation in the training set as described in the following section. An example of a set of shapes for the kidney images before and after Procrustes alignment is shown in Fig. 3.3.

3.1.3 Creating ASM via Principal Component Analysis (PCA)

The ASM must capture the significant variations in shape represented in the collection of manually labelled training samples. This variation can be represented in a convenient way by performing a principal component analysis (PCA) on the aligned training samples. PCA (also known as the Karhunen-Loeve transform [31]) transforms the original data into a new space in which the significant modes of variation in the training data occur along the orthogonal axes. The orientation of the axes is given by the eigenvectors of the covariance matrix of the original training samples, and the corresponding eigenvalues provide a value for the variance of the training data in the direction of those eigenvectors. This representation is very useful because one can quickly and easily identify the direction and extent of the variation in the training data set. As a result, one can put constraints on ASM



(a)



(b)

Fig. 3.3 An example showing a set of kidney and spine LPs: (a) with aligned centers-of-gravity and (b) after complete alignment via Procrustes Analysis.

based on the significant modes of variation in the training data. PCA is carried out via the following steps:

1. Compute the mean of the aligned training samples,

$$\bar{\boldsymbol{p}} = \frac{1}{M}\sum_{i=1}^{M}\hat{\boldsymbol{p}}_{train_i}. \qquad 3.2$$

2. Compute the covariance, $\boldsymbol{C}$, of the training samples,

$$\boldsymbol{C} = \frac{1}{M-1}\sum_{i=1}^{M}(\hat{\boldsymbol{p}}_{train_i}-\bar{\boldsymbol{p}})(\hat{\boldsymbol{p}}_{train_i}-\bar{\boldsymbol{p}})'. \qquad 3.3$$

3. Determine the eigenvectors, ϕ_i, and eigenvalues, λ_i, of $\boldsymbol{C}$. Sort the eigenvalues and eigenvectors from largest to smallest eigenvalue.

4. Compute the total variance (sum of eigenvalues),

$$V_T = \sum_{i=1}^{2N}\lambda_i. \qquad 3.4$$

5. Choose the first t eigenvalues such that their sum captures at least a large fraction, f_v, of the total variation in the training set,

$$\sum_{i=1}^{t}\lambda_i \geq f_v V_T, \qquad 3.5$$

where f_v would typically be a number like 0.98 (i.e., captures 98% of the variation in the training set).

6. Model any new sample $\hat{\boldsymbol{p}}$ as

$$\hat{\boldsymbol{p}} = \bar{\boldsymbol{p}} + \Phi \boldsymbol{b}, \tag{3.6}$$

where the columns of Φ are the first t eigenvectors corresponding to the largest t eigenvalues.

The eigenvectors within the matrix Φ define the axes of a new t-dimensional vector space to which we can transform our aligned training vectors, $\hat{\boldsymbol{p}}_{train_i}$. The elements of $\boldsymbol{b}$ define the extent of the shape along the axes formed by the corresponding eigenvectors. Hence, these elements of $\boldsymbol{b}$ define the parameters of the ASM, so that fitting the ASM to a new set of LPs is optimizing its parameters, the vector elements of $\boldsymbol{b}$. The process of finding this new set of candidate LPs is described in the following section.

3.1.4 Training the GLM

To find a new set of candidate LPs to submit to the ASM, one first has to create GLM for all of the LPs. This GLM takes the form of an objective function that is minimized to find new LPs in new test images. The GLM training process is as follows:

1. For each LP, sample the gray-level profile (k pixels on either side of the LP) along the normal to the boundary that passes through that LP. Take the derivative of the profile and normalize it by the absolute sum of the element values. This results in a $2k + 1$ length vector, $\boldsymbol{g}_d$, for each LP.
2. Calculate the covariance, $\boldsymbol{S}_d$, of all derivative profiles, $\boldsymbol{g}_d$, for a given LP across all images in the training set.

3. Calculate the mean derivative profile, $\bar{g}_d$, by averaging all the profiles for an LP across all images in the training set.
4. Create the GLM component for the LP as an objective function of the form

$$f(g_d) = (g_d - \bar{g}_d)^T S_d^{-1} (g_d - \bar{g}_d), \qquad 3.7$$

where g_d is, in a testing (not training) mode, the derivative profile from a candidate LP in a new test image.

3.1.5 Finding a New Set of LPs

The objective function in Eq. 3.7 is the squared Mahalanobis distance from the mean gray-level profile, $\bar{g}_d$, to the new derivative profile, g_d. To allow for a multiresolution search for candidate LPs, the GLMs is created at several resolutions of the training images. This leads to the creation of one objective function for each LP at every search resolution. The images and overlaid LP normal profiles in Fig. 3.4 illustrate the process of creating the GLMs for multiple resolutions. When searching for new candidate LPs, these objective functions must be minimized. This search process is accomplished as described in the following pseudo-code:

1. For each resolution of search
2. For each LP defined by the training set:
3. Sample gray-level profile, g_{samp}, of length l, where $l > 2k + 1$, at the current LP along normal to shape boundary.
4. For each possible subvector, g_d, of g_{samp} (total of $l - 2k$ subvectors) take the deriva-

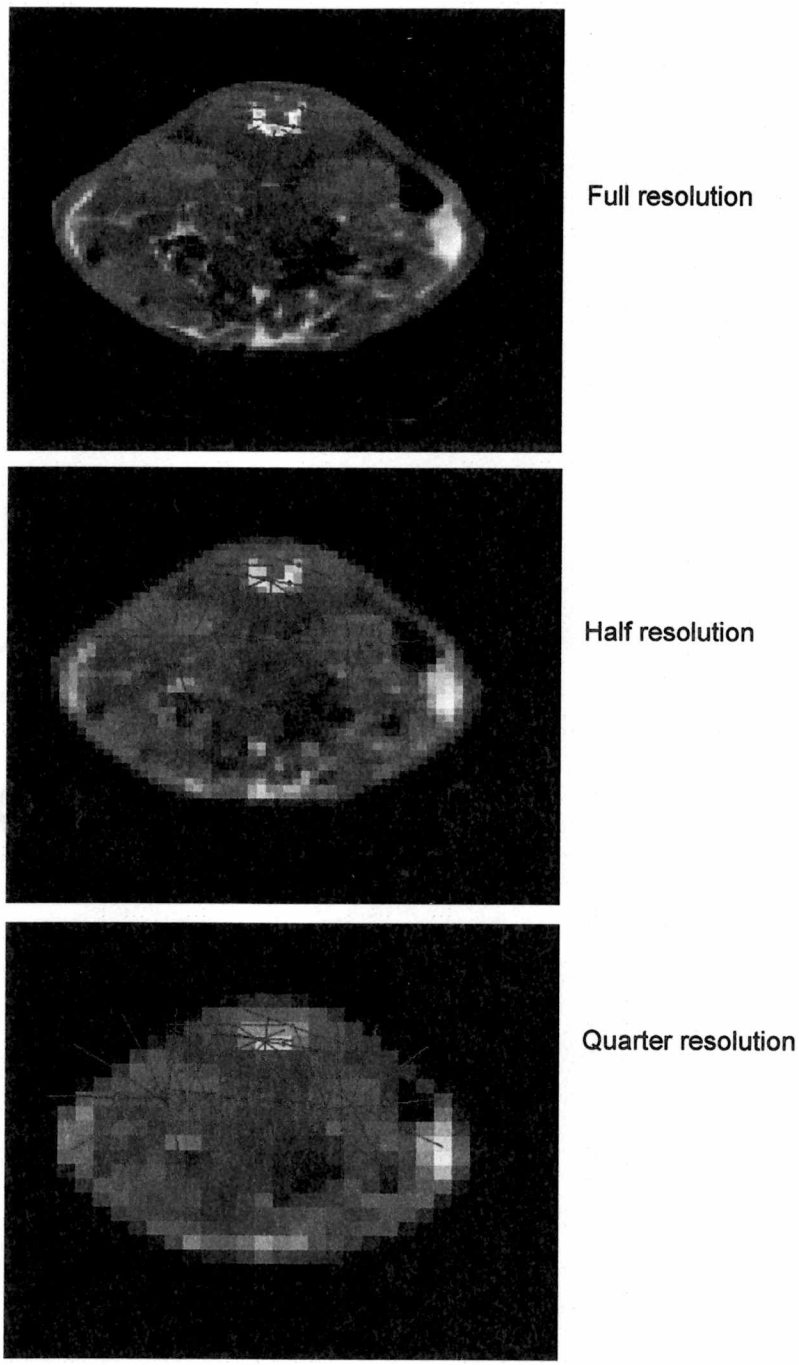


Fig. 3.4 Images with overlaid normal profiles. The images illustrate how grey-level profiles are sampled around landmark points at multiple resolutions of the input image (three resolutions in this case).

tive and normalize $\boldsymbol{g}_d$.

5. Evaluate the GLM objective function, $f(\boldsymbol{g}_d)$, for the current sub-vector.
6. Next subvector.
7. Record the minimum value of the GLM across all subvectors. The location of this minimum will be the location of the new candidate LP.
8. Next LP.
9. Next resolution.

After this process is complete, we have a new set of candidate LPs that must be submitted to a modeling procedure (SM) that attempts to find an instance of SM that most closely fits, within the constraints identified by PCA, this new set of LPs. This procedure is outlined in the following section.

3.1.6 Fitting a Model Instance to a New Set of Points

We call our new set of candidate LPs $\hat{\boldsymbol{p}}$. The goal of the following procedure to is minimize the difference between the set of LPs suggested by the GLMs, $\hat{\boldsymbol{p}}$, and an instance of ASM by adjusting the ASM parameters contained in the vector $\boldsymbol{b}$. We can represent the difference between these two as the sum of square differences between them and, hence, minimize the following expression:

$$D = \left| \hat{\boldsymbol{p}} - T^{-1}_{X_t, Y_t, s, \theta}(\bar{\boldsymbol{p}} + \Phi \boldsymbol{b}) \right|, \tag{3.8}$$

where the second term on the right-hand side is an instance of ASM translated, scaled, and rotated back into the image coordinate frame. The steps to fit the model are as follows:

1. Initialize the ASM parameters, $\boldsymbol{b}$, to zero.
2. Generate instance of ASM,

$$\hat{\boldsymbol{p}} = \bar{\boldsymbol{p}} + \Phi \boldsymbol{b}. \tag{3.9}$$

3. Find pose parameters (X_t, Y_t, s, θ) that best map $\hat{\boldsymbol{p}}$ to $\boldsymbol{p}$ via Procrustes analysis, where s is the scale factor and θ is the rotation angle.
4. Project $\hat{\boldsymbol{p}}$ into ASM coordinate frame,

$$\boldsymbol{p} = T_{X_t, Y_t, s, \theta}(\hat{\boldsymbol{p}}). \tag{3.10}$$

5. Update the ASM parameters— i.e, project the current set of candidate LPs into the PCA subspace,

$$\boldsymbol{b} = \Phi^T(\hat{\boldsymbol{p}} - \bar{\boldsymbol{p}}). \tag{3.11}$$

6. If convergence has not yet occurred (ASM parameters $\boldsymbol{b}$, s, and θ continue to change), return to step 2.
7. If convergence has occurred, then apply the limits to $\boldsymbol{b}$ as determined from the training data such that

$$-s\sqrt{\lambda_i} \le b_i \le s\sqrt{\lambda_i}, \ \forall i = 1, \ldots, t. \tag{3.12}$$

where s is the number of standard deviations away from the mean that each parameter is allowed to vary. The limits imposed by Eq. 3.12 can be thought of as "box limits" because they confine the shape parameters, $\boldsymbol{b}$, to a t-dimensional hypercube, centered at the origin, with dimensions equal to $2s\sqrt{\lambda_t}$. Typically, a value of $s \cong 3$ (includes 98% of training set

shape variability for Gaussian data) works reasonably well. Alternatively, one can impose shape constraints such that

$$\sum_{i=1}^{t} \frac{b_i^2}{\lambda_i} \leq M_t; \tag{3.13}$$

which constrains the shape vector, $\boldsymbol{b}$, to fall within a hyperellipsoid whose size is determined by the Mahalanobis distance, M_t. An example of a $\boldsymbol{b}$ vector and the constrained version, $\boldsymbol{b}_c$, as limited by Eq. 3.12 is shown in Fig. 3.5.

Applying the box limits as described above can be viewed as concatenating the variation of the shape along each eigenvector to make sure it fits within the limits imposed by the training set. This concatenation is performed with no regard to what the underlying image data looks like. Clearly, a more continuous penalty for global shape variation would be more appropriate. These are potential problems with the ASM approach and will be addressed later in Chap. 4 during the development of the new statistical SM.

Once we have converged upon a set of ASM parameters and then shape-constrained it as just described, we generate an instance of ASM. This instance of the model is then translated, scaled, and rotated back to the image coordinate frame as follows:

$$\boldsymbol{p}_{new} = T^{-1}_{X_p, Y_p, s, \theta}(\bar{\boldsymbol{p}} + \Phi \boldsymbol{b}). \tag{3.14}$$

This new set of LPs, $\boldsymbol{p}_{new}$, is, in a sense, a compromise between the set of LPs generated using GLM and the constraints imposed by SM on that set of LPs.

3.1.7 ASM Iterative, Multiresolution Search

The two model optimization procedures (GLM and SM) described in the previous section alternate in finding new sets of LPs. The SM procedure generates $\boldsymbol{p}_{new}$ and passes it

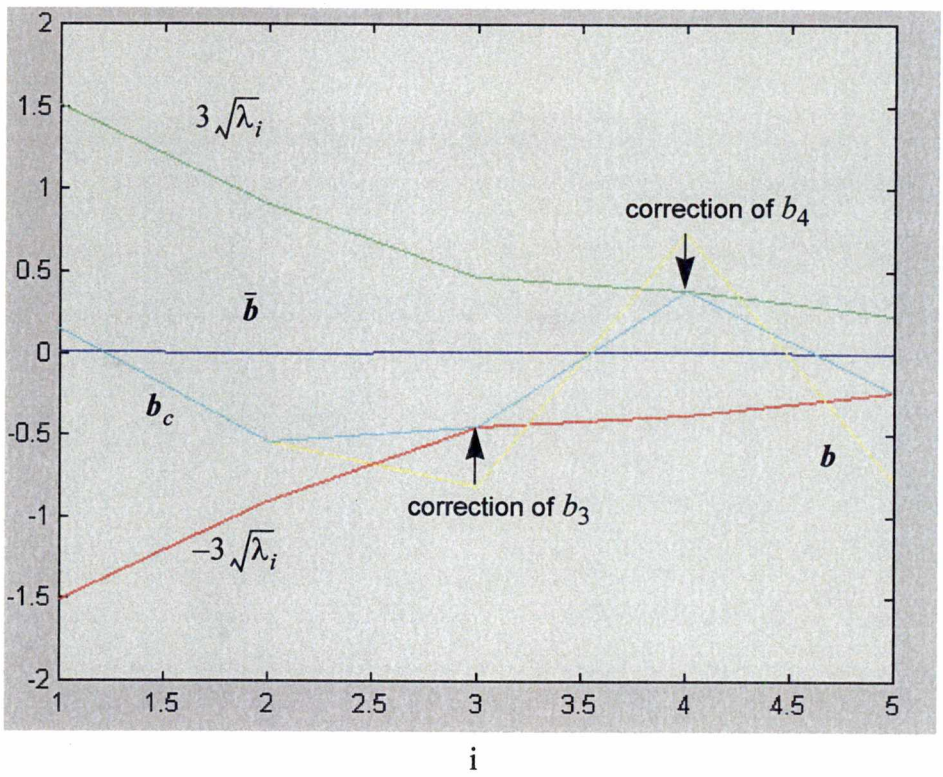


Fig. 3.5 Plot showing the mean shape vector. Also shown are the allowable shape envelope, a calculated shape vector, and the constrained version of that shape vector.

to the LP search routine, where the GLM is optimized once again to find the next set of candidate LPs. These new points (still called $\boldsymbol{p}_{new}$) are passed back to the SM routine to find the model that best fits $\boldsymbol{p}_{new}$. This process of "ping-ponging" back and forth between GLM and SM optimization routines continues until the set of points generated by SM does not change significantly from one iteration to the next.

Recall that we created the GLM for multiple resolutions during training. When optimizing the ASM, we start at the lowest resolution so that the gray-level profiles in the test image cover a large area in the test image. This concept can be seen by noting the area covered by the profiles in Fig. 3.4. It appears that the profiles get longer at lower resolution, but in reality the profiles stay the same length (number of pixels), and the pixels are getting larger. Of course, the low-resolution image leads to a very rough estimate of the LP position, but this rough estimate is refined by using it as the initial guess for the next, higher-resolution search. This search process continues all the way up to the native resolution of the test image. Two examples showing the final result of the ASM segmentation are shown in Fig. 3.6.

3.2 Strengths and Shortcomings of ASM

There are several powerful features of ASM that make it advantageous for certain segmentation tasks and also provide many opportunities for research in the field. Of all of the deformable model techniques described to this point, none are better suited than ASM for applications in which substantial a priori information is available for incorporation into the segmentation process. This built-in a priori knowledge of the object's shape and boundary characteristics acquired from the training data set allows ASM to segment

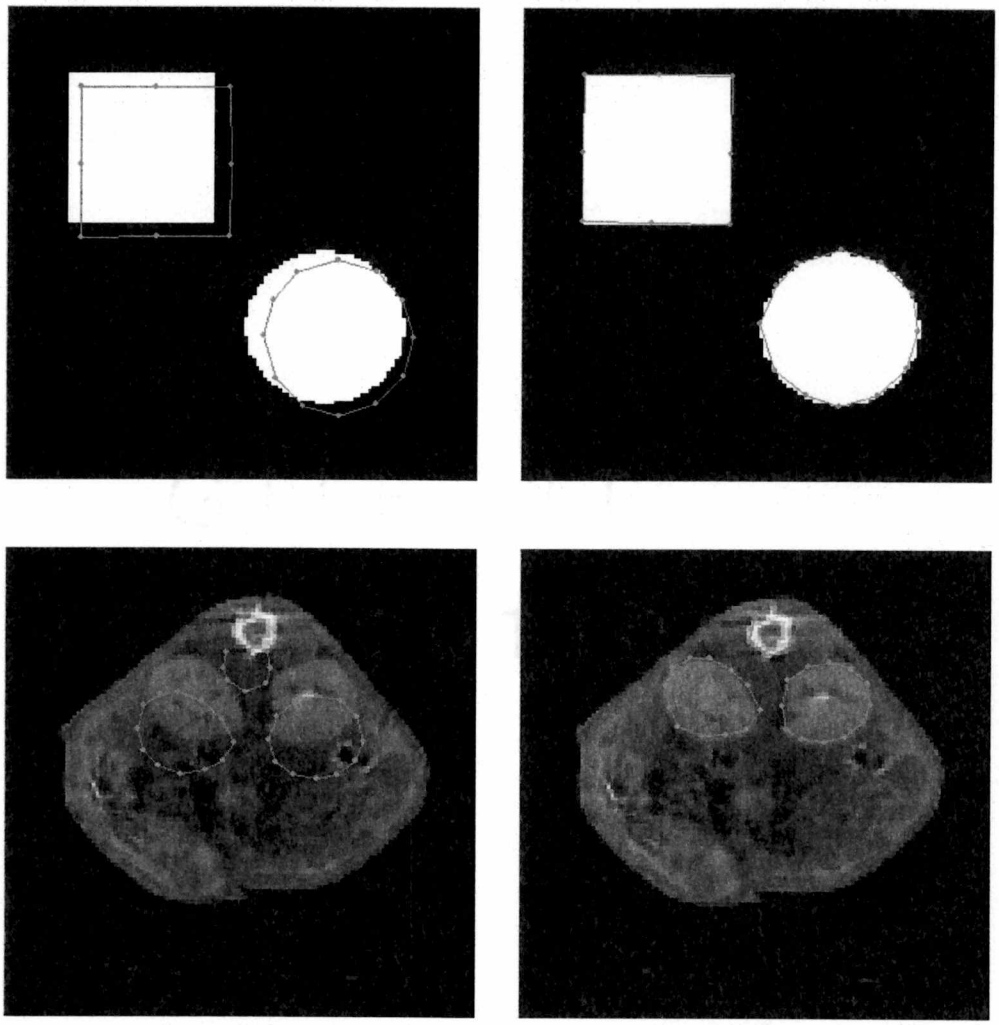


Fig. 3.6 Result of using ASM to segment shapes. A synthetic image and an X-ray CT image of a mouse are shown. The images on the left show the initial ASM position, those on the right show the final segmentation result.

objects with very faint and/or missing boundaries with remarkable accuracy. In addition, the ability to simultaneously optimize ASM to segment multiple objects with open and/or closed boundaries provides a useful flexibility in one's approach to a particular image analysis problem. In the examples to be shown later, some image objects are included when training ASM simply to provide a context for other, more difficult-to-segment objects.

Although ASM is a powerful tool for image segmentation applications, there are a tremendous number of opportunities for research to improve its robustness, accuracy, and applicability to a larger class of problems. For example, the GLM component of the ASM currently uses only edge information in the image to adjust the boundary LPs. This presents opportunities to include other information describing the characteristics of interior objects and the object's context (background). GLM also can get trapped in local minima, in part because it does not consider any prior shape knowledge (local or global) during optimization. In addition, the SM and GLM components of ASM are completely disconnected in that they do not share any useful information with one another during the contour deformation process. An example of this is the truncation of the global shape parameters in $\boldsymbol{b}$ to force-fit them into the limits imposed by the training set with no regard to the underlying image data. In many cases, this leads to poor segmentation results. And finally, the optimization strategy employed by ASM is not rigorous in that it lacks some key components necessary for a truly automatic analysis technique. For example, currently there is not a confidence measure that gives the user an idea of how well the segmentation performed. These shortcomings are addressed in Chap. 4.

3.3 An Improved ASM

Some of the shortcomings of ASM can be addressed by making practical improvements to the existing ASM technique. This section describes the implementation of the ASM algorithm and these practical ASM enhancements. Several changes that I made to the SM and GLM components led to substantial improvements in performance for which some examples are shown. The practical improvements described in this chapter are important because they are carried forward into the new statistical shape model derived in Chap. 4. Further information on these enhancements can be found in [43] and [44].

3.3.1 ASM Implementation

I implemented the published ASM technique using the MATLAB programming language. This implementation was invaluable in that it required a thorough understanding of the entire algorithm. The process of implementation also helped reveal the problematic aspects of the ASM approach, which generated a collection of ideas for improvement. This ASM implementation has been, and will continue to be, the platform for my research into statistical-based deformable models.

3.3.2 Resolution-Dependent Shape-Model

The first change in the ASM approach was to the shape constraint procedure used to control the final ASM boundary shape. Recall that the condition given by Eq. 3.12 is applied to the final shape parameter vector, $\boldsymbol{b}$, to constrain shapes that vary significantly from those in the training set. The challenge is to select a value of s that is small enough to keep the final shape within the bounds defined by the training set, yet large enough to allow the ASM to deform to fit the semi-rigid object in the image.

Typically, the SM shape constraint parameter, s, is held constant for each resolution of the multiresolution search. It was discovered that this can lead to undesirable segmentation results, especially in low-resolution images. Because the pixels are so large in low-resolution images, an error in the GLM fit of only a pixel or two can result in a large shape variation. As an example, see the images in the left-hand column of Fig. 3.7. The GLM has a difficult time with the low-resolution image because of some competing edges below the right kidney. Note how the shape is deformed (right kidney larger and lower than the left), and this error propagates through the mid- and full-resolution images. In the final result (bottom left, Fig. 3.7) ASM has missed the right kidney as well as the spine.

To prevent this significant shape deformation at low resolution, a resolution-dependent scaling term, r, was added to Eq. 3.12 as follows

$$-rs\sqrt{\lambda_i} \le b_i \le rs\sqrt{\lambda_i},\ \forall i = 1, \ldots, t, \qquad 3.15$$

where r is equal to the image resolution (e.g., 0.25, 0.5, or 1.0 for Fig. 3.7). This tightens the constraints on the shape at low resolution and leads to the results shown in the right-hand column of Fig. 3.7. This same approach has also been applied with successful results to the ellipsoid constraint given in Eq. 3.13 as

$$\sum_{i=1}^{t} \frac{b_i^2}{\lambda_i} \le rM_t. \qquad 3.16$$

Because of the inherent instability of GLM at low resolution, a more strict shape constraint (box or ellipsoid constraint) leads to better overall segmentation results.

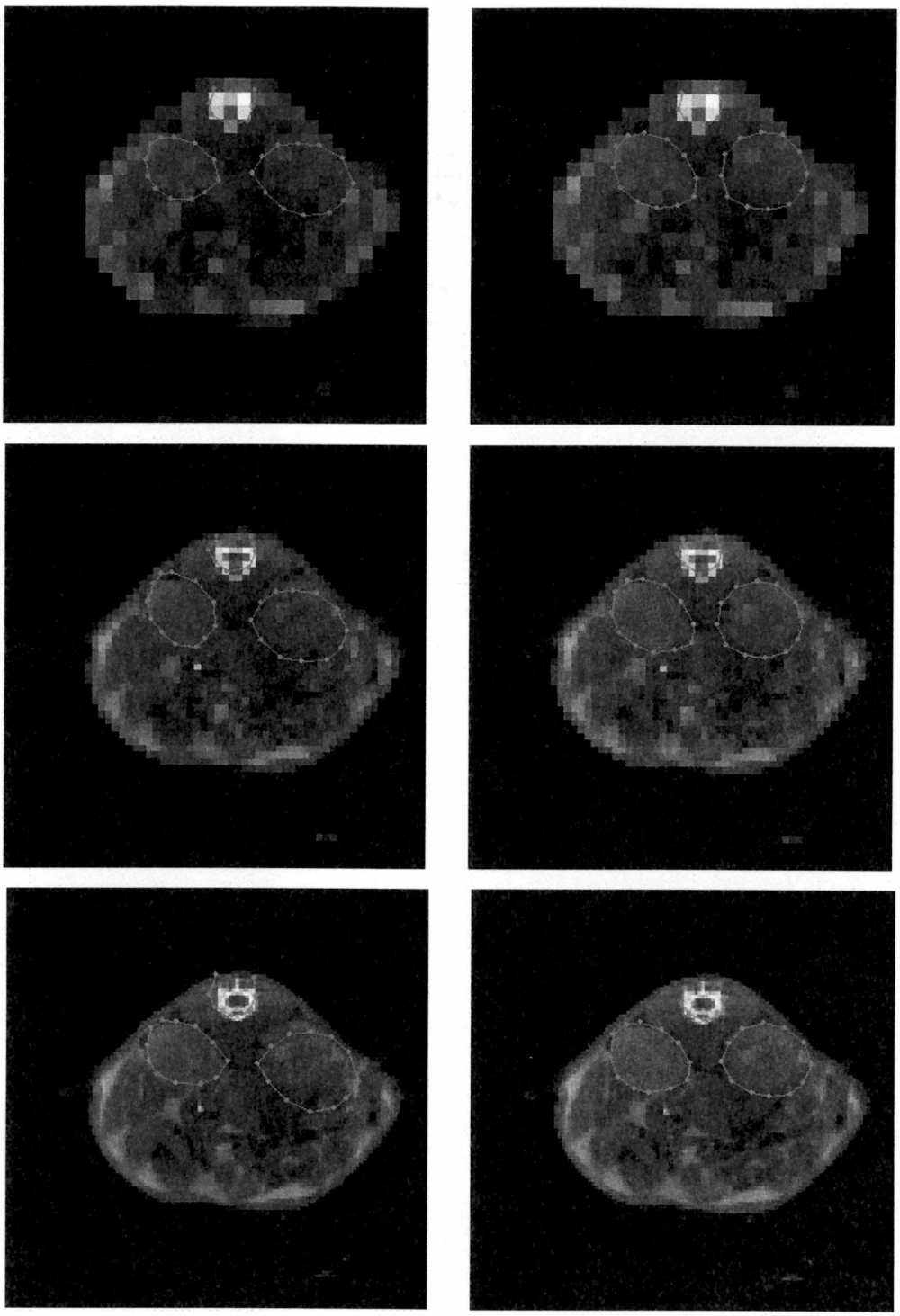


Fig. 3.7 Example of multiresolution ASM search. Results shown with standard, resolution-independent shape constraint (constant s) (left column) and with resolution-dependent shape constraint (varying s) (right column). Images are quarter resolution (top), half resolution (middle), and full resolution (bottom).

3.3.3 Improved GLM Objective Function

I improved the GLM objective function in two ways. First, a new objective function was developed to achieve better numerical stability. Second, both gray-level derivative (edge) information and absolute gray-level intensity information were incorporated into this new form of the objective function. The details are described in this section.

Numerically Stable GLM

The objective function presented to optimize the GLM at each LP (see Eq. 3.7) works well except when the gray-level profiles for a given LP do not vary much across all of the images within the image training set. The reason for this lies in the covariance matrix term, $\boldsymbol{S}_d$. If any of the values within the gray-level profiles do not vary, then the corresponding variance term(s) (diagonal elements of $\boldsymbol{S}_d$) will approach zero. This makes the covariance matrix singular, and inverting it during the evaluation of Eq. 3.7 is an ill-conditioned problem. This was first discovered during tests of the implemented ASM on a set of synthetic images with consistent gray-level profiles across all images in the training set. The algorithm used to invert the covariance matrix was reporting large condition numbers indicative of a very nearly singular matrix.

To solve this numerical problem, I adopted a new approach that resulted in a more stable objective function. With the same technique used to build the shape component of the ASM (see Sec. 3.1 for details), a PCA was performed on the set of gray-level profiles for each LP across all images in the training set. This was followed by the construction of the transformation matrix, Φ_d, whose columns are the t_g eigenvectors corresponding to the t_g largest eigenvalues of the covariance matrix, $\boldsymbol{S}_d$. Now when a new image is being ana-

lyzed, the gray-level derivative profile, $\boldsymbol{g}_d$, for each LP is transformed into the PCA-derived subspace using Eq. 3.17:

$$\boldsymbol{g}_d^{pca} = \Phi_d(\boldsymbol{g}_d - \bar{\boldsymbol{g}}_d), \tag{3.17}$$

where $\bar{\boldsymbol{g}}_d$ is the mean gray-level derivative profile for the current LP across all of the training images. The new objective function for the GLM to be minimized then becomes

$$f(\boldsymbol{g}_d) = \boldsymbol{g}_d^{pca,} \boldsymbol{D}_d \boldsymbol{g}_d^{pca} + \frac{\left|\boldsymbol{g}_d - \bar{\boldsymbol{g}}_d\right|^2 - \left|\boldsymbol{g}_d^{pca}\right|^2}{\sigma_{res}}, \tag{3.18}$$

where $\boldsymbol{D}_d$ is a t-dimensional diagonal matrix whose diagonal elements are λ_i^{-1}, $\forall i = 1, \ldots, t_g$, and σ_{res} is the variance of the residual given by

$$\sigma_{res} = \frac{\lambda_{t_g}}{2n_g}, \tag{3.19}$$

where n_g is the number of elements in the gray-level profile, $\boldsymbol{g}_d$. The improved objective function given by Eq. 3.18 is well-conditioned and attains its minimum at acceptable candidate LP positions.[1]

Incorporation of Absolute Intensity into GLM

Recall that GLM is created by measuring intensity gradient profiles through each LP normal to the object boundary. The derivative of these intensity profiles is taken before creating GLM so that only intensity changes, or edges, in the image are used as key features during GLM optimization. It became clear through the initial testing of ASM that some benefit could be derived by including some absolute intensity information in GLM.

1. The PCA-based GLM objective function that was developed included only the first term of Eq. 3.18, and the second term (the residual) is credited to Dr. T. Cootes.

In a variety of segmentation tasks the absolute intensity of the object is consistent and can be characterized, and this information can be incorporated into the search process.

For example, note the segmentation result in Fig. 3.8 (an enlarged version of the result shown in Fig. 3.7). The left kidney segmentation is nearly perfect, but the boundary around the right kidney has included some additional structure at the right and bottom-right. When optimizing GLM, the relatively weak edge of the kidney was overpowered by the strength of some spurious edges between the collections of high-intensity contrast agent (white spots) and the lower-intensity abdominal structures. It is clear from looking at the images in the training set that kidney boundaries should not enclose high-intensity groups of pixels, but because GLM was using only edge information, this knowledge about absolute intensity values was not encapsulated in the model.

To provide a means of encompassing this knowledge into GLM, an additional profile is captured during GLM training. This second profile is also composed of sampled intensity values along the normal line to the LP, but differs from the original profile in two important ways. First the absolute intensities (not the derivative values) are used when building the GLM; second, the image is sampled only on the *inside* of the object boundary. The reason for this is that in many segmentation applications, including the ones presented here, the object has a more predictable appearance than the background. Hence, the gray-level appearance of the object (excluding the background) can be effectively modeled by sampling only the interior of the object's boundary. Cootes, Edwards, and Taylor have published work [23] on an active appearance model (AAM) that incorporates absolute gray-level intensities of objects in a quite different manner. AAM actually uses the

Fig. 3.8 ASM segmentation result showing inclusion of contrast agent. Other abdominal structure is also shown within the right kidney boundary.

values of *all* pixel intensity values within an object during training, not just profiles. Also, AAM uses only gray-level information, not a combination of gray-level and derivative information as presented here.

The GLM objective function, $f(\boldsymbol{g}_d)$, was modified to include this new information as follows:

$$f(\boldsymbol{g}_d, \boldsymbol{g}_i) = \alpha f_d(\boldsymbol{g}_d) + \beta f_i(\boldsymbol{g}_i), \tag{3.20}$$

where α and β are constants (range 0 to 1) that determine the relative contribution of the derivative and intensity model terms; $\boldsymbol{g}_i$ is the new absolute intensity profile; f_d is the same as the objective function, f, in Eq. 3.18; and f_i is similarly defined as

$$f_i(\boldsymbol{g}_i) = \boldsymbol{g}_i^{pca\prime} \boldsymbol{D}_i \boldsymbol{g}_i^{pca} + \frac{\left|\boldsymbol{g}_i - \bar{\boldsymbol{g}}_i\right|^2 - \left|\boldsymbol{g}_i^{pca}\right|^2}{\sigma_{res}}. \tag{3.21}$$

To determine the effect of this new GLM objective function, it was used to segment both synthetic and real-world images. A set of synthetic images was created by placing a white square and an ellipse on a black background as shown at the top of Fig. 3.9. A training set of eight synthetic images was created where each image in the set contained the same two shapes with small affine transformations individually applied to each one. LPs were manually placed on the borders of the square and ellipse in each of the eight training images, and the ASM was trained. A test image was created that had the same two shapes, but a black band was included at the interior edge of the ellipse (see middle image in Fig. 3.9) to introduce competing edges. The ASM was then used to segment the square and ellipse using only the derivative-based GLM ($\alpha = 1$, $\beta = 0$). This generated the result shown in the middle of Fig. 3.9. Note that the square segmentation looks reasonable, but the elliptical ASM boundary has locked partially onto the exterior edge and partially onto the interior edge of the ellipse. If, however, the intensity-based GLM is also incorporated into the GLM objective function ($\alpha = 1$, $\beta = 1$), the ASM locks onto the interior edge only. A couple of points do not quite reach the edge because the shape component of the ASM prevents the boundary from getting too small. The same type of effect can be seen in the kidney-image result shown in Fig. 3.10 as compared to the derivative-based GLM result in Fig. 3.8.

3.4 Segmentation Results Using Enhanced ASM

This section presents the results of training and applying the improved ASM to a set of mouse kidney images. To build a training set, a collection of 16 axial CT images of mouse

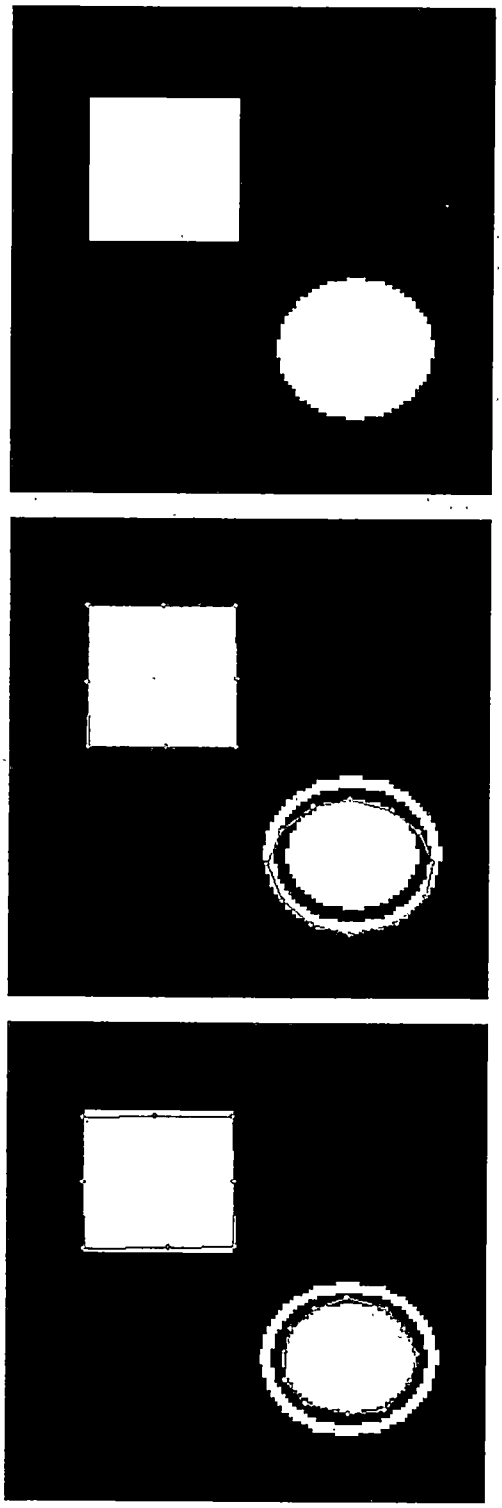


Fig. 3.9 Example of ASM segmentation. Shown are an image in training set (top), result of ASM segmentation on a new image using standard gradient-based GLM (middle), and result of ASM segmentation using the combination gradient- and intensity-based GLM.

Fig. 3.10 ASM segmentation using the derivative and intensity GLM. The image is an X-ray CT image of mouse kidneys.

kidneys were captured using a micro-CT scanner at Oak Ridge National Laboratory [78]. The kidneys and spine were then manually segmented in each of the 16 images by placing a collection of LPs on the spine and kidney borders (see Fig. 3.2). ASM was trained and then applied to each of the 16 images to segment both kidneys and the spine of the mouse. The accuracy of the segmentation was analyzed using a hold-one-out approach. Some results of the segmentation are shown in Fig. 3.11 and Fig. 3.12

Overall, the ASM segmentation results for the kidneys and spine was fairly good. One must note, however, a couple of problems. First, the ASM parameters were finely tuned before the results shown here were achieved. Some of the parameters are quite sensitive and, if slightly changed, lead to poorer result for some of the images. Second, even with the finely tuned ASM, a few results are suboptimal. In particular, note the result on subject

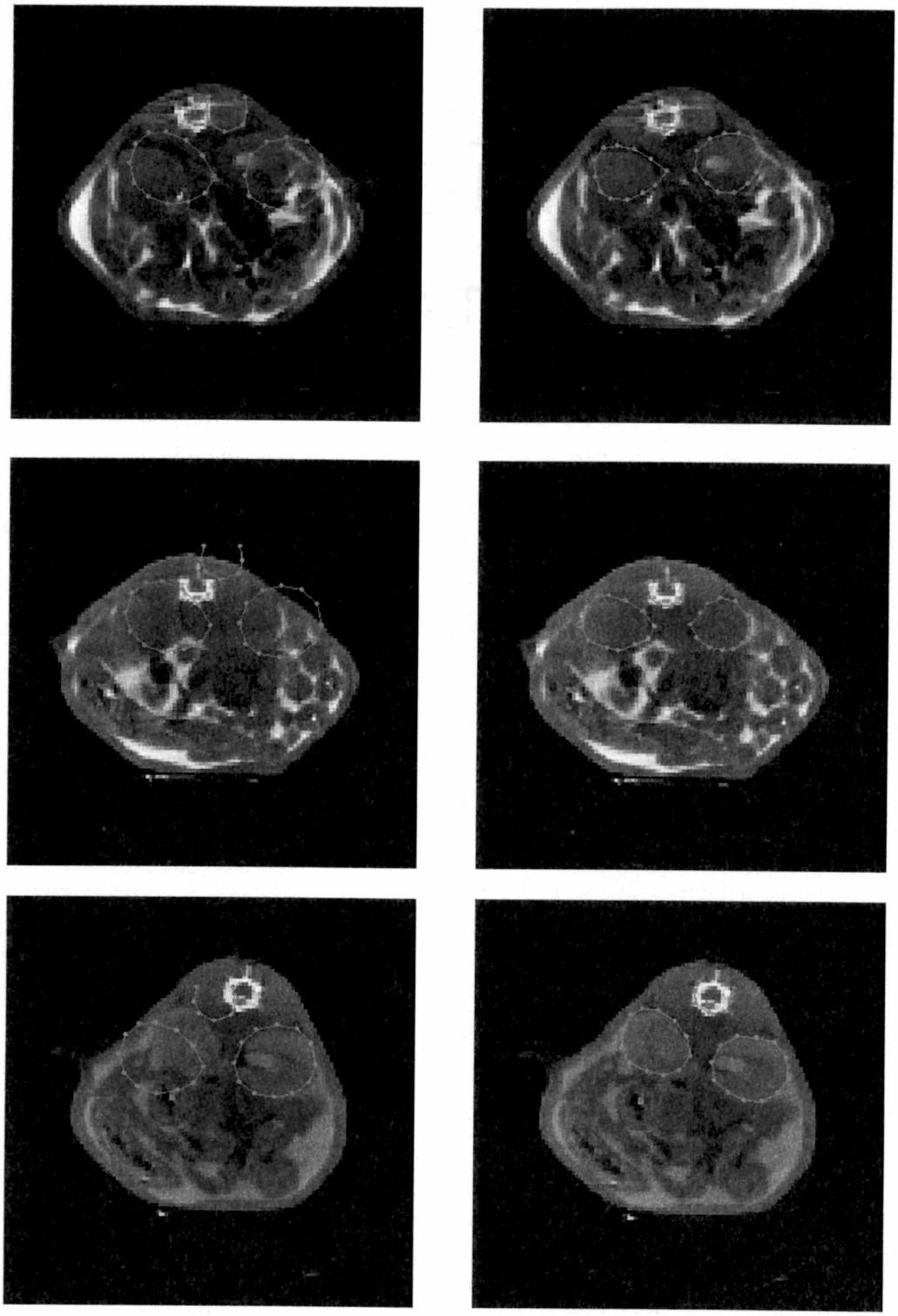


Fig. 3.11 ASM results on three mouse kidney CT images. From top to bottom these are subjects #2, #5, and #8. The images on the left are the initial ASM position, and the images on the right show the final position of the ASM after convergence.

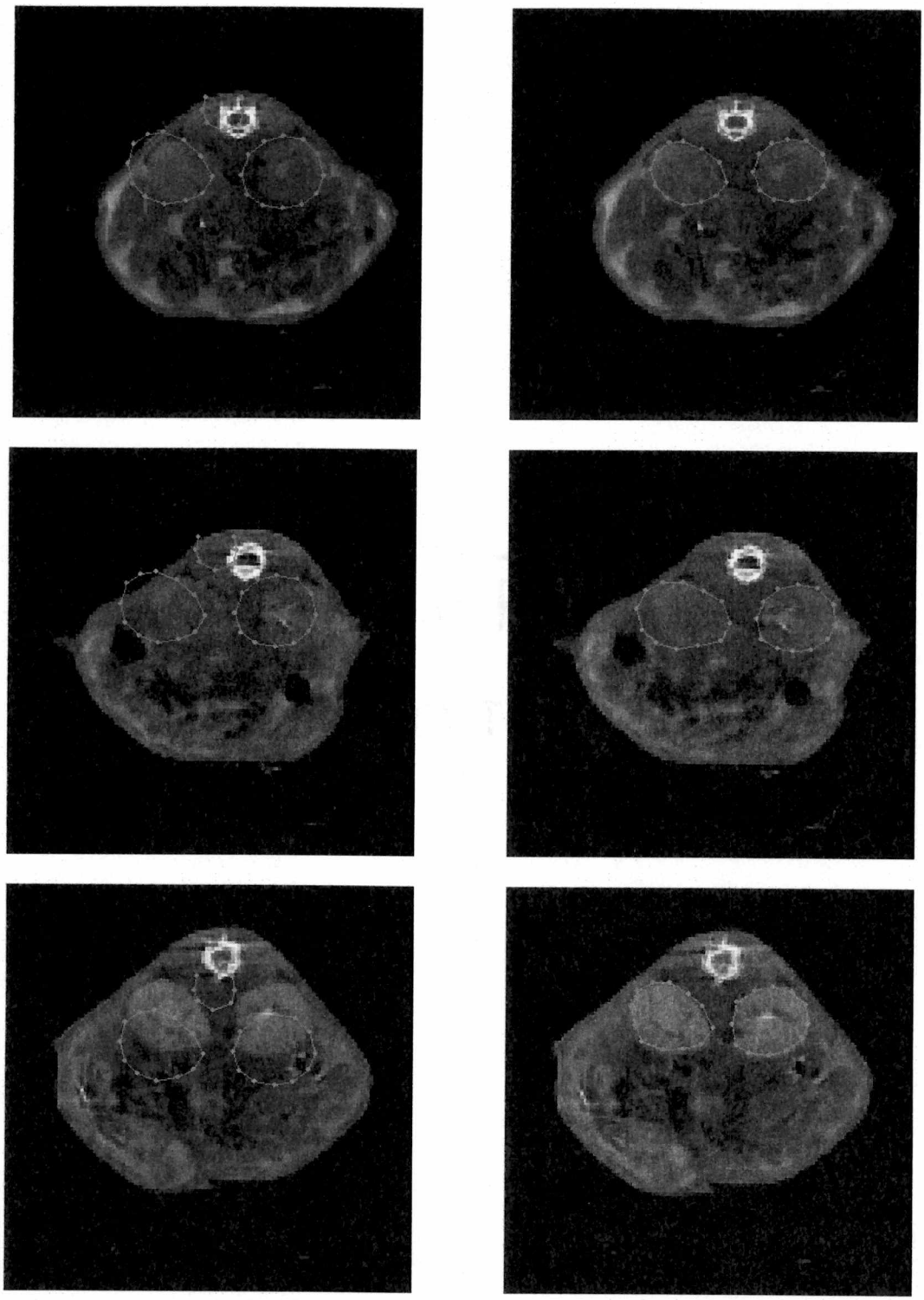


Fig. 3.12 ASM results on three mouse kidney CT images. From top to bottom these are subjects #9, #13, and #16. The images on the left are the initial ASM position, and the images on the right show the final position of the ASM after convergence.

#13. The right kidney is well-segmented, but the left kidney boundary has missed the left edge of the kidney by quite a few pixels. A potentially larger problem is that there are currently no measures in the ASM algorithm that provide an indication of the quality of the final result of the ASM segmentation. In a truly automated ASM analysis system, it is imperative to have some type of "confidence" measure for the segmentation results.

3.5 The Need for a New Statistical Shape Model

The remaining problems with ASM motivated the development of a new statistical-based deformable shape model. The goal of developing this new algorithm was achieved through the accomplishment of the three objectives described below. The description is followed by a brief introduction to the research approach employed to meet these objectives.

3.5.1 Objective 1: Combining the SM and GLM Optimization Processes

As noted earlier, the published ASM approach employs a GLM optimization process that uses only the local gray-level gradient profiles from the training set to determine LP movements. The presence of spurious edge information can create multiple, erroneous matches (due to multiple local minima in the objective function) during the GLM search. If the GLM optimization process considers both gradient profiles *and* shape characteristics when determining LP movements, a more optimal decision may be possible in the presence of multiple edges.

The following example clearly demonstrates this problem with the existing ASM optimization scheme. When the GLM objective function for each individual LP is being evaluated, the location corresponding to the absolute minimum of that objective function is

chosen as the new location for the LP. There may be several local minima that indicate a variety of possible locations to move the LP, but only the absolute minimum is chosen. To illustrate why this might be problematic, see the plot of the GLM objective function for a single LP (#17) shown in Fig. 3.13. The data were acquired during iteration #5 of an ASM optimization on mouse subject #13. This LP happens to be the left-most LP on the left kidney of the mouse. The absolute minimum of this objective function occurs at profile position 1, but there is another local minimum at profile position 4. Fig. 3.14(a) shows the original LP position (red dot) and the updated LP position (yellow dot) if the absolute minimum (profile position 1) is chosen. The GLM objective function plot for LP #17 looks very similar over several subsequent iterations of ASM optimization as well (i.e., it has two local minima separated by several pixels). If the absolute minimum is chosen at every iteration of the ASM fit, the final segmentation result is that shown in Fig. 3.15. Note that the left edge of the left kidney is quite faint and was missed by the ASM boundary.

As an experiment, the ASM optimization procedure was forced to select not the absolute minimum, but the second local minimum for LP #17 at profile position 4. The effect of this at a single iteration of ASM optimization can be seen in Fig. 3.14(b). This was similarly done in subsequent iterations of the ASM optimization as well and resulted in the final segmentation shown in Fig. 3.16. Obviously this result is an improvement over that in Fig. 3.15 and was achieved by manual selection of a different local minimum for a single LP. The question then becomes how to automatically determine which local minimum is the correct one for any given LP. An answer to the question appears to lie in the devel-

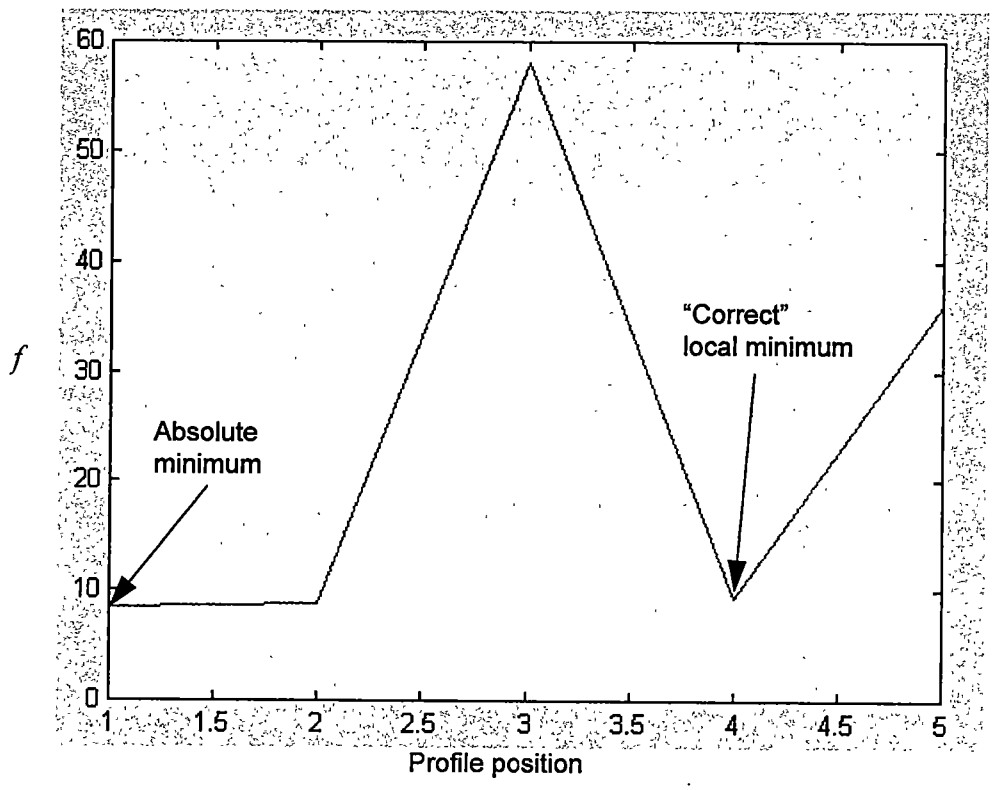
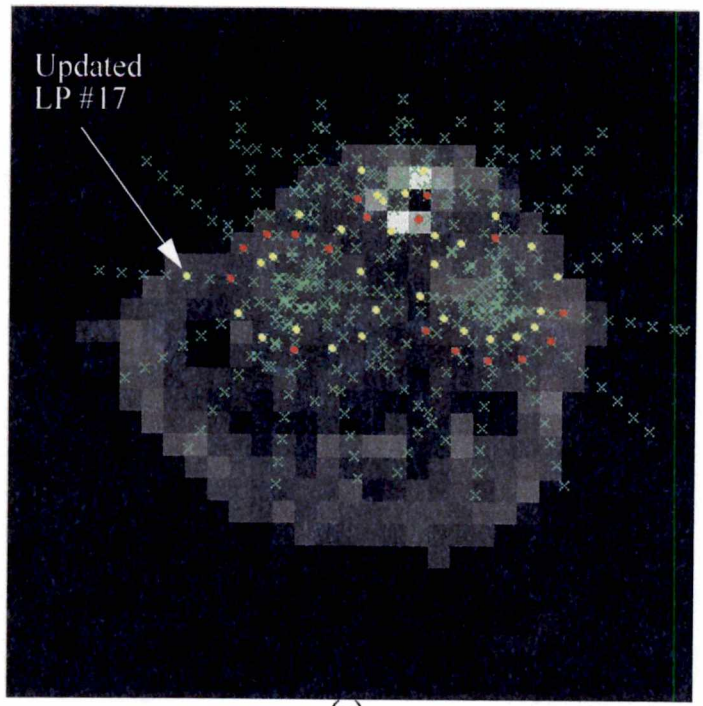
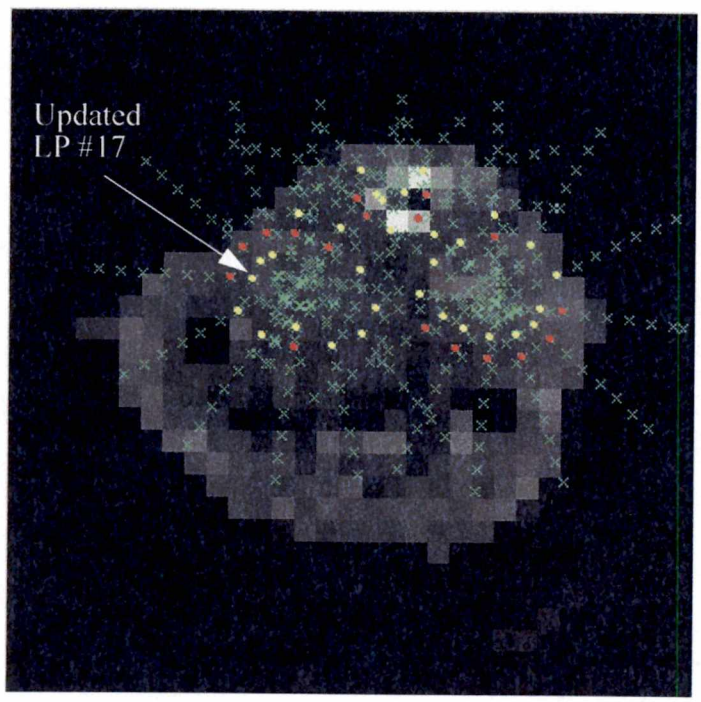


Fig. 3.13 Plot of the GLM objective function. This plot shows subject #13, LP #17, iteration #5. Note that there are two local minima at profile position 1 and profile position 4. The absolute minimum is at position 1, but better segmentation results are achieved using the local minimum at profile position 4.



(a)



(b)

Fig. 3.14 Results of GLM fit before (a) and after (b) correction of LP 17. The green 'x's indicate the profile sample positions for all LPs, the red dots are the previous LP positions, and the yellow dots are the new candidate LP positions suggested by the GLM. Note that LP #17 [left-most yellow dot in (a)] has been pushed in to the right by 3 pixels.

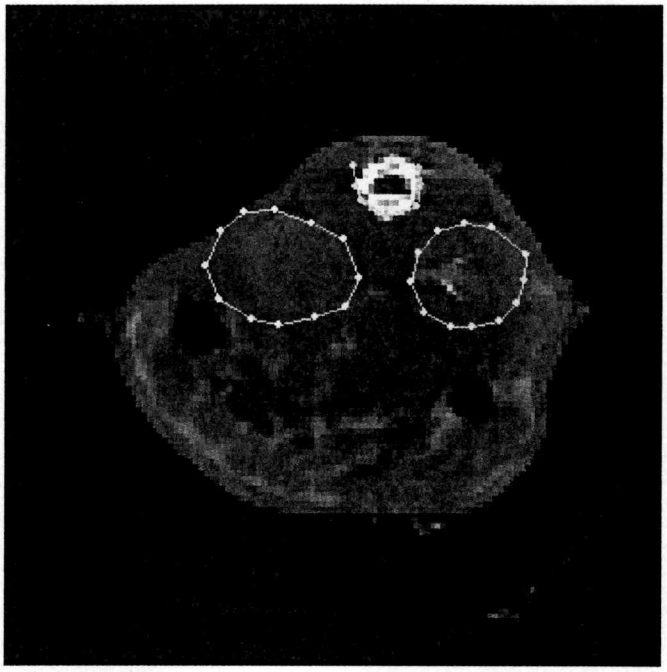


Fig. 3.15 Poor ASM result due to ambiguity in GLM fit at individual LPs.

opment of a new boundary optimization approach that considers the SM and GLM components simultaneously.

3.5.2 Objective 2: Employing a Continuous Penalty for Shape Variation

Recall that in the current ASM optimization scheme, if any of the shape modes varies outside of the limits dictated by the training set, a discontinuous adjustment is made to the out-of-bounds mode to move it back within the bounds defined by Eq. 3.12. This discontinuous penalty will affect the overall shape and cause movement of all LPs in the boundary. Smaller corrections to the shape that are made on a regular basis would avoid this step-wise disruption of the overall shape. Another goal of this research was to employ a continuous penalty for shape variations of the boundary as dictated by the training set.

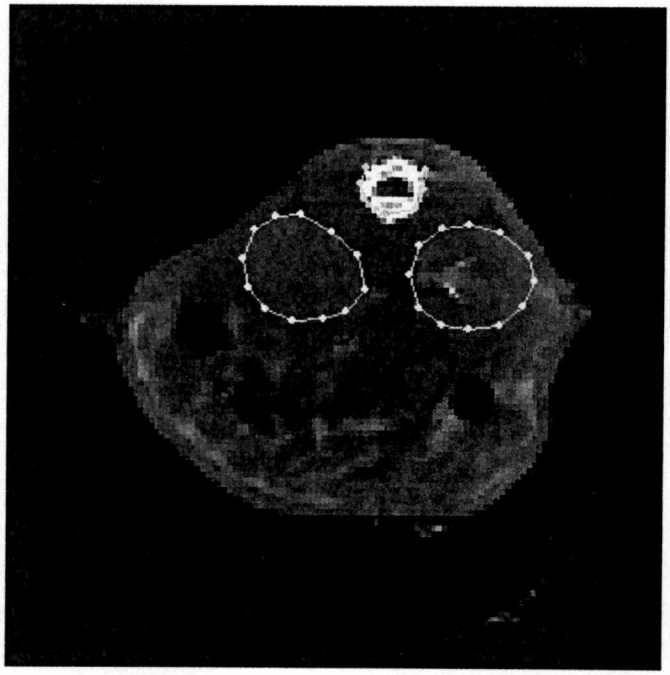


Fig. 3.16 ASM segmentation of subject 13 after adjustment of LP 17.

3.5.3 Objective 3: Including Local Shape Characteristics in Optimization

In published descriptions of the ASM training procedure, authors commonly describe the LP labelling process in two steps. First, "critical" LPs are placed on key features of the object such as corners and high curvature features. Second, additional "interpolation" LPs are distributed along the boundary between the critical LPs. Although this distinction is made between the two types of LPs during manual boundary placement, both sets of LPs are treated identically throughout the remainder of the training and optimization processes. Hence, the important local shape information the user is attempting to preserve around the critical LPs is de-emphasized when all LPs (both critical and interpolated) are lumped into a single global model of shape, the GSM. The introduction of local shape

models (LSMs) to preserve the local shape information around each user-defined critical LP was a goal of this research.

3.5.4 Objective 4: Developing a Segmentation Confidence Metric

The lack of interaction between the GLM and the SM creates another unfortunate situation in that no confidence measure is available that approximates how well the ASM segmentation algorithm has performed on any given image. In fully automated applications, the lack of a confidence metric renders the ASM technique useless. The reformulation of the optimization scheme will provide a means to generate a performance confidence metric that will allow the use of this new statistical-based deformable model in a fully automated application. Chap. 4 presents the development of a new statistical shape model that addresses this and the other three previously mentioned objectives.

CHAPTER 4

DEVELOPMENT OF A PROBABILISTIC FRAMEWORK FOR MODEL-BASED SEGMENTATION

This chapter presents the development of a new probabilistic deformable shape model based on compound Bayes decision theory [31]. This new model-based technique formulates the segmentation process as a maximum a posteriori probability (MAP) problem in which candidate image regions are evaluated as to whether they are likely (in a probabilistic sense) to lie along the ideal segmentation boundary. The prior probabilities are modeled on the basis of prior knowledge of both boundary shape/location and gray-level appearance near the boundary as determined from a training set of manually segmented images. A compound Bayesian objective function is formulated and then optimized to find the optimal segmentation boundary for the object of interest. This new approach, referred to as a probabilistic shape and appearance model (PSAM), is general enough to accommodate a variety of segmentation approaches but was formulated with the intention of accommodating the image measurement techniques described for ASM. Boundary finding research presented in [59], [70], [87], [89], and [96] is related to this work in that a probabilistic framework is formulated, but the compound Bayesian approach developed here results in a more general formulation into which these previous approaches fit as special cases.

This chapter develops the theory behind the formulation of this general probabilistic technique, and presents the derivation and optimization of an objective function based on

object shape and gray-level appearance that allows application of this new PSAM algorithm to objects within new images.

4.1 A New Probabilistic Shape and Appearance Model

As described in Chap. 3, some important practical enhancements were made to the existing ASM approach: (1) resolution dependence was incorporated into the shape model, (2) the numerical stability of the GLM was improved, and (3) additional intensity information was incorporated into the GLM. Despite these improvements, additional, more fundamental changes to ASM are needed to make it a more useful tool. In summary, the goals for this new PSAM are to

1. reformulate the statistical model objective function so that it considers both the gray-level and shape information simultaneously during contour deformation,
2. employ a continuous penalty for shape variations of the boundary as dictated by the training set,
3. formulate models of local shape characteristics to provide better control of relative local movement around critical LPs, and
4. develop a segmentation confidence metric.

The formulation of a probabilistic objective function that can be optimized to achieve these goals is now presented.

4.1.1 Compound Bayesian Approach

Whenever a priori information is available for an application and can be used to formulate probabilistic functions, a Bayesian approach to solving the problem is a reasonable consideration. The PSAM approach takes the shape and gray-level training data, formulates the boundary-finding problem as probabilistic one, and then incorporates this information into an a posteriori probability objective function. To meet the aforementioned goals, this objective function must contain terms dependent on global shape, local shape, and local gray-scale profile information. Once formulated, this function is optimized to find the boundary that corresponds to the MAP.

Before the writing of the objective function, some new variables must be introduced. First, we will define the boundary we are searching for as an (x, y) coordinate matrix, $\boldsymbol{S}_v$, of size $N \times 2$, where N is the number of LPs needed to represent the boundary. (See Chap. 3 for the discussion of an LP-based boundary.) Each row of $\boldsymbol{S}_v$ corresponds to the (x, y) coordinate of a boundary point as follows:

$$\boldsymbol{S}_v = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdots & \cdots \\ x_N & y_N \end{bmatrix} \tag{4.1}$$

The subscript v is used to foreshadow the relationship of the boundary with a combined shape-pose parameter vector, v, to be described later in this chapter. Defining the boundary this way accommodates some flexibility in how the boundary is represented. The points in $\boldsymbol{S}_v$ are typically LPs that lie directly on the boundary (as used in PSAM), but in addition, other (x, y) coordinate locations could be included that lie in image regions

that are on the interior or exterior of the object boundary, for example. These points could be used to capture other object characteristics that are located at off-boundary positions.

Next, we will define our feature vector matrix, $\boldsymbol{G}$, to be used later in the compound formulation of Bayes rule. In this formulation, the feature vectors (rows of $\boldsymbol{G}$) contain features extracted from the neighborhood of each LP on the boundary. If the length of each feature vector is m, then the size of $\boldsymbol{G}$ will be $N \times m$. These feature vectors may contain any image information that is relevant to the given application, such as gray-level intensities in the neighborhood of each pixel, local texture measurements, or gray-level gradient information (as used in ASM and PSAM). We can write $\boldsymbol{G}$ as

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \cdots \\ \boldsymbol{g}_N \end{bmatrix}, \tag{4.2}$$

where

$$\boldsymbol{g}_j = h(\boldsymbol{i}_j), \ \forall j = 1, \ldots, N. \tag{4.3}$$

In Eq. 4.3, $\boldsymbol{i}_j$ is a vector containing raw gray-level values sampled from the neighborhood of the (x_j, y_j) LP in the test image, and h is an operator (e.g., a gradient) that transforms these raw gray-level values into the feature vector, $\boldsymbol{g}_j$. Now that we have defined our parameters, we can formulate the probabilistic objective function.

Bayes decision theory is a fundamental approach to solving probabilistic problems. A typical application of Bayesian theory is calculation of the probability of an event given that a different, but related, event has already occurred (and can be measured). Compound Bayes decision theory is an extension of this idea that allows the prediction of the proba-

bility of multiple simultaneous events, given that several different, but related, simultaneous events have already occurred (and can be simultaneously measured). The compound form of Bayes rule [31] can be written as

$$P(\omega|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\omega)P(\omega)}{p(\boldsymbol{X})}, \qquad 4.4$$

where ω is the vector of the events one wishes to predict. The rows of the matrix $\boldsymbol{X}$ represent the measured events (e.g., feature vectors) that have already occurred. $P(\omega)$ is the a priori probability for the vector, ω; $p(\boldsymbol{X}|\omega)$ is the conditional probability density function for $\boldsymbol{X}$, given the event vector, ω; and $p(\boldsymbol{X})$ is the prior probability density of the measured event matrix, $\boldsymbol{X}$. Finally, $P(\omega|\boldsymbol{X})$ is the a posteriori probability of ω.

In the PSAM boundary-finding application, the goal is to maximize the a posteriori probability of the boundary $\boldsymbol{S}_v$, given the measured image features in $\boldsymbol{G}$. Using the compound version of Bayes rule in Eq. 4.4, we can write the a posteriori probability expression for the boundary, $\boldsymbol{S}_v$, given a collection of feature vectors, $\boldsymbol{G}$, as

$$P(\boldsymbol{S}_v|\boldsymbol{G}) = \frac{p(\boldsymbol{G}|\boldsymbol{S}_v)P(\boldsymbol{S}_v)}{p(\boldsymbol{G})}, \qquad 4.5$$

where $P(\boldsymbol{S}_v)$ is the prior probability of a boundary instance, $\boldsymbol{S}_v$; $p(\boldsymbol{G}|\boldsymbol{S}_v)$ is the conditional density of $\boldsymbol{G}$ given the boundary instance, $\boldsymbol{S}_v$; and $p(\boldsymbol{G})$ is the prior probability density for $\boldsymbol{G}$. The goal is then to optimize Eq. 4.5 by searching over all possible values of $\boldsymbol{S}_v$ to find the one, $\boldsymbol{S}_v^*$, that corresponds to the MAP value as given by

$$P(\boldsymbol{S}_v^*|\boldsymbol{G}) = max_{\boldsymbol{S}_v}\frac{p(\boldsymbol{G}|\boldsymbol{S}_v)P(\boldsymbol{S}_v)}{p(\boldsymbol{G})}. \qquad 4.6$$

Finding $\boldsymbol{S}_v$* can be further simplified without losing the generality that we wish to maintain for a variety of boundary-finding applications. First, because the logarithm is a monotonically increasing function, optimizing the logarithm of a function yields the same result as optimizing the original function. Therefore, the objective function [we will now call it $J(\boldsymbol{S}_v)$] can be written as

$$J(\boldsymbol{S}_v) = \ln p(\boldsymbol{G}|\boldsymbol{S}_v) + \ln P(\boldsymbol{S}_v), \tag{4.7}$$

where we have dropped the term $-\ln p(\boldsymbol{G})$ since it is independent of $\boldsymbol{S}_v$ and is therefore a constant.

Depending on the application at hand, the term $\ln p(\boldsymbol{G}|\boldsymbol{S}_v)$ may be difficult to calculate. The level of difficulty is contingent in large part on the level of independence that can be assumed for the current problem. If independence is assumed between the feature vectors in $\boldsymbol{G}$, and if it is also assumed that each feature vector, $\boldsymbol{g}_j$, is dependent only on it corresponding location, (x_j, y_j) (i.e., j^{th} row of $\boldsymbol{S}_v$), then the conditional density of $\boldsymbol{G}$ given $\boldsymbol{S}_v$ can be rewritten as [31]

$$p(\boldsymbol{G}|\boldsymbol{S}_v) = \prod_{j=1}^{N} p(\boldsymbol{g}_j|S_{v_{j,1}}, S_{v_{j,2}}) = \prod_{j=1}^{N} p(\boldsymbol{g}_j|x_j, y_j). \tag{4.8}$$

Plugging this expression back into Eq. 4.7, we can write

$$J(\boldsymbol{S}_v) = \sum_{j=1}^{N} \ln p(\boldsymbol{g}_j|S_{v_{j,1}}, S_{v_{j,2}}) + \ln P(\boldsymbol{S}_v) = \sum_{j=1}^{N} \ln p(\boldsymbol{g}_j|x_j, y_j) + \ln P(\boldsymbol{S}_v). \tag{4.9}$$

It is useful to note that the first term is the "data-driven" term of the objective function in that it depends on image characteristics (external energy term), while the second term is

more "model-driven" in that it is independent of the image, depending only on prior distributions of boundary shape and location (internal energy term).

Note how this formulation can potentially accommodate all of the goals outlined at the start of the chapter. First, this objective function allows simultaneous optimization with respect to image-derived gray-scale information (first term) and shape information (second term). Secondly, because this objective function is based on a probabilistic framework, we can interpret its value as a measure of how well the final boundary fits the distribution approximated by those contained in the training set. This measure can broken down into two pieces: (1) the first term measures how well the final gray-level information matches that which was extracted from the training data, and (2) the second term measures how well the overall shape and location of the boundary matches that which was extracted from the training set. The third and final goal of the formulation is to allow for incorporation of local shape characteristics. The term that can accommodate this is the prior boundary instance probability, $P(\boldsymbol{S}_v)$, but the details of this will be explained later in the chapter.

Optimizing $J(\boldsymbol{S}_v)$ over all possible boundary vectors, $\boldsymbol{S}_v$, can be a daunting task for several reasons. First, depending on the number of LPs used to represent the boundary, $\boldsymbol{S}_v$ could be a very long vector, and finding the maximum of J with respect to each of the $2N$ elements of $\boldsymbol{S}_v$ can be computationally demanding. As will be detailed later, PCA can be used to reduce the dimensionality of $\boldsymbol{S}_v$ to resolve this problem. The boundary $\boldsymbol{S}_v$ can be approximated in the PCA subspace as a vector, $\boldsymbol{v}$, with fewer dimensions. Hence, optimizing the objective function in the PCA sub-space with respect to the more compact vector,

$\boldsymbol{v}$, is a simpler task. If we substitute the new boundary representation, $\boldsymbol{v}$, into Eq. 4.7, the objective function then takes the form

$$J(\boldsymbol{v}) = \ln p(\boldsymbol{G}|\boldsymbol{v}) + \ln P(\boldsymbol{v}), \tag{4.10}$$

where

$$p(\boldsymbol{G}|\boldsymbol{v}) = \prod_{j=1}^{N} p(\boldsymbol{g}_j|f_{x_j}(\boldsymbol{v}), f_{y_j}(\boldsymbol{v})) = \prod_{j=1}^{N} p(\boldsymbol{g}_j|x_j, y_j), \tag{4.11}$$

or

$$\ln p(\boldsymbol{G}|\boldsymbol{v}) = \sum_{j=1}^{N} \ln p(\boldsymbol{g}_j|f_{x_j}(\boldsymbol{v}), f_{y_j}(\boldsymbol{v})) = \sum_{j=1}^{N} \ln p(\boldsymbol{g}_j|x_j, y_j), \tag{4.12}$$

where $f_{x_j}(\boldsymbol{v})$ and $f_{y_j}(\boldsymbol{v})$ are functions that map $\boldsymbol{v}$ from the PCA subspace back into the image (x, y)-coordinate space.

To use this objective function in practice, we must know all of the individual conditional probability densities, $p(\boldsymbol{g}_j|x_j, y_j)$, $\forall j = 1, \ldots, N$. In the PSAM approach, these conditional densities can be straightforwardly measured from the training data. Finally, the prior probability of the PCA boundary vector, $P(\boldsymbol{v})$, must also be known. Once again, this can be estimated from the training data, as will be seen later in this chapter.

As mentioned at the beginning of the chapter, previous research into probabilistic boundary-finding fits into this formulation. For example, Kervrann and Heitz [59], Staib and Duncan [87], and Wang and Staib [96] all use the values of individual pixels in an edge-detected image as the external energy force to attract the boundary. Relating their approach to the formulation presented in Eq. 4.9, the value of the pixel at each LP in the edge-detected image can be interpreted as the image-derived feature vector (containing

only a single element). Also, in previous research ([87],[96]) the conditional probability densities are interpreted as correlations of a noise-corrupted template (i.e., the image under test) with an ideal template (the current template estimate in an iterative update scheme). In this research, the conditional densities are not modeled, as is proposed here, using a corresponding collection of boundary-derived feature vectors within the training set.

4.1.2 Employing the Compound Bayesian Approach for PSAM

We will now explore the specific application of this compound Bayesian formulation to boundary-finding using the image measurement concepts employed by ASM (i.e., using LPs and normal gray-level profiles). Recall that the boundary is defined as a set of LPs that are positioned at pixel locations within the image that correspond to the boundary of the object of interest. The number (N) and density of LPs along the object's boundary (and therefore the number of image-derived feature vectors) is determined by the N manually placed LPs in the training set.

Before describing the details of the implementation, we will summarize how this approach accomplishes the goals laid out at the start of the chapter. We now have a reformulation of the shape model objective function (Eq. 4.10) that simultaneously considers information both on shape (the internal energy term) and local gradient profile (the external energy term) during the deformation process. Next, we need to formulate the probabilistic models for the individual components of this objective function: the shape model and the gray-level model.

Global Shape Model

The formulation of the shape model term, $P(\boldsymbol{v})$ (see Eq. 4.10), is presented here. Because this term incorporates the influence of all LPs along the boundary, it is called the global shape model term. Before specifying a model for $P(\boldsymbol{v})$, we must first describe the process of forming the PCA-based global shape vector, $\boldsymbol{v}$.

Although it was convenient to represent the boundary as a matrix, $\boldsymbol{S}_{\boldsymbol{v}}$, during the compound Bayesian description of the objective function, it is simpler for implementation to represent the entire collection of (x, y) coordinates as one long vector, $\boldsymbol{p}$, as was originally described in Sec. 3.1.1. Reiterating the description in Sec. 3.1.1 for the sake for clarity, let the manually selected boundary for the i^{th} image in an M-image training set be represented by a collection of landmark points (LPs), $\boldsymbol{p}_{train_i}$ as

$$\boldsymbol{p}_{train_i} = \left[x_{i_1}\ x_{i_2}\ \dots\ x_{i_N}\ y_{i_1}\ y_{i_2}\ \dots\ y_{i_N}\right],\ \forall i = 1, \dots, M, \tag{4.13}$$

where the N LPs that make up the boundary for the i^{th} sample are defined by the coordinate pairs

$$(x_{i_j}, y_{i_j}); \forall j = 1, \dots, N. \tag{4.14}$$

Recall that the alignment of the sets of LPs to a common coordinate frame is done via Procrustes analysis [45] to form the aligned sets of LPs:

$$\hat{\boldsymbol{p}}_{train_i} = \left[\hat{x}_{i_1}\ \hat{x}_{i_2}\ \dots\ \hat{x}_{i_N}\ \hat{y}_{i_1}\ \hat{y}_{i_2}\ \dots\ \hat{y}_{i_N}\right],\ \forall i = 1, \dots, M. \tag{4.15}$$

Using this training data set of aligned LPs, $\hat{\boldsymbol{p}}_{train_i}$, we can create a GSM using PCA [31]. Although previous work in this area almost always employs the PCA transform, its purpose has not been clearly described. Performing PCA on the training data samples serves three primary purposes. In order of least to most significant, these are as follows:

1. **Reduces computation:** Applying PCA to a set of vectors will allow a reduction in the dimensionality of the problem, if desired. This reduction is accomplished by disregarding the insignificant modes of variation (i.e., those modes with relatively small eigenvalues). Reducing the dimensionality of the problem reduces computation time.
2. **Provides numerical stability for the model:** Multivariate Gaussian distributions are commonly used to model multidimensional data samples. Evaluating the equation for the Gaussian distribution always involves taking the inverse of the covariance matrix for the sample data used to specify the model. This is evident when observing the form of a probabilistic multivariate Gaussian density function, $p(\boldsymbol{p})$, for an LP-based boundary vector

$$\boldsymbol{p} = \left[x_1\ x_2\ \ldots\ x_N\ y_1\ y_2\ \ldots\ y_N\right]. \tag{4.16}$$

The Gaussian distribution is parameterized using the covariance, $\boldsymbol{C}$, and mean, $\bar{\boldsymbol{p}}$, calculated from the training samples as follows:

$$P(\boldsymbol{p}) = \frac{1}{(2\pi)^N |\boldsymbol{C}|^{1/2}} \exp\left\{-\frac{1}{2}[(\boldsymbol{p}-\bar{\boldsymbol{p}})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{p}-\bar{\boldsymbol{p}})]\right\}. \tag{4.17}$$

As discussed in more detail in Sec. 3.3.3, if any single element of the multidimensional data set does not vary across all of the samples, the resulting covariance matrix, $\boldsymbol{C}$, will be singular, making its inversion impossible. PCA provides a straightforward means for removing those (near) zero modes of variation that cause the covariance matrix to be (nearly) singular so that numerical problems can be avoided during the covariance matrix inversion.

3. **Places hard limits on solution variation:** Once the PCA transform has been deter-

mined and the insignificant modes of variation have been excluded, any vector to which this PCA transformation is applied will automatically lose those excluded modes of variation. For example, if an input vector, $\boldsymbol{v}$, is PCA-transformed, and the result is then inverse PCA-transformed to form the vector, $\tilde{\boldsymbol{v}}$, the output vector, $\tilde{\boldsymbol{v}}$, will be the same as the input vector, $\boldsymbol{v}$, minus the excluded modes of variation. Thus, the process of zeroing insignificant modes of variation has the desired effect of forcing new PCA-transformed vectors to look like those in the training set of data samples.

To formulate the PCA transform, we start by calculating the mean, $\bar{\boldsymbol{p}}$, and the covariance, $\boldsymbol{C}$, of the $\hat{\boldsymbol{p}}_{train_i}$, $\forall i = 1, \ldots, M$ as in Eq. 3.2 and Eq. 3.3, respectively. Next, we calculate the eigenvectors, ϕ_k, $\forall k = 1, \ldots, 2N$, and the eigenvalues, λ_k, $\forall k = 1, \ldots, 2N$, of $\boldsymbol{C}$. Recall from Sec. 3.1.3 that we choose the t_s largest eigenvalues (out of $2N$ total) that capture 98% of the total variation in the training set. One can now calculate the forward PCA transform, $\boldsymbol{b}$, for a new aligned global shape, $\hat{\boldsymbol{p}}$,[1] as

$$\boldsymbol{b} = \Phi^{\mathrm{T}}(\hat{\boldsymbol{p}} - \bar{\boldsymbol{p}}), \tag{4.18}$$

where Φ is a matrix whose columns are the eigenvectors, ϕ_k, $\forall k = 1, \ldots, t_s$, of the covariance matrix, $\boldsymbol{C}$, for the collection of $\hat{\boldsymbol{p}}_{train_i}$, $\forall i = 1, \ldots, M$. This shape parameter vector, $\boldsymbol{b}$, is an approximate representation of the LP-based global shape boundary, $\hat{\boldsymbol{p}}$. It is only approximate because we discarded the last $2N - t_s$ eigenvectors where the statistical variation from the shapes the training set was insignificant.

1. Note that we have dropped the i subscript here because we are looking at new global shapes that are not part of the $\forall i = 1, \ldots, M$ training vectors.

The inverse-PCA transform can be written to generate a new shape, $\hat{\boldsymbol{p}}$, from the PCA shape vector, $\boldsymbol{b}$, as follows

$$\hat{\boldsymbol{p}} = \bar{\boldsymbol{p}} + \Phi \boldsymbol{b} = \bar{\boldsymbol{p}} + \begin{bmatrix} \phi_1 & \dots & \phi_{t_s} \end{bmatrix} \begin{bmatrix} b_1 \\ \dots \\ b_{t_s} \end{bmatrix}. \tag{4.19}$$

The elements of $\boldsymbol{b}$ govern the amount of shape variation from the mean, $\bar{\boldsymbol{p}}$, along each eigenvector, ϕ_k.

It is important to understand that the PCA transform in Eq. 4.18 operates on shape vectors that are *aligned* to the mean shape, $\bar{\boldsymbol{p}}$, and to recall that $\bar{\boldsymbol{p}}$ was created by first aligning all of the original training vectors, $\boldsymbol{p}_{train_i}$, $\forall i = 1, \dots, M$. Consequently, the inverse PCA transform in Eq. 4.19 will generate a shape vector, $\hat{\boldsymbol{p}}$, that is aligned to $\bar{\boldsymbol{p}}$, but in the end we wish for it to be aligned with the true boundary of the object of interest within the image. We must also therefore determine pose parameters that will transform the current boundary estimate, $\hat{\boldsymbol{p}}$, so that it is aligned with the true boundary of the object of interest in the test image. Let us introduce a vector, z, that contains these pose parameters,

$$z = \begin{bmatrix} s & \theta & T_x & T_y \end{bmatrix}^{\mathrm{T}}, \tag{4.20}$$

where s is scale, θ is the rotation, and T_x, T_y are the x- and y-translations, respectively. We can then combine the global shape parameter vector, $\boldsymbol{b}$, and the pose parameter vector, z, to form the combined shape-pose parameter vector, $\boldsymbol{v}$, as

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{b} \\ z \end{bmatrix}, \tag{4.21}$$

where the length of z is $t_p = 4$ and the overall length of $\boldsymbol{v}$ is then $t = t_s + t_p$. So, given an aligned boundary, $\hat{\boldsymbol{p}}$, we then apply the appropriate pose parameters to translate, rotate, and scale the boundary to fit the object of interest within the image. This pose-corrected boundary, $\boldsymbol{p}$, is calculated using the following expressions:

$$\boldsymbol{p} = [\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}^{\mathrm{T}}] = H_{s,\theta,T_x,T_y}(\hat{\boldsymbol{p}}), \tag{4.22}$$

where H_{s,θ,T_x,T_y} is the pose transformation used to align the LPs to the object of interest as dictated by the pose parameters, s, θ, T_x, T_y. Application of H_{s,θ,T_x,T_y} to generate the individual elements (x_j, y_j) takes the following form:

$$x_j = s\{(\hat{p}_j)\cos\theta - (\hat{p}_{j+N})\sin\theta\} + T_x,$$

$$\forall j = 1, \ldots, N \tag{4.23}$$

and

$$y_j = s\{(\hat{p}_j)\sin\theta + (\hat{p}_{j+N})\cos\theta\} + T_y,$$

$$\forall j = 1, \ldots, N. \tag{4.24}$$

Note that $\hat{p}_j$ in Eq. 4.23 and Eq. 4.24 is found by applying the inverse PCA transform for each shape parameter contained in $\boldsymbol{v}$ as follows (as in Eq. 4.19):

$$\hat{p}_j = \bar{p}_j + \sum_{k=1}^{t_s} \Phi_{j,k} v_k, \ \forall j = 1, \ldots, 2N. \tag{4.25}$$

Now that we have gathered information on the statistical variation of the shape vectors across the training set, we can formulate a probability density function that describes this global shape variability in terms of the combined shape-pose parameter vector, $\boldsymbol{v}$. Choosing a computationally attractive and widely applicable model for varying data, we model

the probability of the combined shape-pose parameter vector, $\boldsymbol{v}$, using a multivariate Gaussian function as follows:

$$P(\boldsymbol{v}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{C}|^{1/2}} \exp\left\{-\frac{1}{2}[(\boldsymbol{v}-\bar{\boldsymbol{v}})^{\mathrm{T}}\boldsymbol{C}^{-1}(\boldsymbol{v}-\bar{\boldsymbol{v}})]\right\}, \tag{4.26}$$

where

$$\bar{\boldsymbol{v}} = \begin{bmatrix} \bar{\boldsymbol{b}} \\ \bar{\boldsymbol{z}} \end{bmatrix}, \tag{4.27}$$

and

$$\bar{\boldsymbol{z}} = \left[\bar{s}\ \bar{\theta}\ \overline{T_x}\ \overline{T_y}\right]^{\mathrm{T}}. \tag{4.28}$$

The average pose parameters, $\bar{s}$, $\bar{\theta}$, $\overline{T_x}$, and $\overline{T_y}$, are calculated by averaging the corresponding pose parameters gathered for each training vector, $\boldsymbol{p}_{train_i}$, $\forall i = 1, \ldots, M$ during the Procrustes alignment procedure. Also note that because PCA was applied to the shape vectors and the insignificant modes of variation were discarded, we can guarantee that $\boldsymbol{C}$ will not be singular.

We can substantially simplify Eq. 4.26 if we can show that the elements of $\boldsymbol{v}$ are independent. We know that the elements v_1 through v_{t_s} (i.e., the elements of the shape parameter vector, $\boldsymbol{b}$) are independent because we constructed them to be so. They are a result of the PCA transform given by Eq. 4.18. In addition, the elements v_{t_s+1} through v_t (i.e., $\bar{s}$, $\bar{\theta}$, $\overline{T_x}$, and $\overline{T_y}$, the parameters of the pose vector, $\boldsymbol{z}$) are also assumed to be independent because a change in any one of these pose parameters has no direct correlation on the change in value of the other three. This independence leads to a diagonal covariance

matrix, $\boldsymbol{C}$, where $C_{k,l} = \sigma_k; \forall k = l$ and $C_{k,l} = 0$, otherwise. Given that $\boldsymbol{C}$ is diagonal, Eq. 4.26 can be simplified to

$$P(\boldsymbol{v}) = \prod_{k=1}^{t} \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{(v_k - \bar{v}_k)^2}{2\sigma_k^2}\right\}, \qquad 4.29$$

where σ_k is the standard deviation of the element v_k. The standard deviations for the pose parameters (σ_t, σ_{t+1}, σ_{t+2}, σ_{t+3}) are calculated from the distribution of pose parameters measured during the Procrustes alignment of the training shapes. This completes the derivation of the GSM term, $P(\boldsymbol{v})$, in Eq. 4.10.

Gray Level Model

The shape prior probability, $P(\boldsymbol{v})$, has now been completely specified, so we will now address the formulation of the GLM conditional density term in Eq. 4.10, $p(\boldsymbol{G}|\boldsymbol{v})$. This term describes the probability of extracting a set of gray-level profiles, $\boldsymbol{G}$, from an image, given a boundary parameter vector, $\boldsymbol{v}$.

Recall that $\boldsymbol{g}_j$ is a vector of the gray-level values along a profile that passes through the j^{th} LP and is normal to the current boundary estimate. Because the location and angle of the gray-level profile, $\boldsymbol{g}_j$, is dependent on the LPs within the boundary, it is necessarily a function of the current shape-pose parameter vector, $\boldsymbol{v}$. The inverse PCA transform can be applied to $\boldsymbol{v}$ to obtain $\boldsymbol{p}$ using Eq. 4.19 and Eq. 4.22. Recall that the elements of $\boldsymbol{p}$ are the coordinate pair values of the LPs.

We can write the pixel intensity elements of the normal profile, $\boldsymbol{i}_j$, as

$$i_{j_k} = \boldsymbol{I}\left(x_j - \left(\frac{(N_g-1)}{2} - k + 1\right)\cos\alpha_j, y_j - \left(\frac{(N_g-1)}{2} - k + 1\right)\sin\alpha_j\right),$$

$$\forall k = 1, \ldots, N_g \text{ and } \forall j = 1, \ldots, N, \qquad 4.30$$

where $\boldsymbol{I}$ is the image under test, N_g is the number of gray-level samples in each profile, and α_j is the angle of the profile through the j^{th} LP normal to the boundary. As noted in Eq. 4.3, the final feature vectors are given as $\boldsymbol{g}_j = h(\boldsymbol{i}_j)$, where h is the gradient operator. The next step is to choose the form of the individual prior probability distribution functions, $p(\boldsymbol{g}_j|x_j, y_j)$, $\forall j = 1, \ldots, N$. A multivariate Gaussian distribution is assumed as a model for the profiles and can be expressed as

$$p(\boldsymbol{g}_j|x_j, y_j) = \frac{1}{(2\pi)^{N_g/2}|\boldsymbol{Q}_j|^{1/2}}\exp\left\{-\frac{1}{2}((\boldsymbol{g}_j - \bar{\boldsymbol{g}}_j)'\boldsymbol{Q}_j^{-1}(\boldsymbol{g}_j - \bar{\boldsymbol{g}}_j))\right\}, \qquad 4.31$$

where $\boldsymbol{Q}_j$ is the covariance matrix measured from the collection of gray-level profile vectors through the j^{th} LP across all images in the training set, and $\bar{\boldsymbol{g}}_j$, is the average gray-level profile measured through the j^{th} LP across all images in the training set. Like the covariance matrix $\boldsymbol{C}$ in the shape model, the covariance matrix $\boldsymbol{Q}$ can in many instances become singular, so that finding $\boldsymbol{Q}_j^{-1}$ is impossible. One effective way to avoid these singularities is to perform PCA on the collection of $\boldsymbol{g}_j$s for a given LP, and then exclude those modes of variation that are small. The PCA transform for the vector $\boldsymbol{g}_j$ is

$$\boldsymbol{d}_j = \Psi_j^{\mathrm{T}}(\boldsymbol{g}_j - \bar{\boldsymbol{g}}_j), \qquad 4.32$$

where the columns of the matrix Ψ_j are the eigenvectors $[\psi_{j_k}; \forall k = 1, \ldots, t_g(j)]$ corresponding to the $t_g(j)$ largest eigenvalues $[\lambda_{j_k}; \forall k = 1, \ldots, t_g(j)]$ of the covariance matrix

for the collection of $g_{train_{i,j}};\forall i = 1, \ldots, M$. The gray-level parameter vector, d_j, is an approximate representation of the gray-level profile, g_j. The PCA-transformed vector, d_j, is only approximate because we discarded the last $2N_g - t_g(j)$ eigenvectors where the variation from the gray-level profiles within the training set was insignificant.

The inverse-PCA transform can be written to generate a new gray-level profile, g_j, from the PCA gray-level profile, d_j, through the j^{th} LP as follows:

$$g_j = \bar{g}_j + \Psi_j d_j = \bar{g}_j + \begin{bmatrix} \psi_{j_1} & \cdots & \psi_{j_{t_g(j)}} \end{bmatrix} \begin{bmatrix} d_{j_1} \\ \cdots \\ d_{j_{t_g(j)}} \end{bmatrix}. \tag{4.33}$$

The elements of d_j govern the amount of variation from the mean gray-level profile, $\bar{g}_j$, along each eigenvector.

The PCA transform can be used to define an approximate expression for the exponential term in Eq. 4.31 as follows

$$\frac{-(g_j - \bar{g}_j)^{\mathrm{T}} Q_j^{-1} (g_j - \bar{g}_j)}{2} \approx -\frac{1}{2}\left\{ \sum_{k=1}^{t_g(j)} \frac{d_{j_k}^2}{\lambda_{j_k}} + \frac{|g_j - \bar{g}_j|^2 - |d_j|^2}{\sigma_g(j)} \right\}, \tag{4.34}$$

where $\sigma_g(j)$ is the variance of the GLM residual not captured by the PCA approximation of the Mahalanobis distance (the first term). This residual variance is approximated by

$$\sigma_g(j) \approx \frac{\lambda_{j_{t_g(j)}}}{2N_g}. \tag{4.35}$$

An expression has been developed that does not require the calculation of the potentially singular covariance matrix, Q_j^{-1}.

For mathematical convenience, we can simplify and rewrite the exponential term on the right-hand side of Eq. 4.34 as follows

$$-\frac{1}{2}\left\{\sum_{k=1}^{t_g(j)}\frac{d_{j_k}^2}{\lambda_{j_k}}+\frac{\left|g_j-\bar{g}_j\right|^2-\left|d_j\right|^2}{\sigma_{g_j}}\right\}=-\frac{1}{2}\dot{g}_j^{\mathrm{T}}K_{g_j}\dot{g}_j, \tag{4.36}$$

where

$$\dot{g}_j = g_j-\bar{g}_j, \tag{4.37}$$

$$K_{g_j} = \frac{1}{\sigma_{g_j}}(I-\Psi_j\Psi_j^{\mathrm{T}})+\Psi_j\Lambda_j\Psi_j^{\mathrm{T}}, \tag{4.38}$$

and

$$\Lambda_j = diag\left[\frac{1}{\lambda_{j_1}}\ \frac{1}{\lambda_{j_2}}\ \cdots\ \frac{1}{\lambda_{j_{t_g(j)}}}\ 0\ \cdots\ 0\right]. \tag{4.39}$$

Now Eq. 4.31 can be rewritten as

$$p(g_j|v) = \frac{1}{(2\pi)^{N_g}|Q_j|^{1/2}}\exp\left\{-\frac{1}{2}\dot{g}_j^{\mathrm{T}}K_j\dot{g}_j\right\}. \tag{4.40}$$

Substituting the individual gray-level profile probability back into the combined gray-level profile likelihood probability (Eq. 4.12), we can write

$$p(G|v) = \prod_{j=1}^{N}\frac{1}{(2\pi)^{N_g}|Q_j|^{1/2}}\exp\left\{-\frac{1}{2}\dot{g}_j^{\mathrm{T}}K_j\dot{g}_j\right\}. \tag{4.41}$$

This completes the derivation of the GLM term, $p(G|v)$, in Eq. 4.12.

Local Shape Model

In published descriptions of the ASM training procedure, authors commonly describe the LP labelling process in two steps. First, "critical" LPs are placed on key features of the object such as corners and other high-curvature features. Second, additional "interpolation" or "standard" LPs are manually distributed along the boundary between the critical LPs. Although this distinction is made between the two types of LPs during manual boundary placement, both types of LPs are treated identically throughout the remainder of the training and optimization processes. Hence, the important local shape information the user is attempting to preserve around the critical LPs is deemphasized when all LPs (both critical and interpolated) are lumped into a single global model of shape, the GSM. Local shape models (LSMs) are introduced to preserve the local shape information around each user-defined critical LP. A probabilistic term that depends on LSMs is introduced to the objective function in Eq. 4.12 to reward (add to) the objective function for boundaries whose local shapes closely match those local shapes defined in the training set. This new reward term is formulated as a joint probability of a collection of local shapes.

Adding reward and/or penalty terms to probabilistic objective functions is quite common in other applications such as iterative tomographic image reconstruction [36] to enforce some type of local constraint (e.g., smoothness). Although adding reward or penalty terms will in many cases be effective in practice, they are sometime difficult to justify theoretically. Adding a term to a probabilistic objective function mathematically implies that the term is independent of the other terms in the objective function. Although this is not always the case, positive results can sometimes still be obtained in practice.

The formulation of the LSMs is similar to the GSM formulation in that a vector of LPs, $\boldsymbol{l}_{train_{i,j}}$, which is a subset of the global-shape training LPs, $\boldsymbol{p}_{train_i}$, is identified in the neighborhood of the j^{th} critical LP during training. These training local shapes are formed by extracting a subset of LPs from the global shape training vector $\boldsymbol{p}_{train_i}$ (defined in Eq. 4.13) as follows

$$\boldsymbol{l}_{train_{i,j}} = [x_{i_{h(j)}}, \ldots, x_{i_{h(j)+N_l-1}}, y_{i_{h(j)}}, \ldots, y_{i_{h(j)+N_l-1}}]; \quad \forall i = 1, \ldots, M; \forall j = 1, \ldots, N_c \quad , \qquad 4.42$$

where M is the number of training images, N_c is the number of user-defined critical LPs, $h(j)$ defines the index of the first LP in $\boldsymbol{p}_{train_i}$ belonging to the j^{th} local shape, and N_l is the number of LPs in each local shape. Fig. 4.1 shows an example for $N_c = 2$ and $N_l = 3$.

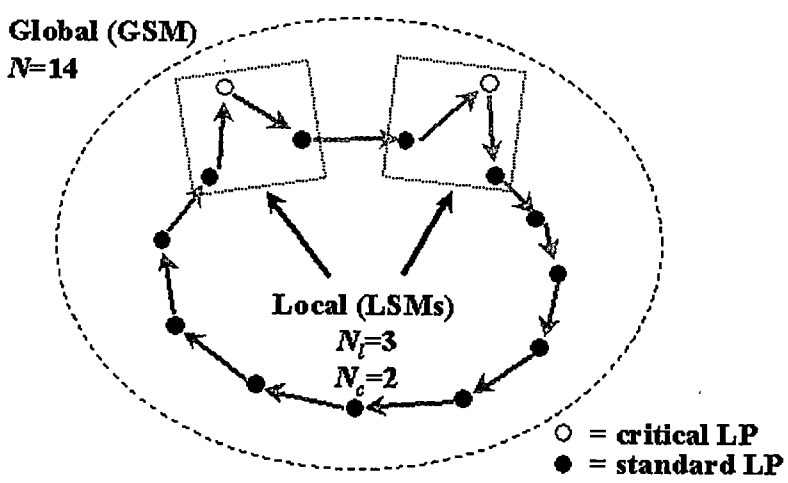


Fig. 4.1 Illustration of a shape boundary. The boundary model contains both a GSM and two LSMs around critical LPs.

Our goal now is to formulate the joint probability density for the local shapes to be used during optimization. Assuming independence of local shapes, we can form the overall local shape probability as the product of the individual LSM probabilities as

$$P(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_{N_c}) = \prod_{j=1}^{N_c} P(\hat{\boldsymbol{l}}_j), \tag{4.43}$$

where $\hat{\boldsymbol{l}}_j$ is an aligned local shape vector containing the coordinates of the local shape of interest. The process of formulating $\hat{\boldsymbol{l}}_j$ is described in detail later in the section, but note here that the elements of the local shape vectors are simply the (x, y) coordinates of the LP locations. Now if we take the log and add this joint probability as a reward term to the existing a posteriori objective function (Eq. 4.10), we can write the new objective function as

$$J(\boldsymbol{v}) = \ln p(\boldsymbol{G}|\boldsymbol{v}) + \ln P(\boldsymbol{v}) + \ln P(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_{N_c}). \tag{4.44}$$

A standard multivariate Gaussian distribution for the j^{th} aligned local shape, $\hat{\boldsymbol{l}}_j$, can be written as

$$P(\hat{\boldsymbol{l}}_j) = \frac{1}{(2\pi)^{N_c/2}|\boldsymbol{W}_j|^{1/2}} \exp\left\{-\frac{1}{2}((\hat{\boldsymbol{l}}_j - \bar{\boldsymbol{l}}_j)^{\mathrm{T}} \boldsymbol{W}_j^{-1} (\hat{\boldsymbol{l}}_j - \bar{\boldsymbol{l}}_j))\right\}, \tag{4.45}$$

where $\bar{\boldsymbol{l}}_j$ is the mean local shape across the training set and $\boldsymbol{W}_j$ is the covariance of the local shape vectors. Keeping the notation consistent with the GSM, recall that the aligned local shapes are defined as $\hat{\boldsymbol{l}}_{train_{i,j}}$. The mean local shape can be calculated as

$$\bar{\boldsymbol{l}}_j = \frac{1}{M} \sum_{i=1}^{M} \hat{\boldsymbol{l}}_{train_{i,j}}; \forall j = 1, \ldots, N_c, \tag{4.46}$$

and the covariance can be found using

$$W_j = \frac{1}{M-1}\sum_{i=1}^{M}(\hat{l}_{train_{i,j}} - \bar{l}_j)(\hat{l}_{train_{i,j}} - \bar{l}_j)^{\mathrm{T}}. \qquad 4.47$$

Note that we can write $\hat{l}_j$ as a function of v, since $\hat{l}_j$ is a subset of the current boundary estimate that is computed on the basis of the current value of the combined shape-pose vector, v.

We take a similar approach in establishing the GSM in that we will work with a PCA-transformed version of the local shapes. Recall that in the GSM formulation, we enumerated three reasons for using the PCA transform to model the data: (1) to reduce computation, (2) to provide numerical stability, and (3) to place hard limits on sample variation. For the local shape models, we use the PCA transform only for the second reason—that is, to provide numerical stability. The benefit of reduced computation is also achieved via dimensionality reduction, but the reduction is insignificant compared to the computation encountered during the entire model fitting process.

We encounter the same problem as we did with the GSM in that there is no guarantee that W_j will not be singular, so we will formulate the PCA transform to create a more numerically stable form of the distribution. To formulate the PCA transform, we must first align all of the local shapes and then calculate their mean and covariance. The local shapes across all training images for a given critical LP are first aligned via Procrustes analysis, as was described for the GSM in Chap. 3.

As we did in the GSM formulation we can perform PCA on the collection of local shape vectors, $\hat{l}_{train_{i,j}}$, and then write the forward PCA transform of a new aligned local

shape vector, $\hat{\boldsymbol{l}}_j$[1]. Just as it was in the case of the GSM, is important to understand here that the PCA transform in Eq. 4.57 operates only on shape vectors that are aligned to $\bar{\boldsymbol{l}}_j$ and to recall that $\bar{\boldsymbol{l}}_j$ was created by first aligning all of the training local shape vectors, $\hat{\boldsymbol{l}}_{train_{i,j}}$, by applying affine transformations to each training vector. Before the PCA transform can be applied to a new candidate local shape vector, $\boldsymbol{l}_j$, we must also determine pose parameters that align the local shape vector with the corresponding local shape mean, $\bar{\boldsymbol{l}}_j$. We can align it as follows:

1. First, recall that our ultimate goal is to formulate our PSAM with respect to the combined shape-pose parameter vector, $\boldsymbol{v}$, so we will need to define the current local shape vector in terms of the current estimate of $\boldsymbol{v}$. Note that $\boldsymbol{l}_j$ is a collection of (x, y) coordinates that is a subset of the current boundary vector, $\boldsymbol{p}$ (see Eq. 4.22), and can be written as

$$\boldsymbol{l}_j = \left[x_{h(j)_1}\ x_{h(j)_2}\ \cdots\ x_{h(j)_{N_l}}\ y_{h(j)_1}\ y_{h(j)_2}\ \cdots\ y_{h(j)_{N_l}}\right], \tag{4.48}$$

   where $\boldsymbol{h}(j)$ is a vector containing the indices of the LPs in $\boldsymbol{p}$ that belong to the j^{th} local shape vector, $\boldsymbol{l}_j$.

2. Center the new shape vector, $\boldsymbol{l}_j$, by subtracting the centroid of the local shape from

1. Once again, we have dropped the i subscript here because we are looking at new local shapes that are not part of the $i = 1, \ldots, M$ training vectors.

each coordinate as follows:

$$\boldsymbol{l}_{cen_{j_k}} = \boldsymbol{l}_{j_k} - \sum_{i=1}^{N_l} \frac{\boldsymbol{l}_{j_i}}{N_l}, \text{ for } k = 1, \ldots, N_l, \tag{4.49}$$

and

$$\boldsymbol{l}_{cen_{j_k}} = \boldsymbol{l}_{j_k} - \sum_{i=N_l+1}^{2N_l} \frac{\boldsymbol{l}_{j_i}}{N_l}, \text{ for } k = N_l + 1, \ldots, 2N_l. \tag{4.50}$$

3. Calculate the scale, $\tilde{s}_j$, and the rotation, γ_j, needed to align $\boldsymbol{l}_{cen_j}$ to $\bar{\boldsymbol{l}}_j$ as

$$\tilde{s}_j = \sqrt{\tilde{a}_j^{\,2} + \tilde{b}_j^{\,2}} \tag{4.51}$$

and

$$\gamma_j = \operatorname{atan}\frac{\tilde{b}_j}{\tilde{a}_j}, \tag{4.52}$$

where

$$\tilde{a}_j = \frac{\boldsymbol{l}_{cen_j} \bullet \bar{\boldsymbol{l}}_j}{\left|\boldsymbol{l}_{cen_j}\right|^2} = \frac{\sum_{i=1}^{2N_l} l_{cen_{j_i}} \bar{l}_{j_i}}{\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2} \tag{4.53}$$

and

$$\tilde{b}_j = \frac{\left[\sum_{i=1}^{N_l} l_{cen_{j_i}} \overline{l_{j_{N_l+i}}} - \sum_{i=1}^{N_l} l_{cen_{j_{N_l+i}}} \bar{l}_{j_i}\right]}{\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2}. \tag{4.54}$$

4. Apply the transformation to form the aligned local shape vector, $\hat{\boldsymbol{l}}_j$:

$$\hat{l}_{j_k} = \tilde{s}_j\left(l_{cen_{j_k}}\cos\gamma_j - l_{cen_{j_{k+N_l}}}\sin\gamma_j\right),\ \forall k = 1, \ldots, N_l \tag{4.55}$$

and

$$\hat{l}_{j_k} = \tilde{s}_j\left(l_{cen_{j_{k-N_l}}}\sin\gamma_j + l_{cen_{j_k}}\cos\gamma_j\right),\ \forall k = N_l + 1, \ldots, 2N_l. \tag{4.56}$$

Now that we have the aligned version of our local shape vector, $\hat{\boldsymbol{l}}_j$, we can formulate the forward PCA transform for that vector as

$$\boldsymbol{c}_j = \Omega_j^{\mathrm{T}}(\hat{\boldsymbol{l}}_j - \bar{\boldsymbol{l}}_j), \tag{4.57}$$

where the columns of the matrix Ω_j are the eigenvectors $[\omega_{j_k};\forall k = 1, \ldots, t_l(j)]$ corresponding to the $t_l(j)$ largest eigenvalues $[\lambda_{j_k};\forall k = 1, \ldots, t_l(j)]$ of the covariance matrix, $\boldsymbol{W}_j$, for the collection of $\hat{\boldsymbol{l}}_{train_{i,j}};\forall i = 1, \ldots, M$. Similar to global shape parameter vector, $\boldsymbol{b}$, in the GSM, this local shape parameter vector, $\boldsymbol{c}_j$, is an approximate representation of the aligned LP-based local shape, $\hat{\boldsymbol{l}}_j$. It is only approximate because we discarded the last $2N_l - t_l(j)$ eigenvectors where the variation from the training set was insignificant.

The inverse-PCA transform can be written to generate a new local shape, $\hat{\boldsymbol{l}}_j$, from the PCA local shape vector, $\boldsymbol{c}_j$, as follows:

$$\hat{\boldsymbol{l}}_j = \bar{\boldsymbol{l}}_j + \Omega_j\boldsymbol{c}_j = \bar{\boldsymbol{l}}_j + \begin{bmatrix}\omega_{j_1} & \ldots & \omega_{j_{t_l(j)}}\end{bmatrix}\begin{bmatrix}c_{j_1} \\ \ldots \\ c_{j_{t_l(j)}}\end{bmatrix}. \tag{4.58}$$

The elements of $\boldsymbol{c}_j$ govern the amount of shape variation from the mean, $\bar{\boldsymbol{l}}_j$, along each eigenvector.

As with the GLM development in the previous section, an alternative expression can be used to approximate the exponential term in Eq. 4.45 as follows:

$$-\frac{1}{2}[(\hat{\boldsymbol{l}}_j-\bar{\boldsymbol{l}}_j)^{\mathrm{T}}\boldsymbol{W}_j^{-1}(\hat{\boldsymbol{l}}_j-\bar{\boldsymbol{l}}_j)]\approx-\frac{1}{2}\left\{\sum_{k=1}^{t_l(j)}\frac{c_{j_k}^2}{\lambda_{j_k}}+\frac{\left|\hat{\boldsymbol{l}}_j-\bar{\boldsymbol{l}}_j\right|^2-\left|\boldsymbol{c}_j\right|^2}{\sigma_{l_j}}\right\}, \tag{4.59}$$

where σ_{l_j} is the variance of the LSM residual not captured by the PCA approximation of the squared Mahalanobis distance (the first term). It is used to normalize, in a Mahalanobis sense, the difference of the distances from $\hat{\boldsymbol{l}}_j$ to $\bar{\boldsymbol{l}}_j$ (the $\left|\hat{\boldsymbol{l}}_j-\bar{\boldsymbol{l}}_j\right|^2$ term) and from $\boldsymbol{c}_j$ to $\bar{\boldsymbol{c}}_j$ (the $\left|\boldsymbol{c}_j\right|^2$ term, since $\bar{\boldsymbol{c}}_j$ is zero via PCA construction). This residual variance is approximated by taking a fraction of the last (smallest) eigenvalue used in the PCA representation as indicated by the following expression:

$$\sigma_{l_j}\approx\frac{\lambda_{j_{t_l(j)}}}{2N_l}. \tag{4.60}$$

The goal has been achieved in that an expression has been developed that does not require the calculation of the potentially singular $\boldsymbol{W}_j^{-1}$.

Once again, for mathematical convenience, the exponential term on the right-hand side of Eq. 4.59 can be simplified as follows:

$$-\frac{1}{2}\left\{\sum_{k=1}^{t_l(j)}\frac{c_{j_k}^2}{\lambda_{j_k}}+\frac{\left|\hat{\boldsymbol{l}}_j-\bar{\boldsymbol{l}}_j\right|^2-\left|\boldsymbol{c}_j\right|^2}{\sigma_{l_j}}\right\}=-\frac{1}{2}\dot{\boldsymbol{l}}_j^{\mathrm{T}}\boldsymbol{K}_{l_j}\dot{\boldsymbol{l}}_j, \tag{4.61}$$

where

$$\dot{\boldsymbol{l}}_j = \hat{\boldsymbol{l}}_j - \bar{\boldsymbol{l}}_j, \tag{4.62}$$

$$\boldsymbol{K}_{l_j} = \frac{1}{\sigma_{l_j}}(\boldsymbol{I} - \Omega_j \Omega_j^{\mathrm{T}}) + \Omega_j \Lambda_j \Omega_j^{\mathrm{T}}, \tag{4.63}$$

and

$$\Lambda_j = diag\left[\frac{1}{\lambda_{j_1}} \frac{1}{\lambda_{j_2}} \cdots \frac{1}{\lambda_{j_{t_l(v)}}} 0 \cdots 0\right]. \tag{4.64}$$

Now Eq. 4.45 can be rewritten as

$$P(\hat{\boldsymbol{l}}_j) = \frac{1}{(2\pi)^{N_l}|\boldsymbol{W}_j|^{1/2}} \exp\left\{-\frac{1}{2}\dot{\boldsymbol{l}}_j^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j\right\}. \tag{4.65}$$

Substituting this individual local shape model probability back into the combined local shape model probability (Eq. 4.43), one can write

$$P(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_{N_c}) = \prod_{j=1}^{N_c} \frac{1}{(2\pi)^{N_l}|\boldsymbol{W}_j|^{1/2}} \exp\left\{-\frac{1}{2}\dot{\boldsymbol{l}}_j^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j\right\}. \tag{4.66}$$

4.2 Objective Function Formulation

As mentioned previously, the original ASM boundary-finding scheme optimizes the SM and the GLM independently. This leads to problems in that the optimal boundary selection will be biased towards either the SM or the GLM, whichever is considered last, instead of considering the contribution of both equally. Based on the probabilistic formulations developed in Sec. 4.1, we can formulate an objective function that, when optimized, will result in a boundary, $\boldsymbol{v}$, that corresponds to the maximum a posteriori

probability of $\boldsymbol{v}$, given a set of feature vectors, $\boldsymbol{G}$, extracted from the image. This objective function is rewritten here with the terms presented in a different order (see Eq. 4.44) for convenience:

$$J(\boldsymbol{v}) = \ln P(\boldsymbol{v}) + \ln P(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_{N_c}) + \ln p(\boldsymbol{G}|\boldsymbol{v}). \tag{4.67}$$

The first term is the GSM component, the second term is the LSM component, and the third term is GLM component. This function now must be expanded before it can be optimized. The details of the objective function expansion for each component are presented below.

4.2.1 Global Shape Objective Function

The global shape term, $P(\boldsymbol{v})$, is given by Eq. 4.29. Taking the logarithm of that function gives us the following expression:

$$\ln P(\boldsymbol{v}) = \sum_{i=1}^{t} \left(\ln \frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(v_i - \bar{v}_i)^2}{2\sigma_i^2} \right). \tag{4.68}$$

Removing terms that are independent of $\boldsymbol{v}$ results in

$$\ln P(\boldsymbol{v}) = -\sum_{i=1}^{t} \frac{(v_i - \bar{v}_i)^2}{2\sigma_i^2}. \tag{4.69}$$

The maximization problem can be turned into one of minimization by taking the negative and, without changing the optimization result, multiplying by a constant (2) to form the first component of the objective function in Eq. 4.67 as

$$J_1(\boldsymbol{v}) = \sum_{i=1}^{t} \frac{(v_i - \bar{v}_i)^2}{\sigma_i^2}. \tag{4.70}$$

Note that $J_1(\boldsymbol{v})$ is the squared Mahalanobis distance of the combined shape-pose parameter vector, $\boldsymbol{v}$, from the mean, $\bar{\boldsymbol{v}}$. Minimizing the squared Mahalanobis distance from the mean is a convenient way to describe the optimization process intuitively.

4.2.2 Local Shape Objective Function

Now the optimization of the local shape term must be addressed. The local shape joint probability term, $P(\hat{\boldsymbol{l}}_1, \ldots, \hat{\boldsymbol{l}}_{N_c})$, is given by Eq. 4.66. Taking the logarithm of that expression we have

$$\ln P(\hat{\boldsymbol{l}}_1, \ldots, \hat{\boldsymbol{l}}_{N_c}) = \sum_{j=1}^{N_c} \left(\ln\left(\frac{1}{(2\pi)^{N_j} |\boldsymbol{W}_j|^{1/2}} \right) - \frac{1}{2} \dot{\boldsymbol{l}}_j^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j \right). \tag{4.71}$$

Once again, removing terms independent of $\boldsymbol{v}$ results in

$$\ln P(\hat{\boldsymbol{l}}_1, \ldots, \hat{\boldsymbol{l}}_{N_c}) = -\frac{1}{2} \sum_{j=1}^{N_c} \dot{\boldsymbol{l}}_j^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j. \tag{4.72}$$

The second component of our objective function can be written as the sum of the squared Mahalanobis distances from each local shape model to its corresponding mean:

$$J_2(\boldsymbol{v}) = \sum_{j=1}^{N_c} \dot{\boldsymbol{l}}_j^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j. \tag{4.73}$$

4.2.3 Gray-Level Profile Objective Function

The third and final term of our objective function is needed that is dependent on the gray level profile information in the image—that is, the likelihood term. The likelihood term, $P(\boldsymbol{G}|\boldsymbol{v})$, is given by Eq. 4.41. Taking the logarithm, we can write

$$\ln P(\boldsymbol{G}|\boldsymbol{v}) = \sum_{j=1}^{N} \left(\ln\left(\frac{1}{(2\pi)^{N_g} |\boldsymbol{Q}_j|^{1/2}} \right) - \frac{1}{2} \dot{\boldsymbol{g}}_j^{\mathrm{T}} \boldsymbol{K}_{g_j} \dot{\boldsymbol{g}}_j \right). \tag{4.74}$$

Once again, all terms that are independent of $\boldsymbol{v}$ can be removed to write

$$\ln P(\boldsymbol{G}|\boldsymbol{v}) = -\frac{1}{2} \sum_{j=1}^{N} \dot{\boldsymbol{g}}_j^{\mathrm{T}} \boldsymbol{K}_{g_j} \dot{\boldsymbol{g}}_j. \tag{4.75}$$

Again, the third component of the objective function can be written as the sum of the squared Mahalanobis distances from each gray-level profile to its corresponding mean:

$$J_3(\boldsymbol{v}) = \sum_{j=1}^{N} \dot{\boldsymbol{g}}_j^{\mathrm{T}} \boldsymbol{K}_{g_j} \dot{\boldsymbol{g}}_j \tag{4.76}$$

4.2.4 Objective Function Formulation Summary

With all three terms of the objective function, $J(\boldsymbol{v}) = J_1(\boldsymbol{v}) + J_2(\boldsymbol{v}) + J_3(\boldsymbol{v})$, formulated, it can be optimized with respect to the combined shape-pose parameter vector, $\boldsymbol{v}$. As designed, J has one term for each of our model components: GSM (J_1), LSM (J_2), and GLM (J_3).

4.3 Objective Function Optimization

Next, the objective function, J, is optimized for the new PSAM algorithm. Because the objective function is a probabilistic one, J will be maximized with respect to $\boldsymbol{v}$ to ensure

that the most likely boundary is found, given the constraints imposed by the GSM, the LSM, and the GLM. The most effective optimization techniques are the gradient-based approaches [18], but these come with the penalty of having to calculate the gradient of the objective function. Also, whenever possible, the analytic form of the gradient will be calculated to ensure the highest accuracy of the gradient value. The gradient of each of the three PSAM components will be calculated individually as

$$\nabla J(\boldsymbol{v}) = \nabla J_1(\boldsymbol{v}) + \nabla J_2(\boldsymbol{v}) + \nabla J_3(\boldsymbol{v}). \tag{4.77}$$

Since $\boldsymbol{v}$ is a multidimensional vector, it is convenient when computing the gradient to rewrite the expression as

$$\nabla J(\boldsymbol{v}) = \begin{bmatrix} \frac{\partial J_1}{\partial v_1} \\ \frac{\partial J_1}{\partial v_2} \\ \cdots \\ \frac{\partial J_1}{\partial v_t} \end{bmatrix} + \begin{bmatrix} \frac{\partial J_2}{\partial v_1} \\ \frac{\partial J_2}{\partial v_2} \\ \cdots \\ \frac{\partial J_2}{\partial v_t} \end{bmatrix} + \begin{bmatrix} \frac{\partial J_3}{\partial v_1} \\ \frac{\partial J_3}{\partial v_2} \\ \cdots \\ \frac{\partial J_3}{\partial v_t} \end{bmatrix}. \tag{4.78}$$

We now calculate the gradient of each component of the model, starting with the GSM.

4.3.1 GSM Gradient Formulation

We formulated the GSM component (J_1) of the objective function in Eq. 4.70. If we take the gradient of this component with respect to our combined shape-pose parameter vector, $\boldsymbol{v}$ we can write

$$\frac{\partial}{\partial v_n} J_1(\boldsymbol{v}) = \frac{\partial}{\partial v_n} \left(\sum_{i=1}^{t} \frac{(v_i - \bar{v}_i)^2}{\sigma_i^2} \right). \tag{4.79}$$

All of the terms go to zero except when $i = n$, so that

$$\frac{\partial}{\partial v_n} J_1(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left(\frac{(v_n - \bar{v}_n)^2}{\sigma_n^2}\right), \tag{4.80}$$

or

$$\frac{\partial}{\partial v_n} J_1(\boldsymbol{v}) = \frac{2(v_n - \bar{v}_n)}{\sigma_n^2}. \tag{4.81}$$

4.3.2 LSM Gradient Formulation

We formulated the LSM component (J_2) of the objective function in Eq. 4.73. If we take the gradient of this component with respect to our combined shape-pose parameter vector, $\boldsymbol{v}$, we can write

$$\frac{\partial}{\partial v_n} J_2(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left(\sum_{j=1}^{N_c} \dot{\boldsymbol{l}}_j(\boldsymbol{v})^{\mathrm{T}} \boldsymbol{K}_{l_j} \dot{\boldsymbol{l}}_j(\boldsymbol{v})\right) = 2\sum_{j=1}^{N_c} \dot{\boldsymbol{l}}_j(\boldsymbol{v})^{\mathrm{T}} \boldsymbol{K}_{l_j} \frac{\partial}{\partial v_n} \dot{\boldsymbol{l}}_j(\boldsymbol{v}). \tag{4.82}$$

Note that $\dot{\boldsymbol{l}}_j$ is being written as $\dot{\boldsymbol{l}}_j(\boldsymbol{v})$ to indicate that it is a function of $\boldsymbol{v}$. We are now left with the task of finding an expression for $\frac{\partial}{\partial v_n} \dot{\boldsymbol{l}}_j(\boldsymbol{v})$. Using Eq. 4.55, Eq. 4.56, and Eq. 4.62, we can write

$$\frac{\partial}{\partial v_n} \dot{\boldsymbol{l}}_j(\boldsymbol{v}) = \frac{\partial}{\partial v_n}(\hat{\boldsymbol{l}}_j(\boldsymbol{v}) - \bar{\boldsymbol{l}}_j) = \frac{\partial}{\partial v_n} \hat{\boldsymbol{l}}_j(\boldsymbol{v}), \tag{4.83}$$

(since $\bar{\boldsymbol{l}}_j$ is not a function of $\boldsymbol{v}$) and, furthermore,

$$\frac{\partial}{\partial v_n} \hat{l}_{j_k}(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left[\tilde{s}_j(\boldsymbol{v})\left(l_{cen_{j_k}}(\boldsymbol{v})\cos\gamma_j(\boldsymbol{v}) - l_{cen_{j_{k+N_l}}}(\boldsymbol{v})\sin\gamma_j(\boldsymbol{v})\right)\right],$$

$$\forall k = 1, \ldots, N_l, \ \forall j = 1, \ldots, N_c, \tag{4.84}$$

and

$$\frac{\partial}{\partial v_n}\hat{l}_{j_k}(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left[\tilde{s}_j(\boldsymbol{v})\left(l_{cen_{j_{k-N_l}}}(\boldsymbol{v})\sin\gamma_j(\boldsymbol{v}) + l_{cen_{j_k}}(\boldsymbol{v})\cos\gamma_j(\boldsymbol{v})\right)\right],$$

$$\forall k = N_l + 1, \ldots, 2N_l,\ \forall j = 1, \ldots, N_c. \tag{4.85}$$

Note that $\boldsymbol{l}_{cen}$, $\tilde{s}$, and γ are all dependent on the combined shape-pose vector, $\boldsymbol{v}$. For convenience, we will drop the $\boldsymbol{v}$ when these terms are used during the rest of the derivation. Using the chain rule, we can rewrite Eq. 4.84 and Eq. 4.85 as

$$\begin{aligned}\frac{\partial}{\partial v_n}\hat{l}_{j_k}(\boldsymbol{v}) = \tilde{s}_j\Big(&\cos\gamma_j\frac{\partial}{\partial v_n}l_{cen_{j_k}} - l_{cen_{j_k}}\sin\gamma_j\frac{\partial}{\partial v_n}\gamma_j - \\ &\sin\gamma_j\frac{\partial}{\partial v_n}l_{cen_{j_{k+N_l}}} - l_{cen_{j_{k+N_l}}}\cos\gamma_j\frac{\partial}{\partial v_n}\gamma_j\Big) + \left(\cos\gamma_j l_{cen_{j_k}} - \sin\gamma_j l_{cen_{j_{k+N_l}}}\right)\frac{\partial}{\partial v_n}s_j\end{aligned}$$

$$\forall k = 1, \ldots, N_l, \tag{4.86}$$

and

$$\begin{aligned}\frac{\partial}{\partial v_n}\hat{l}_{j_k}(\boldsymbol{v}) = \tilde{s}_j\Big(&\sin\gamma_j\frac{\partial}{\partial v_n}l_{cen_{j_{k-N_l}}} + l_{cen_{j_{k-N_l}}}\cos\gamma_j\frac{\partial}{\partial v_n}\gamma_j + \\ &\cos\gamma_j\frac{\partial}{\partial v_n}l_{cen_{j_k}} - l_{cen_{j_k}}\sin\gamma_j\frac{\partial}{\partial v_n}\gamma_j\Big) + \left(\sin\gamma_j l_{cen_{j_{k-N_l}}} + \cos\gamma_j l_{cen_{j_k}}\right)\frac{\partial}{\partial v_n}s_j\end{aligned}$$

$$\forall k = N_l + 1, \ldots, 2N_l. \tag{4.87}$$

In each of the above equations, we are left with the task of finding the partial derivatives of l_{cen_j}, $\tilde{s}_j$, and γ_j. Starting with l_{cen_j} and recalling Eq. 4.49 and Eq. 4.50 we can write

$$\frac{\partial}{\partial v_n}l_{cen_{j_k}} = \frac{\partial}{\partial v_n}\left(l_{j_k} - \sum_{i=1}^{N_l}\frac{l_{j_i}}{N_l}\right) = \frac{\partial}{\partial v_n}l_{j_k} - \frac{1}{N_l}\left(\sum_{i=1}^{N_l}\frac{\partial}{\partial v_n}l_{j_i}\right),$$

$$\forall k = 1, \ldots, N_l \tag{4.88}$$

and

$$\frac{\partial}{\partial v_n} l_{cen_{j_k}} = \frac{\partial}{\partial v_n}\left(l_{j_k} - \sum_{i=N_l+1}^{2N_l} \frac{l_{j_i}}{N_l} \right) = \frac{\partial}{\partial v_n} l_{j_k} - \frac{1}{N_l}\left(\sum_{i=N_l+1}^{2N_l} \frac{\partial}{\partial v_n} l_{j_i} \right),$$

$$\forall k = N_l + 1, \ldots, 2N_l. \qquad 4.89$$

Note that in the two equations above for the partial of $\boldsymbol{l}_{cen_j}$, we are now left with the task of finding the partial of $\boldsymbol{l}_j$. If we look back to Eq. 4.48 where $\boldsymbol{l}_j$ is defined, we can write the partial derivative as

$$\frac{\partial}{\partial v_n} l_{j_k} = \frac{\partial}{\partial v_n} q_{h(j)_k}, \ \forall k = 1, \ldots, N_l, \qquad 4.90$$

and

$$\frac{\partial}{\partial v_n} l_{j_k} = \frac{\partial}{\partial v_n} r_{h(j)_{k-N_l}}, \ \forall k = N_l + 1, \ldots, 2N_l. \qquad 4.91$$

Substituting the expressions for $\boldsymbol{q}$ and $\boldsymbol{r}$ in Eq. 4.23 and Eq. 4.24 and also using the expression for $\hat{\boldsymbol{p}}$ in Eq. 4.25, we can write

$$\frac{\partial}{\partial v_n} l_{j_k} = \frac{\partial}{\partial v_n}(s\{(\hat{p}_{h(j)_k})\cos\theta - (\hat{p}_{N+h(j)_k})\sin\theta\} + T_x),$$

$$\forall k = 1, \ldots, N_l \qquad 4.92$$

and

$$\frac{\partial}{\partial v_n} l_{j_k} = \frac{\partial}{\partial v_n}(s\{(\hat{p}_{h(j)_{k-N_l}})\sin\theta + (\hat{p}_{N+h(j)_{k-N_l}})\cos\theta\} + T_y),$$

$$\forall k = N_l + 1, \ldots, 2N_l. \qquad 4.93$$

Recall that the variables s, θ, T_x, and T_y are the same as $v_{t_s+1}, \ldots, v_{t_s+4}$. We can simplify the above two equations using the chain rule. Starting with Eq. 4.92, we wish to find the individual partial derivatives of $l_j \quad \forall k = 1, \ldots, N_l$ with respect to $v_1, v_2, \ldots, v_{t_s}, v_{t_s+1}, \ldots, v_{t_s+4}$. We can write a single expression for the first t_s partial derivatives as

$$\frac{\partial}{\partial v_m} l_{j_k} = s(\Phi_{h(j)_k, m} \cos\theta - \Phi_{N+h(j)_k, m} \sin\theta)$$

$$\forall k = 1, \ldots, N_l \text{ and } \forall m = 1, \ldots, t_s. \qquad 4.94$$

The remaining partials for the $t_s + 1$ through $t_s + 4$ components are

$$\frac{\partial}{\partial v_{t_s+1}} l_{j_k} = \frac{\partial}{\partial s} l_{j_k} = (\hat{p}_{h(j)_k}) \cos\theta - (\hat{p}_{N+h(j)_k}) \sin\theta,$$

$$\frac{\partial}{\partial v_{t_s+2}} l_{j_k} = \frac{\partial}{\partial \theta} l_{j_k} = -s\{(\hat{p}_{h(j)_k}) \sin\theta + (\hat{p}_{N+h(j)_k}) \cos\theta\},$$

$$\frac{\partial}{\partial v_{t_s+3}} l_{j_k} = \frac{\partial}{\partial T_x} l_{j_k} = 1,$$

$$\frac{\partial}{\partial v_{t_s+4}} l_{j_k} = \frac{\partial}{\partial T_y} l_{j_k} = 0;$$

$$\forall k = 1, \ldots, N_l. \qquad 4.95$$

Similarly, for Eq. 4.93, we write a single expression for the first t_s partial derivatives as

$$\frac{\partial}{\partial v_m} l_{j_k} = s(\Phi_{h(j)_{k-N_l}, m} \sin\theta + \Phi_{N+h(j)_{k-N_l}, m} \cos\theta),$$

$$\forall k = N_l + 1, \ldots, 2N_l \text{ and } \forall m = 1, \ldots, t_s. \qquad 4.96$$

The remaining partials for the $t_s + 1$ through $t_s + 4$ components are

$$\frac{\partial}{\partial v_{t_s+1}} l_{j_k} = \frac{\partial}{\partial s} l_{j_k} = (\hat{L}_{h(j)_{k-N_l}}) \sin\theta + (\hat{L}_{N+h(j)_{k-N_l}}) \cos\theta$$

$$\frac{\partial}{\partial v_{t_s+2}} l_{j_k} = \frac{\partial}{\partial \theta} l_{j_k} = s\left\{(\hat{L}_{h(j)_{k-N_l}}) \cos\theta - (\hat{L}_{N+h(j)_{k-N_l}}) \sin\theta\right\}$$

$$\frac{\partial}{\partial v_{t_s+3}} l_{j_k} = \frac{\partial}{\partial T_x} l_{j_k} = 0$$

$$\frac{\partial}{\partial v_{t_s+4}} l_{j_k} = \frac{\partial}{\partial T_y} l_{j_k} = 1$$

$$\forall k = N_l + 1, \ldots, 2N_l. \qquad 4.97$$

Now, going back to Eq. 4.86 and Eq. 4.87, we have just completed the derivation of the term $\frac{\partial}{\partial v_n} l_{cen_j}$, but we still need to derive the partials of $\tilde{s}_j$ and γ_j.

Using the definition of $\tilde{s}_j$ in Eq. 4.51, we can write the partial derivative of $\tilde{s}_j$ as

$$\frac{\partial}{\partial v_n} \tilde{s}_j = \frac{\partial}{\partial v_n}\left\{(\tilde{a}_j^{\,2} + \tilde{b}_j^{\,2})^{1/2}\right\} = (\tilde{a}_j^{\,2} + \tilde{b}_j^{\,2})^{-\frac{1}{2}}\left(\tilde{a}_j \frac{\partial}{\partial v_n} \tilde{a}_j + \tilde{b}_j \frac{\partial}{\partial v_n} \tilde{b}_j\right), \qquad 4.98$$

where $\tilde{a}_j$ and $\tilde{b}_j$ are given by Eq. 4.53 and Eq. 4.54. We now must derive the partials on the right-hand side. Calculating the partial derivative of $\tilde{a}_j$, we find that

$$\frac{\partial}{\partial v_n} \tilde{a}_j = \frac{\partial}{\partial v_n}\left(\frac{\sum_{i=1}^{2N_l} l_{cen_{j_i}} \bar{l}_{j_i}}{\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2}\right) = \frac{\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2 \sum_{i=1}^{2N_l} \bar{l}_{j_i} \frac{\partial}{\partial v_n} l_{cen_{j_i}} - 2 \sum_{i=1}^{2N_l} l_{cen_{j_i}} \bar{l}_{j_i} \sum_{i=1}^{2N_l} l_{cen_{j_i}} \frac{\partial}{\partial v_n} l_{cen_{j_i}}}{\left(\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2\right)^2}. \qquad 4.99$$

We have derived an expression for $\frac{\partial}{\partial v_n} l_{cen_j}$ in Eq. 4.88 and Eq. 4.89, so we can move on to the derivation of the partial derivative of $\breve{b}_j$ by writing

$$\frac{\partial}{\partial v_n}\breve{b}_j = \frac{\partial}{\partial v_n}\left(\frac{\left(\sum_{i=1}^{N_l} l_{cen_{j_i}}\overline{l_{j_{N_l+i}}} - \sum_{i=1}^{N_l} l_{cen_{j_{N_l+i}}}\overline{l_{j_i}}\right)}{\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2}\right),$$

or

$$\frac{\partial}{\partial v_n}\breve{b}_j = \left\{\left(\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2\right)\left(\sum_{i=1}^{N_l}\overline{l_{j_{N_l+i}}}\frac{\partial}{\partial v_n} l_{cen_{j_i}} - \sum_{i=1}^{N_l}\overline{l_{j_i}}\frac{\partial}{\partial v_n} l_{cen_{j_{N_l+i}}}\right) - \right. \quad . \quad 4.100$$

$$\left. 2\left(\sum_{i=1}^{N_l} l_{cen_{j_i}}\overline{l_{j_{N_l+i}}} - \sum_{i=1}^{N_l} l_{cen_{j_{N_l+i}}}\overline{l_{j_i}}\right)\left(\sum_{i=1}^{2N_l} l_{cen_{j_i}}\frac{\partial}{\partial v_n} l_{cen_{j_i}}\right)\right\}\left(\sum_{i=1}^{2N_l} l_{cen_{j_i}}^2\right)^{-2}$$

Now we have all of the terms derived for $\frac{\partial}{\partial v_n}\tilde{s}_j$, and the only unknown quantity left in Eq. 4.86 and Eq. 4.87 is the partial derivative of our local shape rotation term, γ_j. Referring back to Eq. 4.52, this can be found as follows:

$$\frac{\partial}{\partial v_n}\gamma_j = \frac{\partial}{\partial v_n}\left(\operatorname{atan}\frac{\breve{b}_j}{\tilde{a}_j}\right) = \frac{\tilde{a}_j^2}{\tilde{a}_j^2 + \breve{b}_j^2}\frac{\partial}{\partial v_n}\frac{\breve{b}_j}{\tilde{a}_j} = \frac{\tilde{a}_j\frac{\partial}{\partial v_n}\breve{b}_j - \breve{b}_j\frac{\partial}{\partial v_n}\tilde{a}_j}{\tilde{a}_j^2 + \breve{b}_j^2} \qquad 4.101$$

This completes the derivation of the $\frac{\partial}{\partial v_n} J_2(\boldsymbol{v})$, the gradient of the LSM component of our three-part objective function.

4.3.3 GLM Gradient Formulation

The final component of the objective function to be differentiated is the gray-level model, $J_3(\boldsymbol{v})$. The equation for the gray-level component is given in Eq. 4.76. Note that the form of this component (squared Mahalanobis distance from the mean) is similar to that of the LSM given in Eq. 4.73. We can find the gradient of $J_3(\boldsymbol{v})$ as follows:

$$\frac{\partial}{\partial v_n} J_3(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left(\sum_{j=1}^{N} \dot{\boldsymbol{g}}_j(\boldsymbol{v})^{\mathrm{T}} \boldsymbol{K}_{g_j} \dot{\boldsymbol{g}}_j(\boldsymbol{v}) \right) = 2 \sum_{j=1}^{N} \dot{\boldsymbol{g}}_j(\boldsymbol{v})^{\mathrm{T}} \boldsymbol{K}_{g_j} \frac{\partial}{\partial v_n} \dot{\boldsymbol{g}}_j(\boldsymbol{v}). \qquad 4.102$$

Note that $\dot{\boldsymbol{g}}_j$ is being written as $\dot{\boldsymbol{g}}_j(\boldsymbol{v})$ to indicate that it is a function of $\boldsymbol{v}$. Using Eq. 4.37, we can the calculate the term $\frac{\partial}{\partial v_n} \dot{\boldsymbol{g}}_j(\boldsymbol{v})$ as follows:

$$\frac{\partial}{\partial v_n} \dot{\boldsymbol{g}}_j(\boldsymbol{v}) = \frac{\partial}{\partial v_n} (\boldsymbol{g}_j(\boldsymbol{v}) - \bar{\boldsymbol{g}}_j) = \frac{\partial}{\partial v_n} \boldsymbol{g}_j(\boldsymbol{v}). \qquad 4.103$$

The partial derivative of $\boldsymbol{g}_j$ depends, of course, on how the gray-level profiles are defined during training, as well as on the operator, $h(\boldsymbol{g}_j)$. This research accommodates six possible types of gray level profile accommodated in this research. The profile samples can be one of the following

(1) absolute image intensity values:

$$g_{j_k}(\boldsymbol{v}) = \boldsymbol{I}(Q_{j_k}(\boldsymbol{v}), R_{j_k}(\boldsymbol{v})), \ \forall k = 1, \ldots, N_g, \qquad 4.104$$

(2) edge profiles based on a first-forward difference gradient:

$$g_{j_k}(\boldsymbol{v}) = \boldsymbol{I}(Q_{j_k}(\boldsymbol{v}), R_{j_k}(\boldsymbol{v})) - \boldsymbol{I}(Q_{j_{k+1}}(\boldsymbol{v}), R_{j_{k+1}}(\boldsymbol{v})), \ \forall k = 1, \ldots, N_g, \qquad 4.105$$

(3) edge profiles based on a sobel gradient:

$$g_{j_k}(v) = I(Q_{j_{k-1}}(v), R_{j_{k-1}}(v)) + 2I(Q_{j_k}(v), R_{j_k}(v)) - I(Q_{j_{k+1}}(v), R_{j_{k+1}}(v)),$$

$$\forall k = 1, \ldots, N_g. \qquad 4.106$$

Here, $\boldsymbol{Q}_j$ and $\boldsymbol{R}_j$ are vectors that contain the values of the (x, y) coordinate locations of the samples along the j^{th} normal profile and are defined as (see Eq. 4.30)

$$Q_{j_k} = x_j - \left(\frac{(N_g - 1)}{2} - k + 1\right)\cos\alpha_j, \ \forall k = 1, \ldots, N_g, \qquad 4.107$$

$$R_{j_k} = y_j - \left(\frac{(N_g - 1)}{2} - k + 1\right)\sin\alpha_j, \ \forall k = 1, \ldots, N_g. \qquad 4.108$$

where α_j is the angle of the j^{th} profile. Also recall that $\boldsymbol{I}$ is the image under test and that (q_j, r_j) is the pose-corrected coordinate pair indicating the location of the j^{th} LP within the image. Each of these three profile types can either be un-normalized or normalized by the sum of the absolute values of the profile elements,

$$\sum_{m=1}^{N_g} |g_{j_m}|,$$

This results in six possible gray-level profile measurements. Here, we address the two most commonly used profiles: normalized absolute intensity and normalized first-forward difference gradient.

For normalized absolute intensity, our profile is defined as

$$g_{j_k}(v) = \frac{I(Q_{j_k}(v), R_{j_k}(v))}{\sum_{m=1}^{N_g} |I(Q_{j_m}(v), R_{j_m}(v))|}, \ \forall k = 1, \ldots, N_g. \qquad 4.109$$

We can then write the partial derivative as

$$\frac{\partial}{\partial v_n} g_{j_k}(\boldsymbol{v}) = \frac{\partial}{\partial v_n}\left(\frac{I(Q_{j_k}(\boldsymbol{v}), R_{j_k}(\boldsymbol{v}))}{\sum_{m=1}^{N_g} \left| I(Q_{j_m}(\boldsymbol{v}), R_{j_m}(\boldsymbol{v})) \right|} \right), \forall k = 1, \ldots, N_g. \qquad 4.110$$

We can expand, drop the absolute value (image pixels are always nonnegative), temporarily drop the $\boldsymbol{v}$ for convenience, and write

$$\frac{\partial}{\partial v_n} g_{j_k}(\boldsymbol{v}) = \frac{\left(\sum_{m=1}^{N_g} I(Q_{j_m}, R_{j_m}) \right) \frac{\partial}{\partial v_n} I(Q_{j_k}, R_{j_k}) - I(Q_{j_k}, R_{j_k}) \sum_{m=1}^{N_g} \frac{\partial}{\partial v_n} I(Q_{j_m}, R_{j_m})}{\left(\sum_{m=1}^{N_g} I(Q_{j_m}, R_{j_m}) \right)^2}$$

$$\forall k = 1, \ldots, N_g. \qquad 4.111$$

The only unknown term in the above expression is the partial derivative of the image along the profile, $I(Q_{j_k}, R_{j_k})$. Applying the chain rule, we can write this partial derivative as follows:

$$\frac{\partial}{\partial v_n} I(Q_{j_k}, R_{j_k}) = \frac{\partial}{\partial Q_{j_k}} I \frac{\partial}{\partial v_n} Q_{j_k} + \frac{\partial}{\partial R_{j_k}} I \frac{\partial}{\partial v_n} R_{j_k}, \qquad 4.112$$

We do not have an analytical form of the image, I, so we will calculate its partial derivatives via a finite-difference operation as

$$\frac{\partial}{\partial Q_{j_k}} I = \frac{1}{2\varepsilon} [I(Q_{j_k} - \varepsilon, R_{j_k}) - I(Q_{j_k} + \varepsilon, R_{j_k})], \qquad 4.113$$

and

$$\frac{\partial}{\partial R_{j_k}} \boldsymbol{I} = \frac{1}{2\varepsilon}[I(Q_{j_k}, R_{j_k} - \varepsilon) - I(Q_{j_k}, R_{j_k} + \varepsilon)] \tag{4.114}$$

We can derive the partials of $\boldsymbol{Q}_j$ and $\boldsymbol{R}_j$ as follows:

$$\frac{\partial}{\partial v_n} Q_{j_k} = \frac{\partial}{\partial v_n} x_j + \left(\frac{(N_g - 1)}{2} - k + 1\right) \sin\alpha_j \frac{\partial}{\partial v_n} \alpha_j \tag{4.115}$$

and

$$\frac{\partial}{\partial v_n} R_{j_k} = \frac{\partial}{\partial v_n} y_j - \left(\frac{(N_g - 1)}{2} - k + 1\right) \cos\alpha_j \frac{\partial}{\partial v_n} \alpha_j . \tag{4.116}$$

Going back to Eq. 4.23, we can derive the partial of x_j as

$$\frac{\partial}{\partial v_m} x_j = s(\Phi_{j,m} \cos\theta - \Phi_{N+j,m} \sin\theta), \forall m = 1, \ldots, t_s, \tag{4.117}$$

and for the $t_s + 1, \ldots, t_s + 4$ partials with respect to $\boldsymbol{v}$, we can write

$$\frac{\partial}{\partial v_{t_s+1}} x_j = \frac{\partial}{\partial s} x_j = (\hat{L}_j)\cos\theta - (\hat{L}_{N+j})\sin\theta , \tag{4.118}$$

$$\frac{\partial}{\partial v_{t_s+2}} x_j = \frac{\partial}{\partial \theta} x_j = -s\{(\hat{L}_j)\sin\theta + (\hat{L}_{N+j})\cos\theta\} , \tag{4.119}$$

$$\frac{\partial}{\partial v_{t_s+3}} x_j = \frac{\partial}{\partial T_x} x_j = 1 , \tag{4.120}$$

and

$$\frac{\partial}{\partial v_{t_s+4}} x_j = \frac{\partial}{\partial T_y} x_j = 0 . \tag{4.121}$$

Similarly, going back to Eq. 4.24, the partial derivative of y_j is

$$\frac{\partial}{\partial v_m} y_j = s(\Phi_{j,m}\sin\theta + \Phi_{N+j,m}\cos\theta), \forall m = 1, \ldots, t_s, \qquad 4.122$$

and for the $t_s + 1, \ldots, t_s + 4$ partials with respect to v, we can write

$$\frac{\partial}{\partial v_{t_s+1}} y_j = \frac{\partial}{\partial s} y_j = (\hat{L}_j)\sin\theta + (\hat{L}_{N+j})\cos\theta, \qquad 4.123$$

$$\frac{\partial}{\partial v_{t_s+2}} y_j = \frac{\partial}{\partial \theta} y_j = s\{(\hat{L}_j)\cos\theta - (\hat{L}_{N+j})\sin\theta\}, \qquad 4.124$$

$$\frac{\partial}{\partial v_{t_s+3}} y_j = \frac{\partial}{\partial T_x} y_j = 0, \qquad 4.125$$

and

$$\frac{\partial}{\partial v_{t_s+4}} y_j = \frac{\partial}{\partial T_y} y_j = 1. \qquad 4.126$$

As mentioned previously, the other commonly used form of the gray-level profile in this research is the normalized first-forward difference gradient. The expression for the profile in this case is

$$g_{j_k}(v) = \frac{I(Q_{j_k}, R_{j_k}) - I(Q_{j_{k+1}}, R_{j_{k+1}})}{\sum_{m=1}^{N_g} \left| I(Q_{j_m}, R_{j_m}) - I(Q_{j_{m+1}}, R_{j_{m+1}}) \right|}, \forall k = 1, \ldots, N_g. \qquad 4.127$$

When deriving the gradient of this profile, one must be sure to treat the absolute value in the normalization term (the denominator) in the above expression correctly. Recall that the derivative of the absolute value of a function is as follows:

$$\frac{d}{dx}|f(x)| = \begin{matrix} f'(x); \{\forall x; f(x) \geq 0\} \\ -f'(x); \{\forall x; f(x) < 0\} \end{matrix}. \qquad 4.128$$

The gradient of the normalized profile in Eq. 4.127 results in a lengthy but straightforwardly implemented result. This final expression still requires one to calculate the gradient of the image along the profile points—that is, $\frac{\partial}{\partial v_n} \boldsymbol{I}(Q_{j_k}, R_{j_k})$. The expressions derived starting with Eq. 4.112 can be used in this case as well.

4.3.4 Optimization of the Objective Function

With the analytic gradient calculated, we can now apply a variety of gradient-based optimization schemes to find the minimum of our objective function. This minimum value occurs at a value of $\boldsymbol{v}$ that defines the boundary that simultaneously best fits all three components of our model: global shape, local shape, and gray-level profile appearance.

Gradient descent was chosen as the optimization approach for this research because of its simplicity. The use of other gradient-based optimization techniques such as the conjugate gradient technique can be explored in future research. The iterative update formula for gradient descent in this case is

$$\boldsymbol{v}_{new} = \boldsymbol{v}_{old} - r\nabla \boldsymbol{J}(\boldsymbol{v}) = \boldsymbol{v}_{old} - r[\nabla J_1(\boldsymbol{v}) + \nabla J_2(\boldsymbol{v}) + \nabla J_3(\boldsymbol{v})], \qquad 4.129$$

where r is the parameter that sets the length of the step size in the direction of the gradient. We can monitor the value of the gradient norm at each iteration. When its value is close to zero, we know we have found a minimum, and the process terminates.

Recall that we have found our optimal boundary (or set of LPs) as represented by the shape-pose parameter vector, $\boldsymbol{v}$, which exists in the PCA subspace (refer back to Sec. 3.1.3 and Sec. 4.1.1 for details). Recall that the first t_s elements of $\boldsymbol{v}$ are defined as the global shape parameter vector, $\boldsymbol{b}$, and the last four elements of $\boldsymbol{v}$ are the pose elements (scale, rotation, x-translation and y-translation). The two final steps to convert the PCA-based

parameter vector, v, into our image-based LP vector, p, are (1) inverse-PCA-transform v to generate the aligned (to the mean global shape) LP vector, $\hat{p}$, and (2) apply the pose transformation, P_{s,θ,T_x,T_y}, to generate the final LP vector, p. The following formulas accomplish this final step:

$$\hat{p} = \bar{p} + \Phi b = \bar{p} + \begin{bmatrix} \phi_1 & \cdots & \phi_{t_s} \end{bmatrix} \begin{bmatrix} b_1 \\ \cdots \\ b_{t_s} \end{bmatrix}, \tag{4.130}$$

$$p = P_{s,\theta,T_x,T_y}(\hat{p}) = [x^T y^T]. \tag{4.131}$$

4.3.5 Multi-Resolution PSAM Search

The PSAM algorithm is a multiresolution approach in that this optimization of the objective function is performed at as many resolutions as indicated by the operator during training (see Sec. 3.1.7). Typically, three resolutions (quarter, half, and full) are specified in applications, so that this optimization procedure is performed at three different scales. When the optimization terminates at one resolution, the resulting boundary is scaled up (by a factor of 2) to the next highest resolution as the initial guess for that level.

4.3.6 Weight Parameters for PSAM Components

One advantage of having the update formula in the form of Eq. 4.129 is we can introduce weighting factors for each of the components of our PSAM to emphasize one component of the model over the others. Introducing weighting factors for each of our PSAM components, we can rewrite our update formula in Eq. 4.129 as

$$v_{new} = v_{old} - r\nabla J(v) = v_{old} - r[\nabla J_1(v, w_1) + \nabla J_2(v, w_2) + \nabla J_3(v, w_3)], \tag{4.132}$$

where w_1, w_2, and w_3 are all scalars ranging from 0 to 1. A value of 1 for each of these weighting factors allows them to have full influence on the optimization result. As the value of the weighting factor decreases, the corresponding model component has less influence on the result.

In the two simplest cases of the LSM and GLM (J_2 and J_3), the weighting factors w_2 and w_3 scale the contribution of the gradient; that is:

$$\nabla J_1(\boldsymbol{v}, w_1) = w_1 \nabla J_1(\boldsymbol{v}) \qquad 4.133$$

and

$$\nabla J_2(\boldsymbol{v}, w_2) = w_2 \nabla J_2(\boldsymbol{v}). \qquad 4.134$$

This has the effect of increasing or reducing the contribution of that model component to the total gradient result. Note, however, that the formulation of the weighted GSM is treated differently. The gradient of the GSM is not simply multiplied by the weighting factor, w_1. This is because the purpose of the GSM is to influence the shape of the deforming boundary to resemble, in a global sense, the shape of the boundaries as delineated in the training set. The GSM accomplishes this goal, in part, via the last step of the optimization process—that is, the inverse PCA transform in Eq. 4.130. The only modes of variation (i.e., eigenvectors, $\phi_1, \ldots, \phi_{t_s}$, of the training covariance matrix, $\boldsymbol{C}$) that were retained in the PCA transform were the largest ones, which accounted for most of the training set variation. Because the inverse PCA transformation contains only those modes, it will transform any shape parameter vector into a set of LPs that resembles the global shape as parametrized by the GSM. The key, as is described next, is to use the weighting factor, w_1, earlier in the process (during training) to reduce the influence of the GSM.

The influence of the GSM on the final LP boundary is entirely determined during the training process. More precisely, it depends on the form of the training set covariance matrix, C, from which the modes of variation are extracted via an eigenvalue/eigenvector computation. At one extreme, where the GSM has full influence, we can (as we have to this point) depend completely on the measured covariance between all LPs within the training set. In this scenario, the movement of one LP is dependent on every other LP's movement to a level proportional to the covariance between those two LPs. At the other extreme, where the GSM has no influence, we can say a priori that all LPs are independent of one another and can move freely. In this scenario, the covariance between all LPs is zero, and the covariance matrix would be diagonal. This observation has also been noted by Wang and Staib in [96]. This provides a clue into how one could control, via a weighting factor, the influence of the GSM on the form of the final LP boundary.

The number of points in the LP boundary has been previously defined as N. Consider the factor, w_1N, to be the size of the LP boundary neighborhood within which the movement of one LP can influence another. It is convenient to force the size of the neighborhood to be an odd number so that it includes a current LP and an equal number of neighbors on each side of the LP. If $w_1 = 1$, then all N LPs can have influence on one another as dictated by the covariance within the training set. If $w_1 = 0$, then all LPs move independently. All other values of w_1 define a local neighborhood of a size less than N for which LPs will have influence on one another. This can be implemented simply be zeroing the appropriate diagonals of the training covariance matrix before performing the PCA. For example, if an LP boundary contained six LPs, and w_1 was equal to 0.5, this would

define a neighborhood size of three LPs. The covariance matrix would then be defined to have the following form:

$$\begin{bmatrix}
c_{x_1} & c_{x_1x_2} & 0 & 0 & 0 & c_{x_1x_6} & c_{x_1y_1} & c_{x_1y_2} & 0 & 0 & 0 & c_{x_1y_6} \\
c_{x_2x_1} & c_{x_2} & c_{x_2x_3} & 0 & 0 & 0 & c_{x_2y_1} & c_{x_2y_2} & c_{x_2y_3} & 0 & 0 & 0 \\
0 & c_{x_3x_2} & c_{x_3} & c_{x_3x_4} & 0 & 0 & 0 & c_{x_3y_2} & c_{x_3y_3} & c_{x_3y_4} & 0 & 0 \\
0 & 0 & c_{x_4x_3} & c_{x_4} & c_{x_4x_5} & 0 & 0 & 0 & c_{x_4y_3} & c_{x_4y_4} & c_{x_4y_5} & 0 \\
0 & 0 & 0 & c_{x_5x_4} & c_{x_5} & c_{x_5x_6} & 0 & 0 & 0 & c_{x_5y_4} & c_{x_5y_5} & c_{x_5y_6} \\
c_{x_6x_1} & 0 & 0 & 0 & c_{x_6x_5} & c_{x_6} & c_{x_6y_1} & 0 & 0 & 0 & c_{x_6y_5} & c_{x_6y_6} \\
c_{y_1x_1} & c_{y_1x_2} & 0 & 0 & 0 & c_{y_1x_6} & c_{y_1} & c_{y_1y_2} & 0 & 0 & 0 & c_{y_1y_6} \\
c_{y_2x_1} & c_{y_2x_2} & c_{y_2x_3} & 0 & 0 & 0 & c_{y_2y_1} & c_{y_2} & c_{y_2y_3} & 0 & 0 & 0 \\
0 & c_{y_3x_2} & c_{y_3x_3} & c_{y_3x_4} & 0 & 0 & 0 & c_{y_3y_2} & c_{y_3} & c_{y_3y_4} & 0 & 0 \\
0 & 0 & c_{y_4x_3} & c_{y_4x_4} & c_{y_4x_5} & 0 & 0 & 0 & c_{y_4y_3} & c_{y_4} & c_{y_4y_5} & 0 \\
0 & 0 & 0 & c_{y_5x_4} & c_{y_5x_5} & c_{y_5x_6} & 0 & 0 & 0 & c_{y_5y_4} & c_{y_5} & c_{y_5y_6} \\
c_{y_6x_1} & 0 & 0 & 0 & c_{y_6x_5} & c_{y_6x_6} & c_{y_6y_1} & 0 & 0 & 0 & c_{y_6y_5} & c_{y_6}
\end{bmatrix} \quad 4.135$$

Note that the only nonzero covariance values occur within the neighborhood size defined by the product of w_1 and N. This can also be thought of as building in a scale-based shape similarity in which resulting LP boundaries will resemble the training set boundaries at a scale defined by the value of w_1. Note that if $w_1 = 0$, then the covariance matrix becomes diagonal so that all covariance values are zero, and boundary LPs are free to move independently.

4.4 Confidence Metric Formulation

One of the goals of this research was to develop a confidence metric, or "goodness-of-fit" measurement, that accompanies the LP boundary result. The purpose of this metric is to quantify how well the final LP boundary fits within the shape and gray-level profile

appearance parameters measured from the training set. Although the result of this metric will not necessarily predict the accuracy of the segmentation result, it will indicate the level to which the result resembles the boundaries defined in the training set. As will be seen in Chap. 5, however, a correlation between one component of the developed metric and the segmentation accuracy is identified, allowing the confidence metric to be used as an approximate predictor of segmentation accuracy in some cases.

Recall that during the formulation of the three-component objective function in Sec. 4.2, the value of the objective function for each component (J_1, J_2, and J_3) was the squared Mahalanobis distance of that component's value from the mean. We can use this value straightforwardly to formulate a confidence metric. Immediately after the PSAM training process, which results in the creation of the three-component PSAM, the values of J_1, J_2, and J_3 are computed and stored for every user-delineated LP boundary in the training set. The maximum values of the squared Mahalanobis distances are saved for the GSM, LSM, and GLM across all of the training samples. These maximum values (called $J_{1_{max}}$, $J_{2_{max}}$, and $J_{3_{max}}$) correspond to the most "ill-fitting" LP boundary in the training set relative to each of the PSAM components. These maxima can then be used to set the bounds for each component of the PSAM in terms of how far a new LP boundary measurement should stray from the mean global shape, mean local shape, and mean gray-level profile appearance. These maximum distances are used to normalize the measured squared Mahalanobis distances for a newly computed LP boundary and result in a confidence metric. Confi-

dence metrics can then be computed for each individual model component using the following formulas:

$$c_1 = \frac{J_{1_{max}} - J_1}{J_{1_{max}}}, \tag{4.136}$$

$$c_2 = \frac{J_{2_{max}} - J_2}{J_{2_{max}}}, \tag{4.137}$$

$$c_3 = \frac{J_{3_{max}} - J_3}{J_{3_{max}}}. \tag{4.138}$$

If the value of these confidence metrics is equal to 1, the corresponding component of the LP boundary is equal to the training set mean (Mahalanobis distance of zero). As the values of c_1, c_2, and c_3 decrease, the LP boundary is moving farther away from the mean. At zero, the LP boundary has moved as far away from the mean as the most "ill-fitting" training sample. A negative value for the confidence metric indicates that the LP boundary has fallen outside the bounds of the training set, while a value between 0 and 1 indicates that the result fits within the bounds as determined by the training samples. It is convenient to inspect the individual confidence metrics of c_1, c_2, and c_3 to see how well the current LP boundary satisfies the GSM, LSM, and GLM, respectively. In addition, note that c_1 represents the GSM confidence metric, which has two main components: the shape and the pose. If desired, we can straightforwardly treat each component individually and generate a confidence metric for each component and call them $c_{1_{shape}}$ and $c_{1_{pose}}$. This allows us to check the shape and pose confidences individually.

4.5 PSAM Development Summary

This concludes the theoretical development of the PSAM. Sec. 4.1 outlined four major goals that this new PSAM formulation would satisfy. Let us examine those goals here to summarize how they have been met.

1. **Reformulate the statistical model objective function so that it considers both the gray-level and shape information simultaneously during contour deformation.**

   This new PSAM objective function combines the a priori information contained in the training set with regard to boundary shape and gray-level profile appearance. These two components of the model are no longer optimized independently, but simultaneously. This increases the overall probability that a global minimum will be found during optimization.

2. **Employ a continuous penalty for global shape variations as dictated by the training set.**

   We have a formulated a continuous, probabilistic objective function that, during the optimization process, penalizes a boundary whose shape and gray-level profiles vary from those dictated by the training set.

3. **Formulate models of local shape characteristics to provide better control of the relative local movement of LPs.**

   A local shape modeling scheme was developed and incorporated into the overall PSAM objective function. This local modeling is done in a manner that is very similar to the global shape modeling scheme.

4. **Develop a PSAM confidence metric.**

   The Gaussian-based formulation of each of the model components allows a straightforward measure of the goodness-of-fit of the final boundary with regard to the training data. This confidence metric is based on the Mahalanobis distance of the final boundary's characteristic (shape and gray-level profile appearance) from the corresponding mean characteristic.

The following chapter discusses the implementation of this technique and presents experimental results that demonstrate the useful characteristics of the PSAM boundary-finding formulation.

CHAPTER 5

EXPERIMENTAL RESULTS

This chapter presents the results of the application of the new PSAM algorithm on a variety of data sets, including both synthetic and real images. The chapter first presents some results on synthetic data to illustrate the effect of the three PSAM weighting parameters formulated in Sec. 4.3.6 that allow the operator to control the influence of each of the three model components (GSM, LSM, and GLM). Then three medical-image data sets taken from high-resolution X-ray CT volumes of laboratory mice are segmented via PSAM and ASM, and the accuracy of the segmentation results are compared. Other boundary-finding researchers have used ASM as a starting point ([87], [59], and [96]), but none have done a direct comparison of performance between their segmentation technique and ASM. Finally, this chapter explores the specific interaction of the global and local shape components of the model through experiments on medical images and some ivy-leaf-image data.

5.1 Weight Parameter Experiment on Synthetic Data

Recall from Sec. 4.3.6 that three separate weight components—w_1, w_2, and w_3—were formulated to control the relative influence of the GSM, the LSM, and the GLM, respectively, during the contour optimization. This section presents some results on synthetic data that clearly illustrate how these weighting parameters can be used to achieve applica-

tion-specific results. Also presented here are the confidence metric values that indicate how well the resulting boundaries fit within the distribution defined by the training data.

The synthetic data used for this experiment consisted of a collection of star images such as the one shown in Fig. 5.1. For simplicity, the background is black and the star is white. The training set consisted of a set of seven images created by translating and scaling the original star image to seven different positions and sizes. The LP boundary of each star in the training set was manually delineated.

In the first weight parameter test the first star image was held out, the remaining six examples were used for training, and the PSAM algorithm was run the GSM influence was changing. The LSM influence was set to zero for this test. The result of this experiment is shown in Fig. 5.2. Note that even with the global and local shape parameters w_1 and w_2 set to zero [Fig. 5.2(b), no shape influence], the strong edge information in this synthetic image allows PSAM to accurately find the boundary of the star relying only on

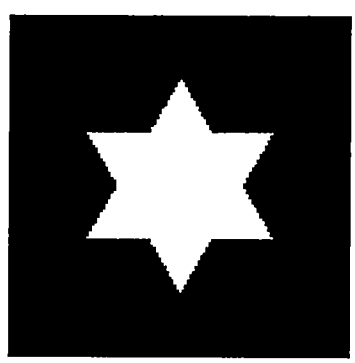
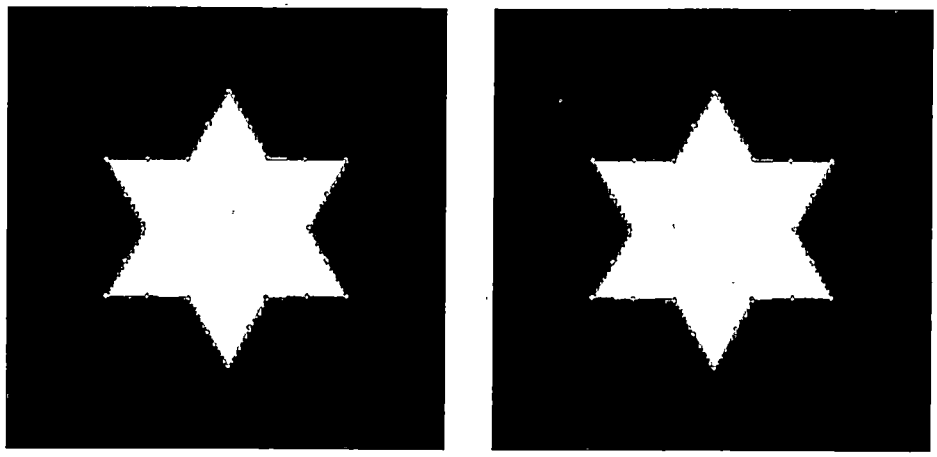


Fig. 5.1 Example of synthetic image used for weight parameter testing.

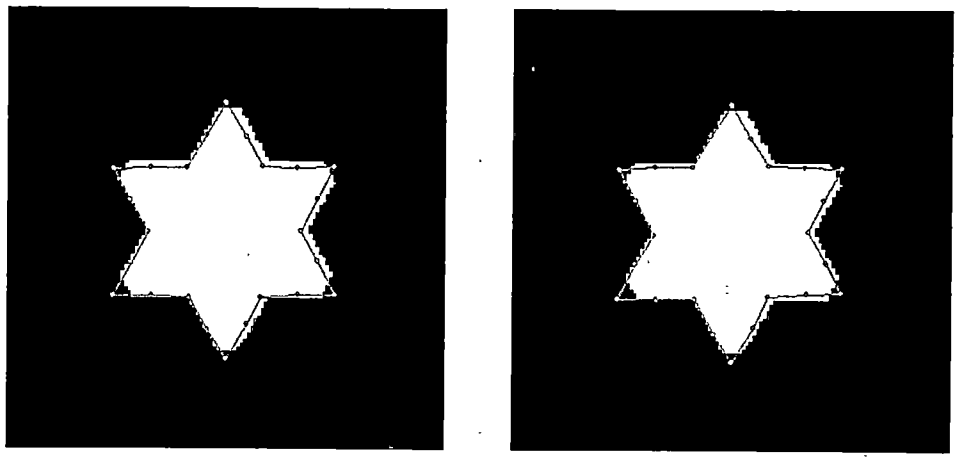


(a) (b)

Fig. 5.2 Segmentation results on star image. Parameter settings: (a) $w_1 = 1$, $w_2 = 0$, and $w_3 = 1$ and (b) $w_1 = 0$, $w_2 = 0$, and $w_3 = 1$. The resulting confidence values for (a) are $c_1 = 0.85$, $c_2 = 0.97$, and $c_3 = 0.53$ and for (b) are $c_1 = 0.85$, $c_2 = 0.87$, and $c_3 = 0.53$.

gray-level information. All of the confidence metrics fall between 0 and 1 for both cases, indicating that the boundary result fits well within the distribution of training examples.

The next experiment introduces a new test image containing a star that has had its points cut off. The resulting star does not look like the images in the training set, and, hence, is useful for illustration purposes. Again, we start with a GSM weight of 1 and then reduce the influence of the GSM over several steps. As can be seen in Fig. 5.3, with a GSM weight of 1, the final boundary is forced to look like the stars in the training set and, hence, does not fit this deformed star very well. The large negative value of the GLM confidence metric ($c_3 = -16.17$) indicates that the final gray-level profiles do not match those defined during training. In practice, a large negative value for any of the three confidence metrics should indicate the occurrence of an unusual result. Though the resulting bound-

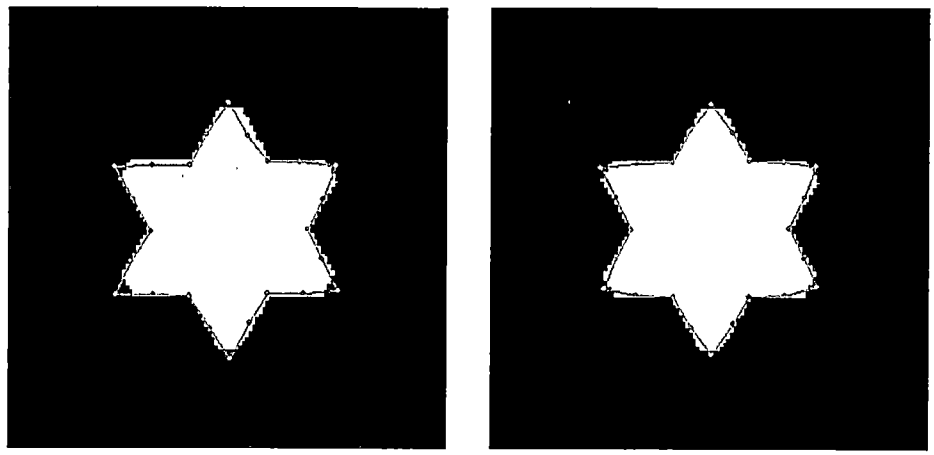


(a) (b)

Fig. 5.3 New segmentation results on star image. Parameter settings: (a) $w_1 = 1$, $w_2 = 0$, and $w_3 = 1$ and (b) $w_1 = 0.5$, $w_2 = 0$, and $w_3 = 1$. The resulting confidence values for (a) are $c_1 = -0.14$, $c_2 = 0.71$, and $c_3 = -16.17$ and for (b) are $c_1 = -0.42$, $c_2 = -0.07$, and $c_3 = -14.92$.

ary may represent a reasonable segmentation, the fact that it looks different from the training set promotes manual intervention by the operator.

If we reduce the influence of the GSM, this increases the relative influence of the GLM, and the final boundary begins to fit the image edges more closely. Note that this happens to a small degree when the GSM weight is reduced to 0.5 in Fig. 5.3. The images in Fig. 5.4 show the results of reducing the GSM weight further, to 0.25 and then to zero. Note that when the GSM influence is reduced to zero, all of the LPs on the boundary fall very to close image edges. This is because the GLM has all of the control in the PSAM objective function, so the boundary will move to edge information independent of prior boundary shape. Finally, note that, in general, the confidence metrics for the GSM and LSM become more negative, indicating a poor fit to the shape-training data. Alternatively,

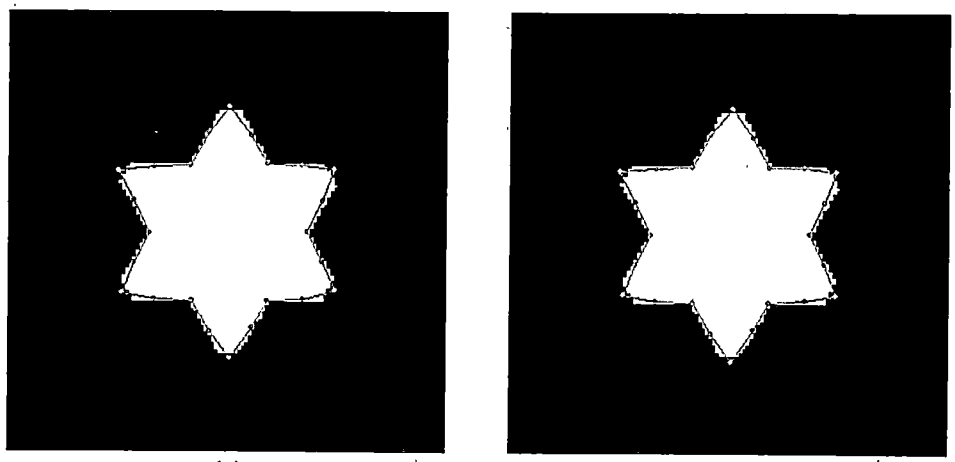


(a) (b)

Fig. 5.4 Segmentation results on star image. Weight parameters are: (a) $w_1 = 0.25$, $w_2 = 0$, and $w_3 = 1$ and (b) $w_1 = 0$, $w_2 = 0$, and $w_3 = 1$. The resulting confidence values for (a) are $c_1 = -0.70$, $c_2 = -0.56$, and $c_3 = -10.24$ and for (b) are $c_1 = -0.60$, $c_2 = -4.14$, and $c_3 = -6.28$.

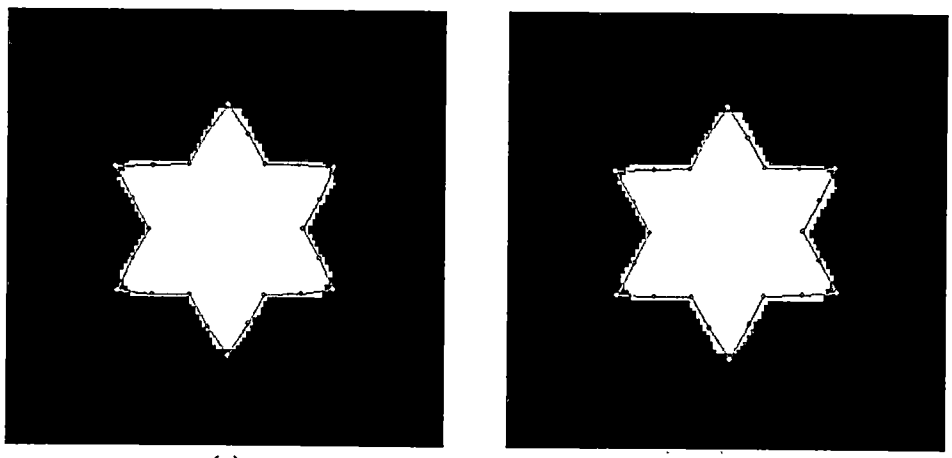
the relative value of the GLM confidence metric increases, indicating a trend toward a better fit to the gray-level profile training data.

We can perform a similar experiment to analyze the impact of changing the LSM weighting factor, w_2. For this experiment we leave the GSM weight equal to zero so that we can isolate the influence of the LSM and GLM on final boundary shape. Starting with an LSM weight of 0 and a GLM weight of 1, the value of the LSM weight relative to the GLM weight is increased. This is accomplished both by increasing w_2 and decreasing w_1. The results in Fig. 5.5 and Fig. 5.6 show how the LSM gains more control of the final boundary shape. In Fig. 5.5 there is little visible change, but note that the GSM and the GLM confidence metrics increase slightly. In Fig. 5.6, note that the LSM confidence met-



(a) (b)

Fig. 5.5 Segmentation results on modified star. Weight parameters are: (a) w_1=0, w_2=0, and w_3=1 and (b) w_1=0, w_2=1, and w_3=1. The resulting confidence values for (a) are c_1=-0.60, c_2=-4.14, and c_3=-6.28 and for (b) are c_1=-0.54, c_2=-3.85, and c_3=-6.82.



(a) (b)

Fig. 5.6 New segmentation results on modified star. Parameters are: (a) w_1=0, w_2=1, and w_3=0.5 and (b) w_1=0, w_2=1, and w_3=0.1. The resulting confidence values for (a) are c_1=-0.24, c_2=-2.04, and c_3=-10.94 and for (b) are c_1=-0.67, c_2=0.24, and c_3=-17.13.

ric, c_2, continues to improve (eventually becoming positive), while the GLM confidence metric decreases.

This experiment illustrates the impact of the PSAM weighting factors on the final boundary shape and position. It also illustrates how the individual confidence metrics can be used as a tool to possibly predict the segmentation result. As will be seen later in this chapter the confidence metrics do not always allow a prediction of the segmentation accuracy, but they can be used to alert the operator to the fact that something unusual has occurred.

5.2 PSAM Results on Synthetic Medical Data

For analysis of the robustness of the PSAM algorithm in the presence of both geometric deformation and noise, a set of synthetic image data was generated. A simulated image of kidney cross sections was created, and then fourteen different geometric distortion were applied to the original image. In addition, different levels of uniform noise were added to each of the original and geometrically distorted images (see Fig. 5.7). The levels define the maximum value of the uniformly distributed noise that is added to each pixel, where each level is set as a percentage of the image depth (256 gray-scale values). In these experiments, four levels of uniform noise distributed over +/-25%, +/-50%, +/-75%, and +/-100% of the image dynamic range (255 levels of gray) were added to each image. The PSAM algorithm was trained only on the noise-free images; the images being tested were held out of the training set to make the results more realistic. The corresponding segmentation results for a subset of these synthetic images are shown in Fig. 5.8. In general, the

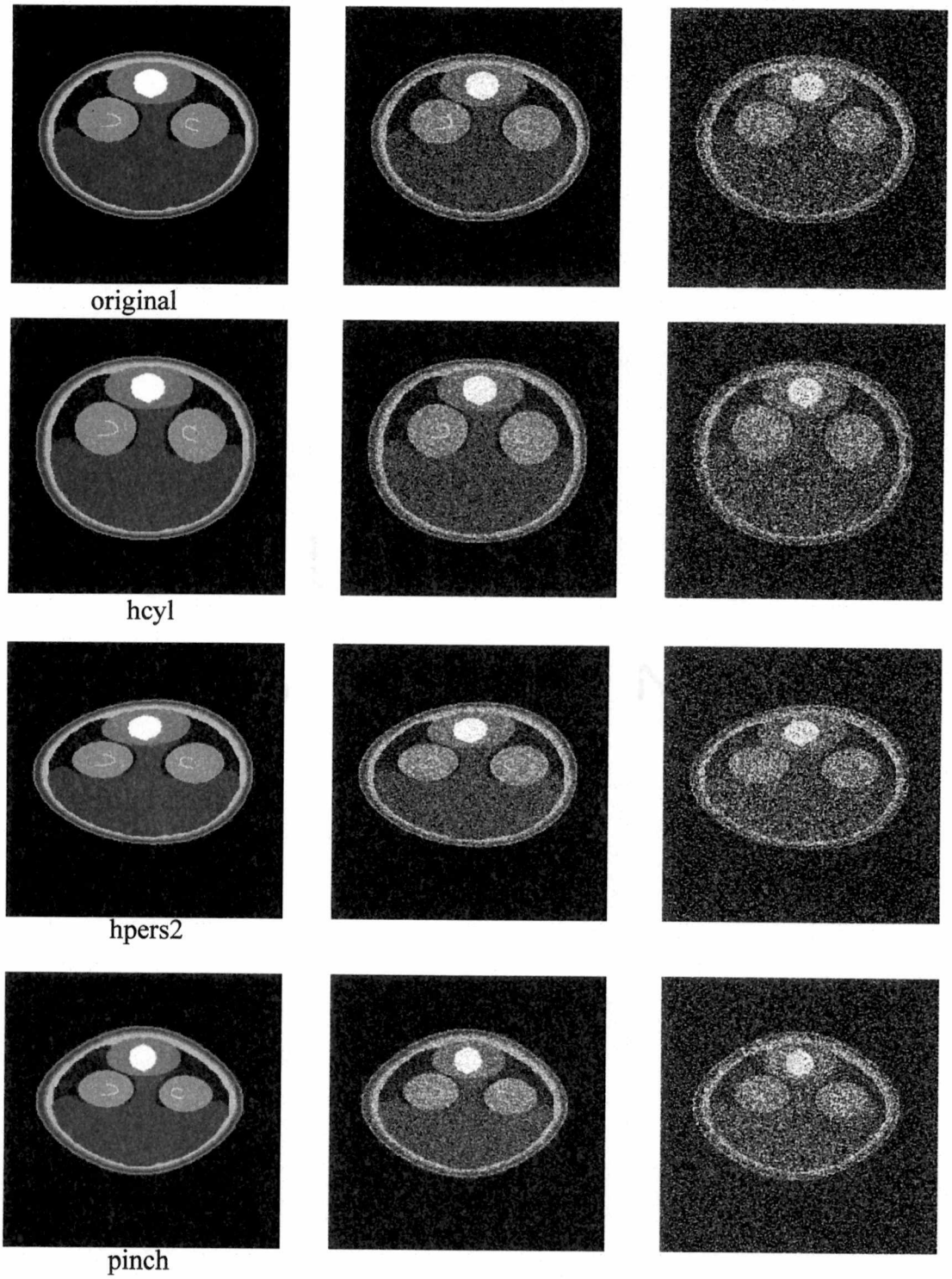


Fig. 5.7 Simulated kidney image with geometric distortion and noise. A variety of distortions (top to bottom) and uniform additive noise (50% middle column, 100% right column) are shown.

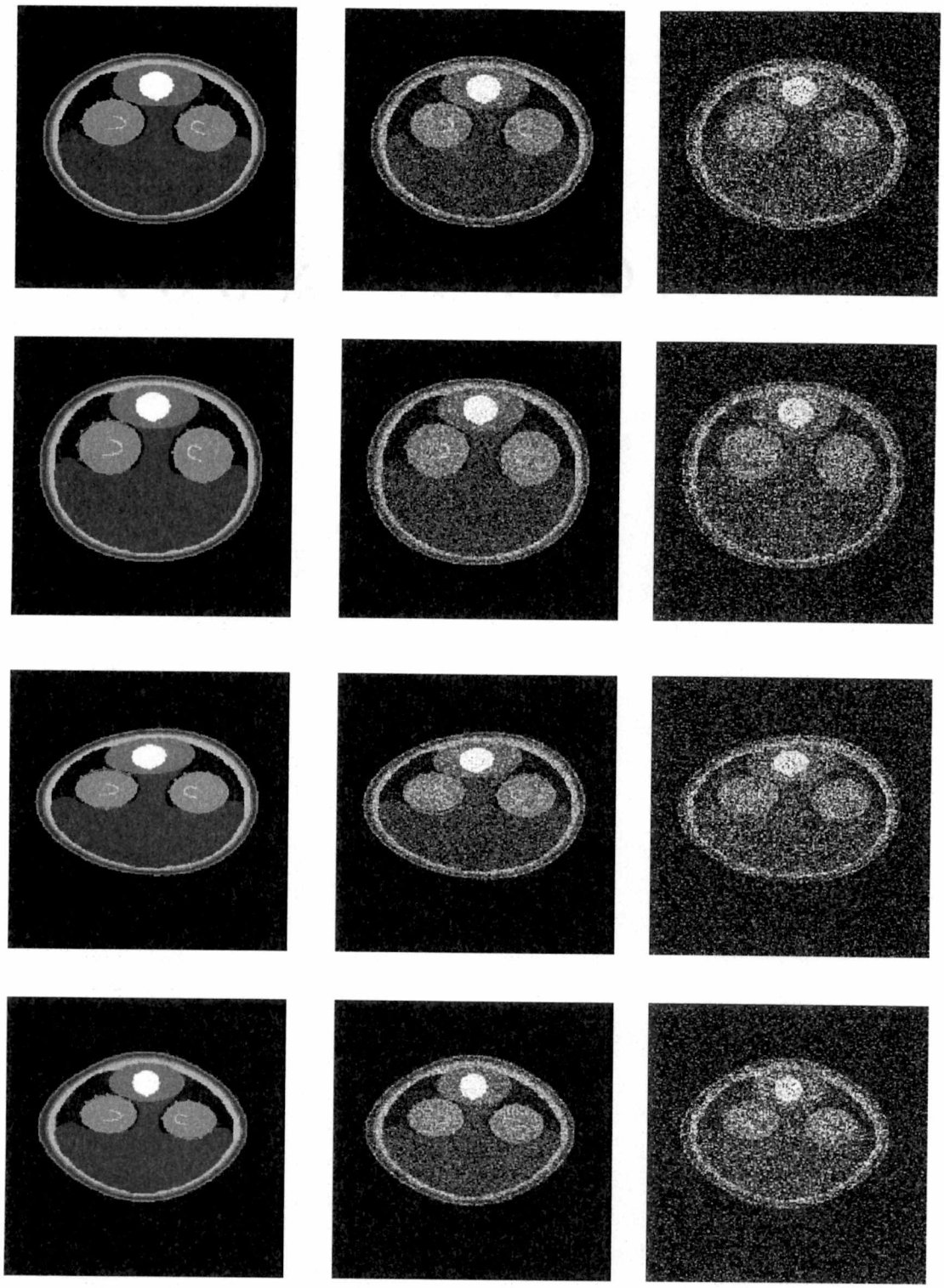


Fig. 5.8 PSAM results with geometric distortion and additive noise. PSAM performed well at noise levels up to 50% and 75% (not pictured) and had some trouble at 100%, particularly with the horizontal perspective transformation image.

PSAM algorithm performed well in the presence of noise and geometric distortion. Only at noise levels of 100% did PSAM begin to have significant problems with segmentation.

5.3 PSAM Results on Medical Image Data

PSAM was evaluated in a real-world situation by using it to segment three sets of medical-image data. All three sets of data were acquired using a micro X-ray CT system built for scanning laboratory mice [78]. The three data sets contained increasingly difficult-to-segment objects and are a good test of the PSAM capabilities. This researcher generated ground truth data for PSAM training by manually placing LPs along the boundaries of the anatomic objects of interest in each training image.[1] PSAM was trained and then applied to all three cases with the weighting parameters set as follows: $w_1 = 1$, $w_2 = 0$, and $w_3 = 1$. In other words, the GSM and the GLM components were weighted with full strength, while the LSM component was not included in the optimization. The effect of the LSM component is addressed in Sec. 5.4. The standard ASM approach was also applied to the same data sets for comparison purposes. For all test cases on both algorithms, a leave-one-out approach was adopted for segmentation performance testing; in other words, the image under test was left out of the training set.

Boundary error information is given for all of the sample data. Boundary error is determined by calculating the distance from each LP in the final boundary result with its corresponding ground truth LP in the training set. The total error is calculated by summing the

1. Although this researcher has substantial experience in accurately delineating anatomic structures within X-ray CT images, future experiments will be performed using ground truth data created by an experienced veterinarian radiologist.

individual error at each LP, and the average error is found by dividing the total error by the number of LPs. This gives the average error on a per-LP basis.

5.3.1 Mouse Skull and Trachea Segmentation

The first data set consisted of 12 transaxial slices through an X-ray CT volume of the skulls of twelve mice. Four of the twelve images are shown in Fig. 5.9. Note the large variation in the pose of each of the animals. This pose variation can make segmentation difficult. One sample image is more clearly illustrated in Fig. 5.10. This particular slice through the skull shows the ear canals and trachea of the animal. A two-boundary model was used during training to delineate the skull and ear canals (open boundary) and the trachea (closed boundary). The same training data were used to train both the PSAM and the ASM algorithms.

This data set is considered to be moderately difficult because the pose of the animal in the training data is changing substantially. The edge information provided by the skull and trachea is fairly strong, however, making the segmentation problem manageable. The plot in Fig. 5.11 shows the total boundary error for each of the twelve images tested. The PSAM algorithm outperforms ASM in 9 of the 12 images and is significantly outperformed by ASM in only one case (image 7). A comparison of the PSAM and ASM segmentation results for image 10 are shown in Fig. 5.12. In this example the average pixel boundary error per LP is 2.3 pixels for PSAM segmentation, compared to 4.2 pixels for ASM segmentation. Comparing the sum of the total boundary error across all LPs for all of the 12 test images, the total error for ASM was 790 pixels versus 632 pixels for PSAM. The ASM total boundary error on this test set was 25% higher than the PSAM error.

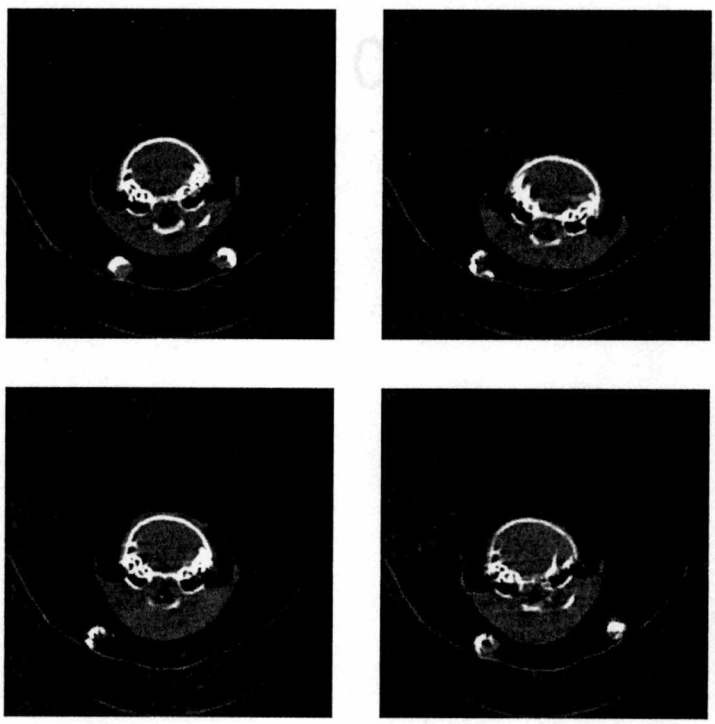


Fig. 5.9 Sample mouse skull images from first training set.

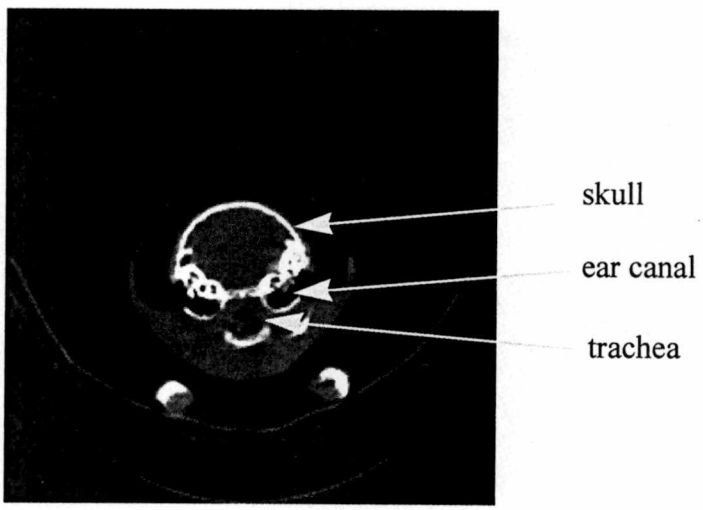


Fig. 5.10 Transaxial slice through an X-ray CT volume of a mouse skull. The indentations are where the ear canal enters the skull. A cross section of the trachea is also visible.

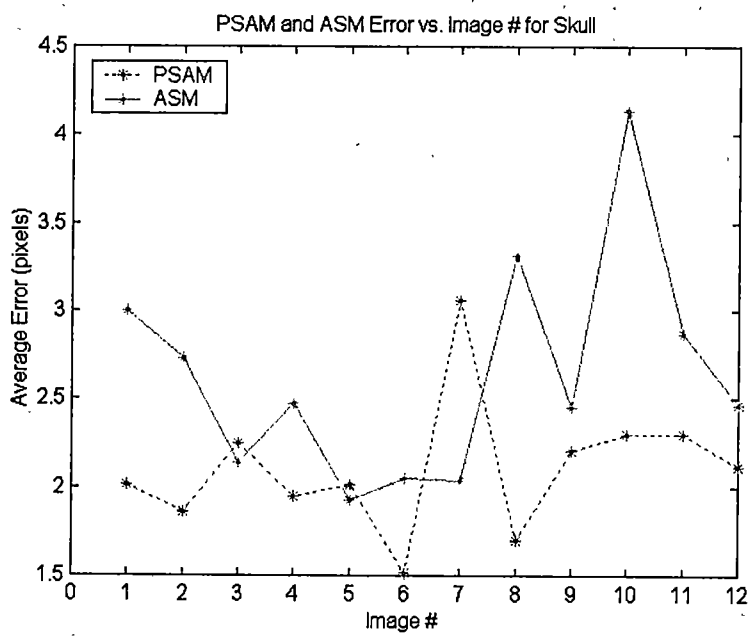
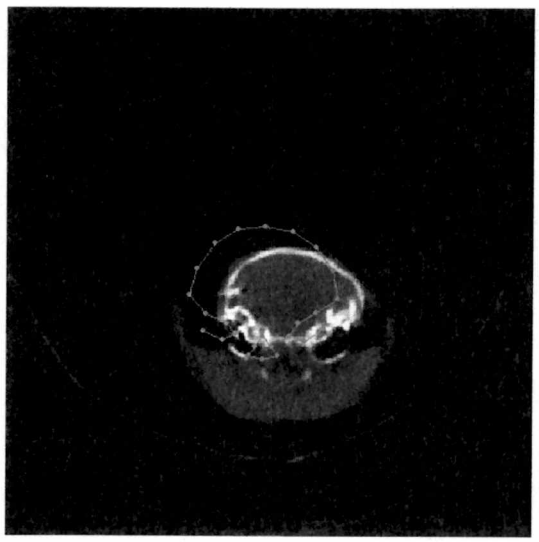
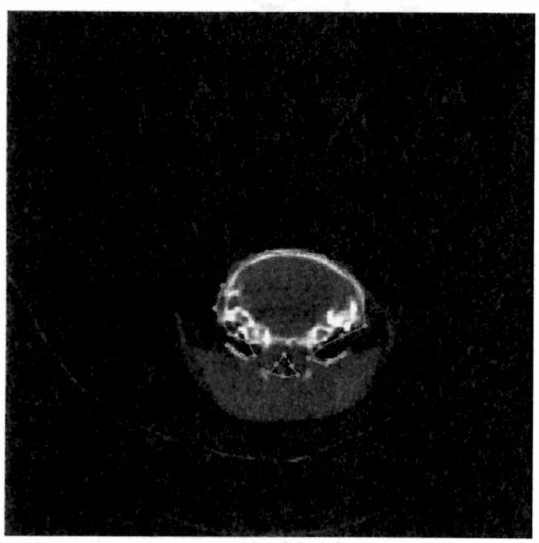


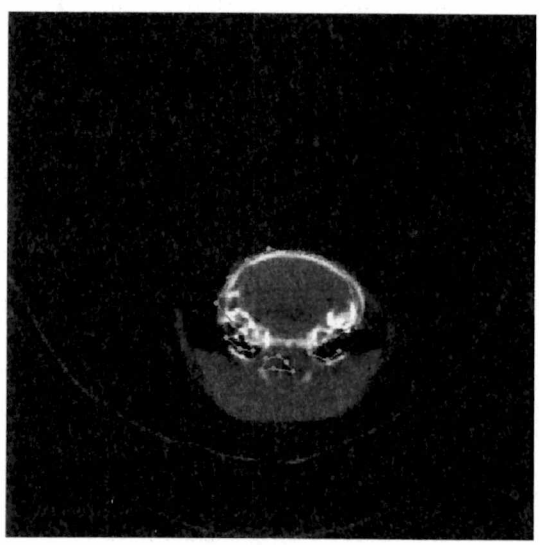
Fig. 5.11 Average boundary error for the skull images. Note that the PSAM algorithm outperforms ASM in 9 out of the 12 images, and is outperformed significantly by ASM in only 1 case (image 7).



(a)

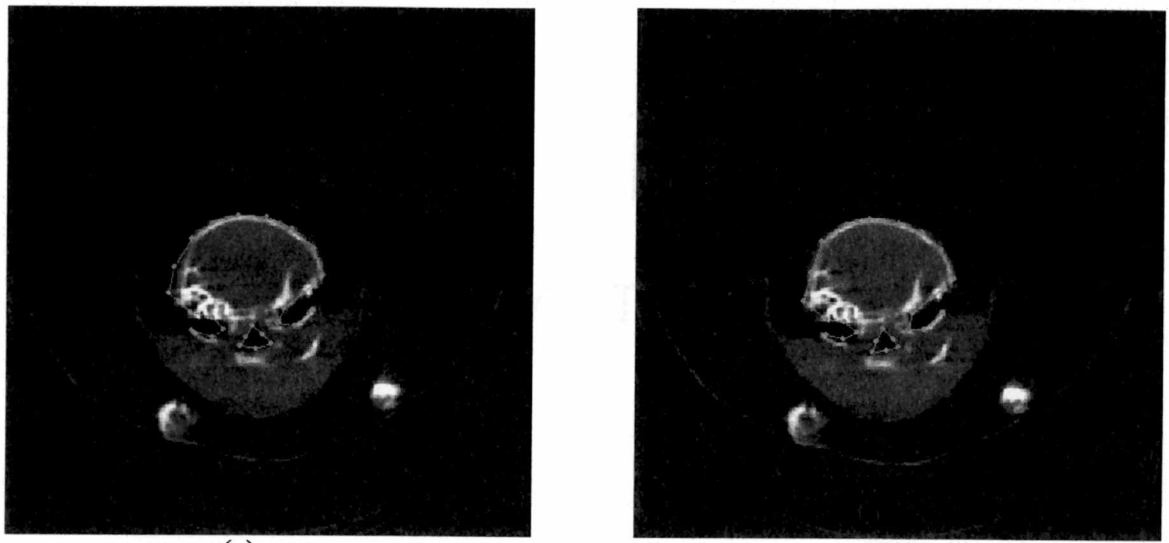


(b)



(c)

Fig. 5.12 Skull sample #10 segmentation: (a) initial position of model; (b) result obtained using PSAM; (c) result obtained using ASM. Note in particular the improved PSAM segmentation of the trachea and left ear canal.



(a) (b)

Fig. 5.13 PSAM segmentation result for image 7. Note in image (a) that the left side of the skull boundary was missed. The improved result in (b) was obtained after reducing the influence of the GSM weight, w_1, from 1.0 to 0.3.

Note, in Fig. 5.11, the PSAM segmentation was outperformed by ASM on image 7. The PSAM segmentation error was due to a variation in the shape of the skull on the left side of image 7. The left side of the skull does not extend out as far as it does in the other 11 images in the training set (see Fig. 5.13). This error was corrected by reducing the GSM weight, w_1, for the PSAM algorithm from 1.0 to 0.3. Note how the weaker GSM allowed the boundary to move in and capture the left side of the skull in Fig. 5.13(b).

Finally, it is also interesting to analyze the relationship between the boundary error and the confidence metrics. The most potentially useful relationship appears to be between the GLM confidence metric, c_3, and the average boundary error. Recall that the GLM confidence metric indicates how well the final gray-level profiles normal to each LP fit within the corresponding distribution of LP gray-level profiles as defined by the training set. If

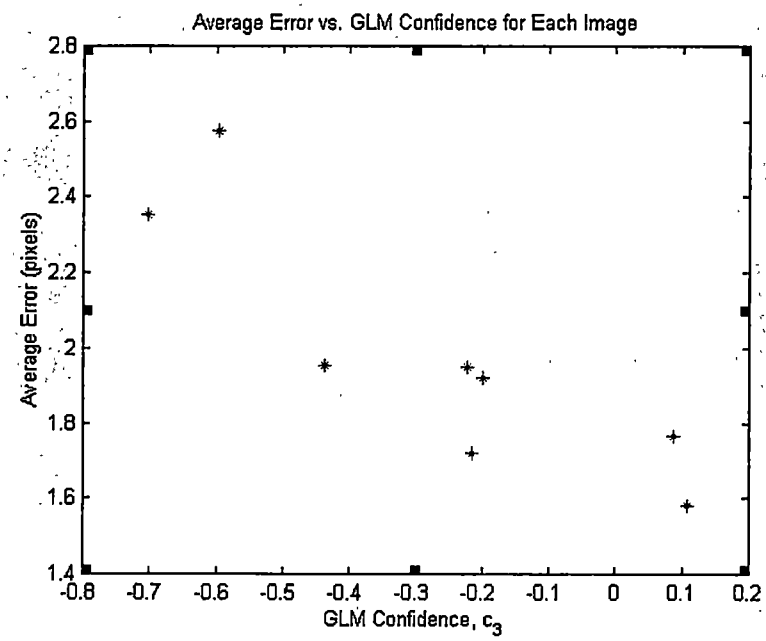


Fig. 5.14 Plot of average boundary error vs GLM confidence for skull images. Note the approximate negative correlation between the two.

for example, there are strong edges in the training set, there will be strong gradients in the training profiles. If the final boundary profiles do not contain strong gradients, then the value of c_3 will be low, indicating the possibility of poor segmentation. The plot in Fig. 5.14 shows a potential negative correlation between the average LP boundary error and the GLM confidence metric, c_3. Based on this plot, a lower GLM confidence roughly corresponds to higher segmentation error. GLM confidence could be monitored in this case so that a low value could flag a potential problem with the segmentation result.

5.3.2 Mouse Heart and Lung Segmentation

The second data set consisted of 12 transaxial X-ray CT slices through the chest of 12 mice revealing a view of the heart and lung anatomy. Four of the 12 images are shown in

Fig. 5.15. Note the large variation in the pose of each of the animals as well as the variation in lung size and appearance. The two lower images contain lungs that have a larger amount of tissue and/or fluid within them, causing them to appear lighter in color (air is black on an X-ray image). These pose, size, and appearance variations make this a more difficult segmentation problem than the previous case. One sample image is more clearly illustrated in Fig. 5.16, which shows the lungs and heart of the animal. A three-boundary model was used during training to delineate the lungs (two closed boundaries) and the lower heart (an open boundary). As with the skull data experiment, the same training data was used to train both the PSAM and ASM algorithms.

The plot in Fig. 5.17 shows the average boundary error for each of the twelve images tested. With this data set, the PSAM algorithm outperformed ASM in 9 of the 12 cases (with a slightly lower error for images 4 and 12), and was outperformed by ASM in three cases (images 8, 9, and 11).The PSAM and ASM segmentation results for image 3 are shown in Fig. 5.18. In the example shown, PSAM generated a more accurate segmentation of both lungs. In addition, the lower boundary of the heart was more accurately delineated. Although ASM had a lower error for 3 of the 12 images, a qualitative comparison of the segmentation results reveals comparable performance. For example, compare the PSAM versus ASM segmentation of image 8 as shown in Fig. 5.19. Qualitatively, the PSAM result appears to be nearly as good as, if not better than, the ASM result. Comparing the sum of the total boundary error across all LPs for all of the 12 test images, the total error for ASM was 1612 pixels versus 1322 pixels for PSAM. The ASM total boundary error on this test set was 22% higher than the PSAM error.

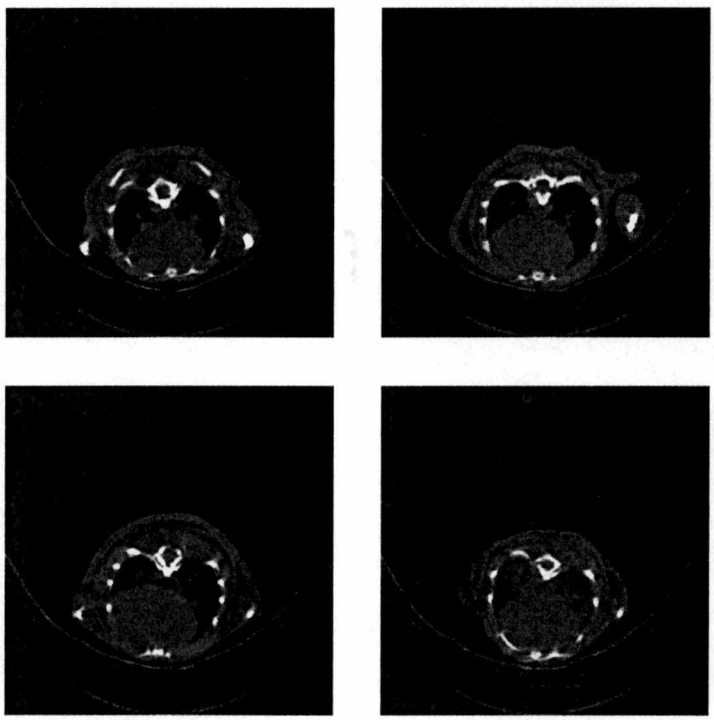


Fig. 5.15 Example heart and lung images from second training set. Note the variation in pose, shape, and intensity of the heart and lungs.

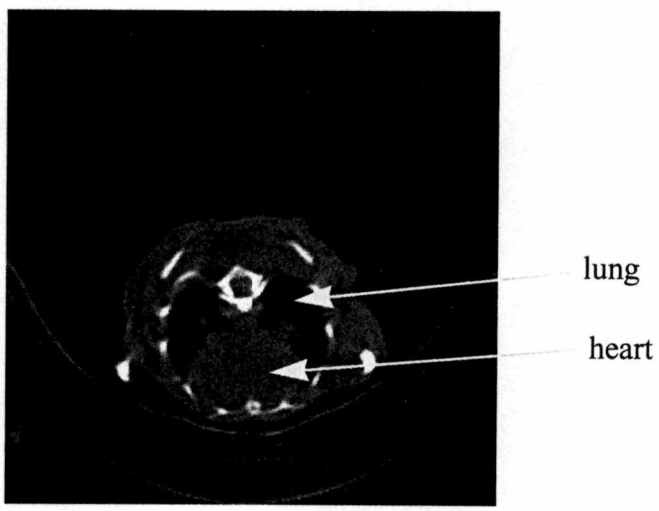


Fig. 5.16 Transaxial slice through an x-ray CT volume of a mouse chest. The two dark objects are the air-filled lungs, and the object between them is the heart.

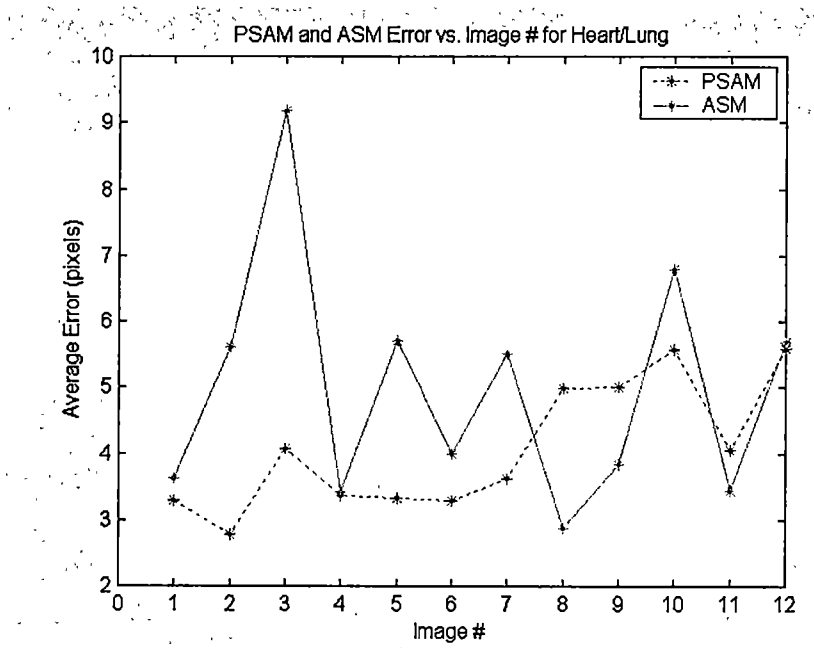
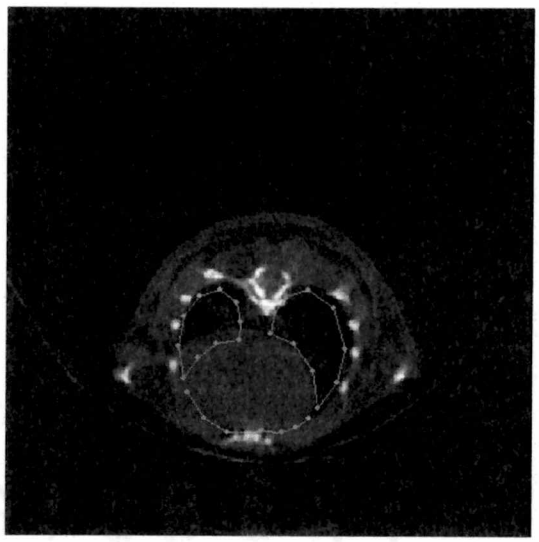
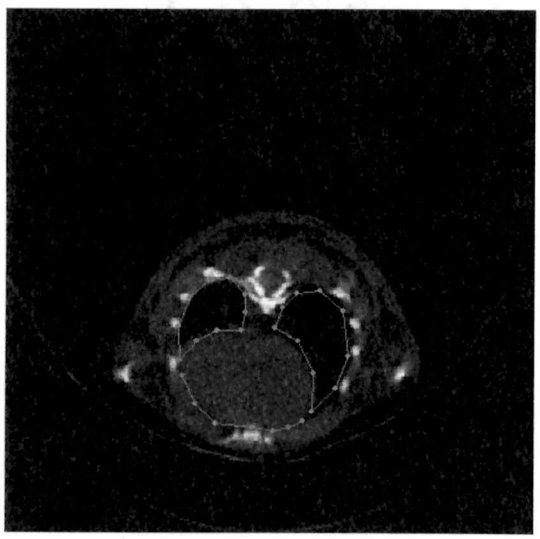


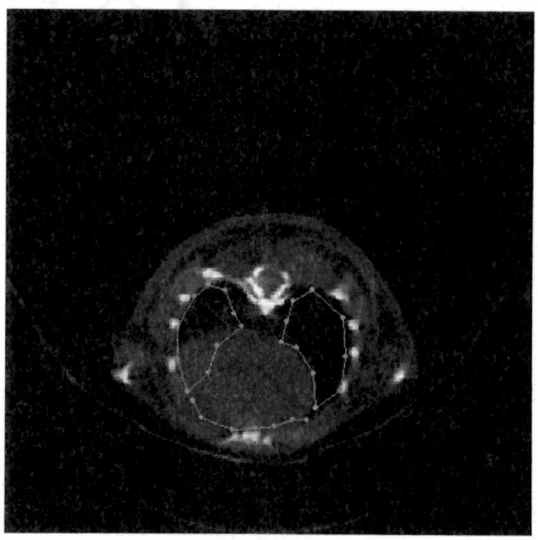
Fig. 5.17 Average boundary error for the heart and lung images. The PSAM boundary error is lower than ASM boundary error in 9 of the 12 images (only slightly lower for images 4 and 12). PSAM has higher error than ASM in three cases (images 8, 9, and 11).



(a)

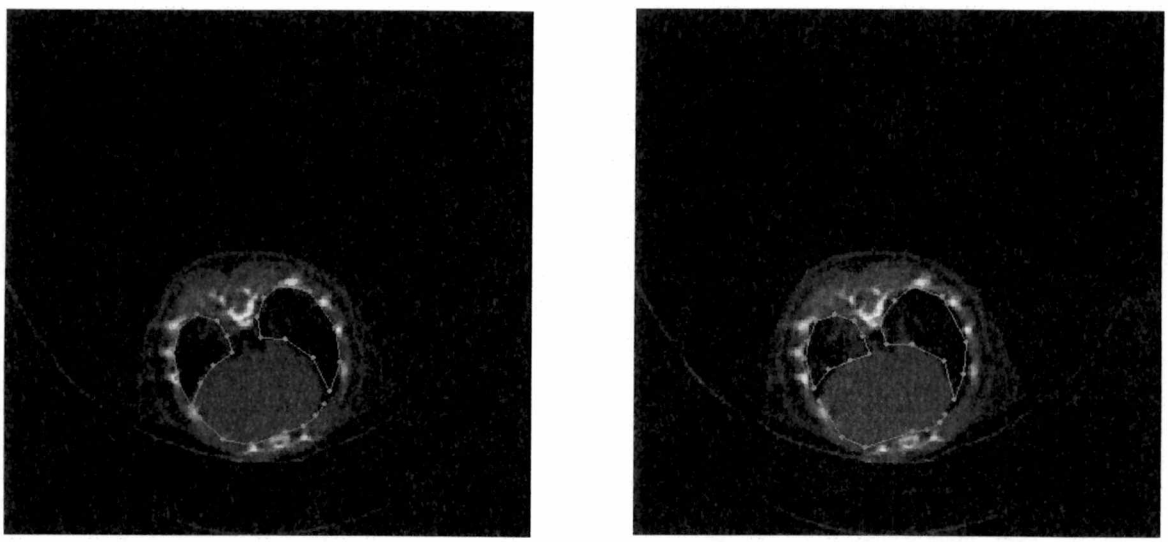


(b)



(c)

Fig. 5.18 Heart and lung segmentation result for image 3: (a) initial position of model; (b) result obtained using PSAM; (c) result obtained using ASM.



(a) (b)

Fig. 5.19 Segmentation result using (a) PSAM and (b) ASM for image 8. qualitatively, the segmentation accuracies are comparable.

Once again, it is interesting to analyze the relationship between the boundary error and the GLM confidence metric, c_3. The plot in Fig. 5.20 shows the average LP boundary error versus the GLM confidence metric, c_3. In this case, there is not the same type of negative correlation as we saw in the skull data set. We do see one extreme case (upper left) where the average error is very high (16.8 pixels), corresponding to a relatively low confidence metric (0.17), but there does not appear to be an overall correlation between c_3 and the boundary segmentation error similar to that which was seen on the skull image data.

5.3.3 Mouse Kidney and Spine Segmentation

The third data set consists of 14 transaxial X-ray CT slices through the abdomen of 14 mice revealing a view of the kidneys. Four of the 14 images are shown in Fig. 5.21. Again,

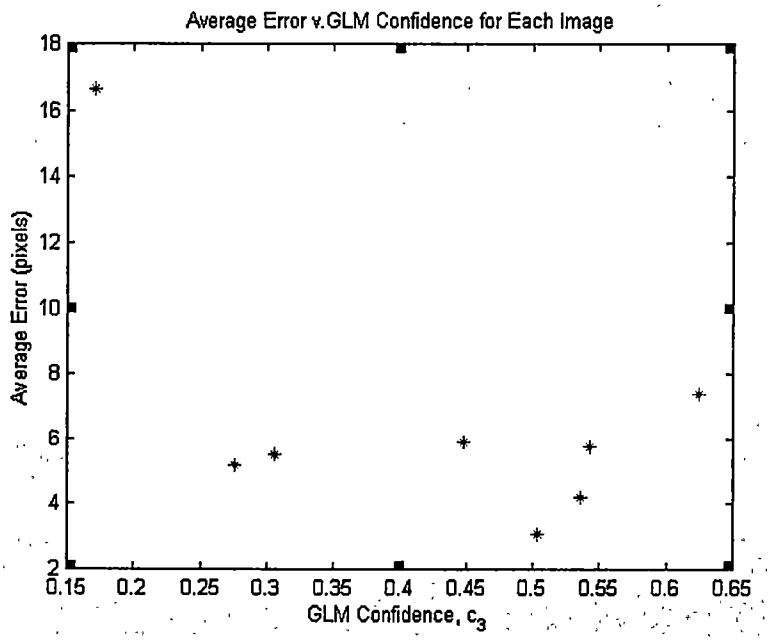


Fig. 5.20 Average error vs GLM confidence for heart and lung images. Note the upper-left point where the error suddenly jumps to a low value of the GLM confidence.

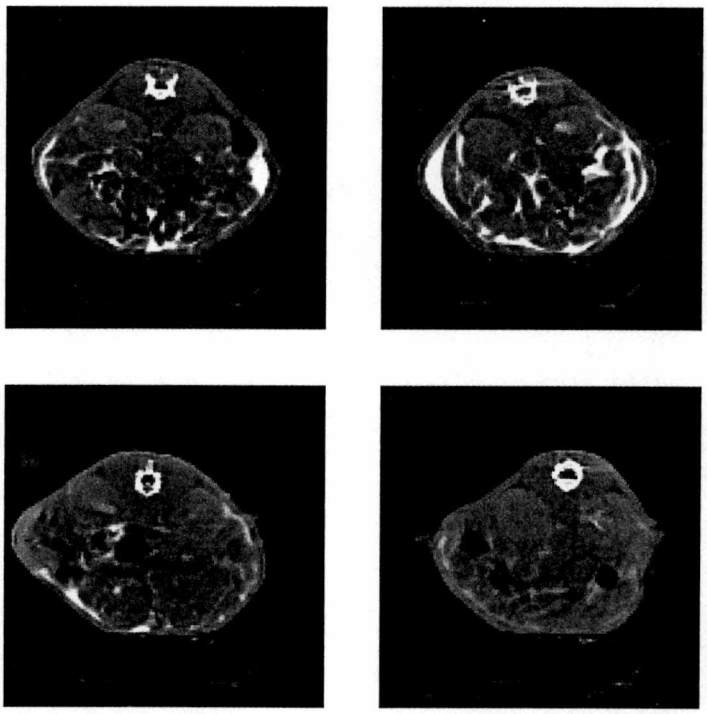


Fig. 5.21 Mouse kidney images from third training set. Note the large variations in pose, shape, intensity, and background complexity.

note the large variation in the pose of each of the animals, the variation in kidney size and appearance, and the complexity and variability of the kidney backgrounds. The white spots in the upper two images are concentrations of X-ray contrast agent. The two lower images contain much less contrast agent compared to the two upper images. These pose, size and appearance variation of the kidneys along with the complexity and variability of the backgrounds make this the most challenging segmentation problem of the three data sets. One sample image is more clearly illustrated in Fig. 5.22. This slice through the abdomen shows the kidneys and spine of the animal as well as some of the intestines in the lower abdomen. A three-boundary model was used during training to delineate the kidneys (two closed boundaries) and the spine (an open boundary). As with the two previous data sets, the same training data was used to train both the PSAM and standard ASM algorithms.

The plot in Fig. 5.23 shows the total boundary error for each of the 14 kidneys images tested. With this data set, the PSAM algorithm outperformed ASM in 10 of the 14 cases. In the first four of those cases (images 1 – 4) PSAM has much lower boundary errors than ASM. ASM error is lower than PSAM error in 4 of the 14 cases, but the difference in error is relatively small. Two PSAM versus ASM segmentation results are shown in Fig. 5.24 and Fig. 5.25. In the first example, PSAM generated a more accurate segmentation of both kidneys and the spine. The second example (Fig. 5.25) is the same one that was presented as a particularly problematic case for ASM in Sec. 3.5. There, it was demonstrated explicitly that the independent optimization of the GSM and the GLM led to a poor segmentation of the left kidney. Use of the new PSAM approach led to near-perfect segmentation of

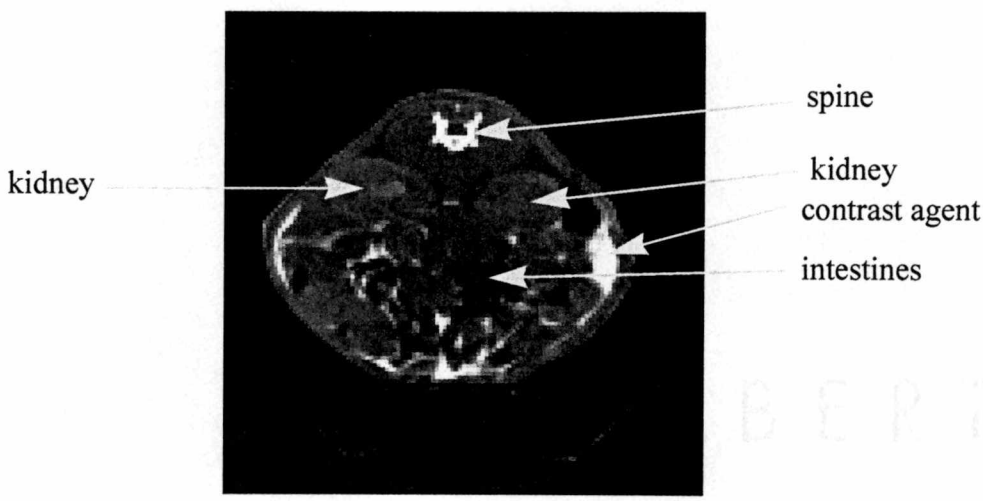


Fig. 5.22 Transaxial slice through an X-ray CT volume of a mouse abdomen. The two elliptical objects are the mouse kidneys, and the white object above them is one of the vertebra in the spine.

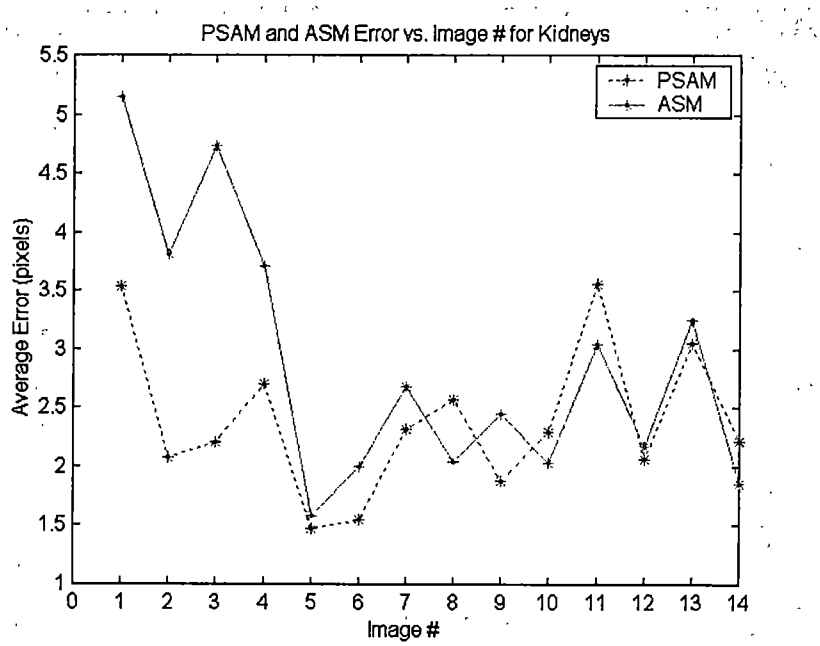
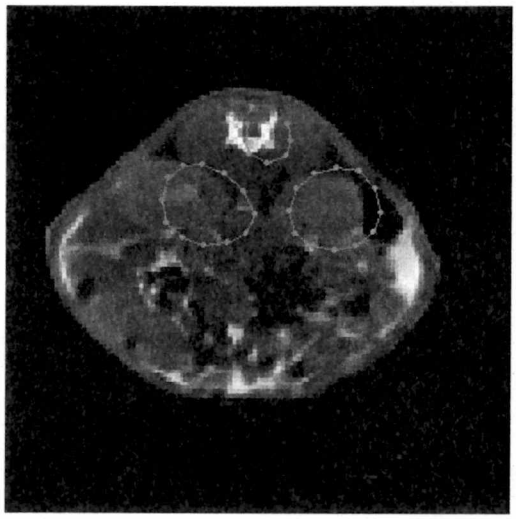
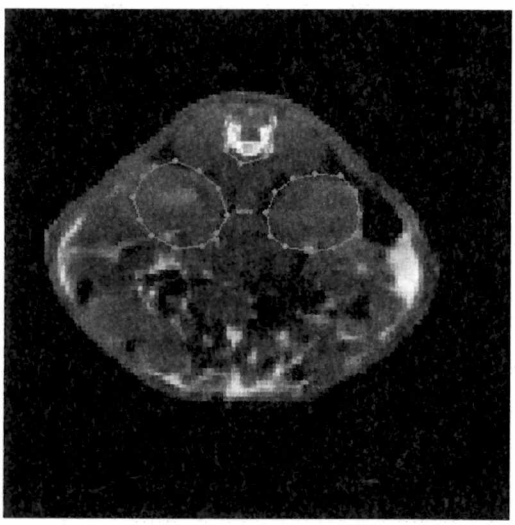


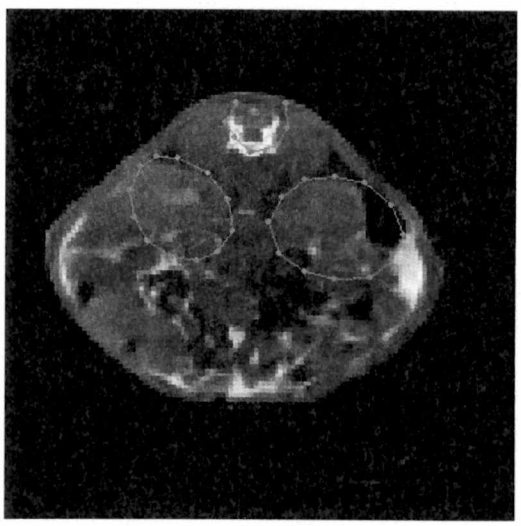
Fig. 5.23 Total boundary error for the kidney and spine images. The PSAM algorithm outperforms ASM in 10 of the 14 images and is slightly outperformed by ASM in the remaining 4 cases.



(a)

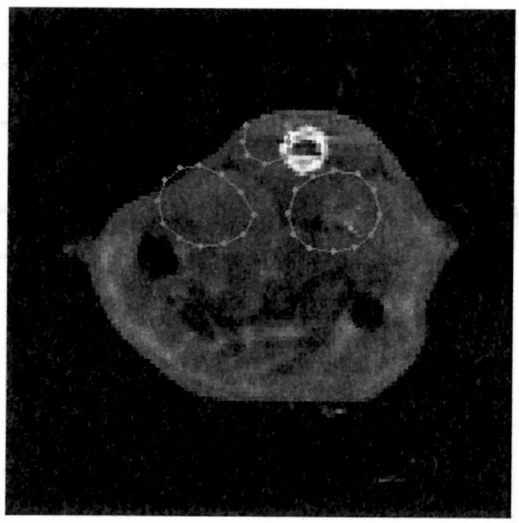


(b)

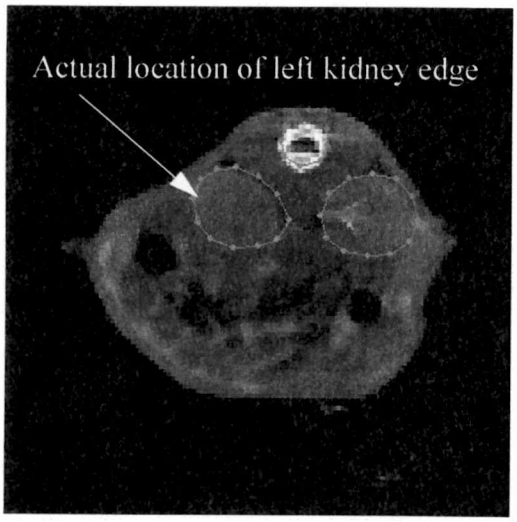


(c)

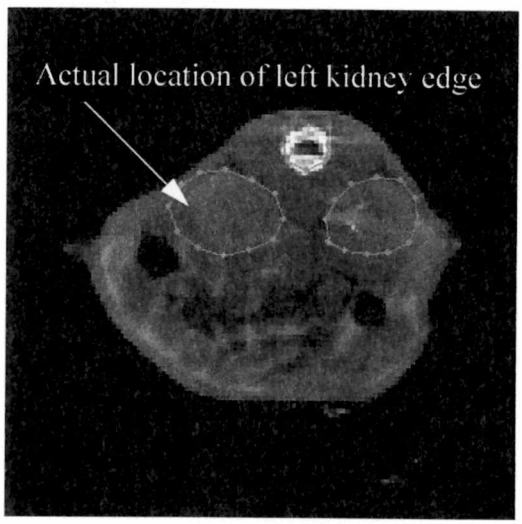
Fig. 5.24 Kidney and spine segmentation result on image 1: (a) initial position of model; (b) result obtained using PSAM; (c) result obtained using ASM.



(a)



(b)



(c)

Fig. 5.25 Kidney and spine segmentation result on image 12: (a) initial position of model; (b) result obtained using PSAM; (c) result obtained using ASM. Because the kidney edge is very faint on a printed version of the image, it has been marked with an arrow for convenience.

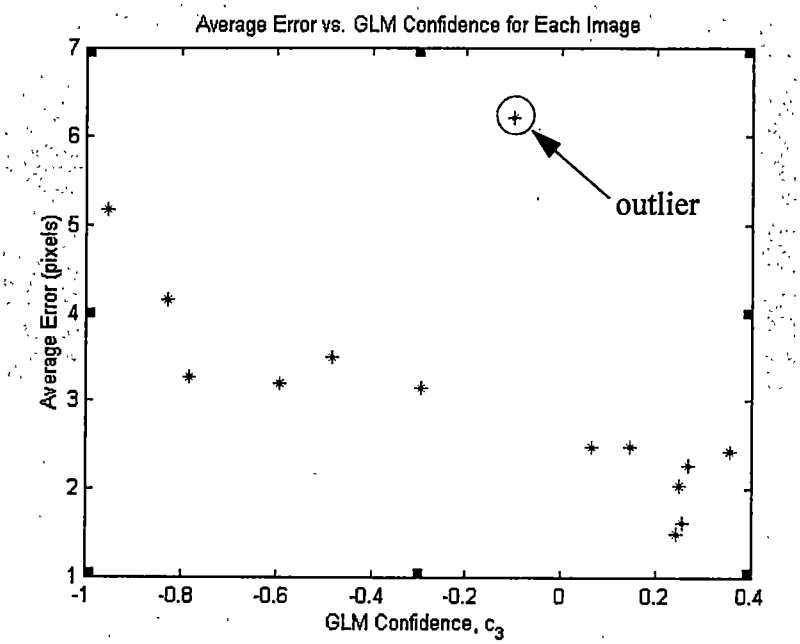


Fig. 5.26 Average error vs GLM confidence for kidney and spine images. In general, lower confidence corresponds (with the exception of the outlier) to higher segmentation error.

both kidneys. Comparing the sum of the total boundary error across all LPs for all of the 14 test images, the total error for ASM was 1256 pixels versus 1038 pixels for PSAM. The ASM total boundary error on this test set was 21% higher than the PSAM error.

Once again, it is interesting to analyze the relationship between the boundary error and the GLM confidence metric, c_3. The plot in Fig. 5.26 shows the average LP boundary error versus the GLM confidence metric, c_3. In this case, there is again (as in the skull data set) an approximate negative correlation between c_3 and the boundary segmentation error. For the case of kidney and spine segmentation shown here, the GLM confidence metric could be used as an approximate predictor of segmentation accuracy.

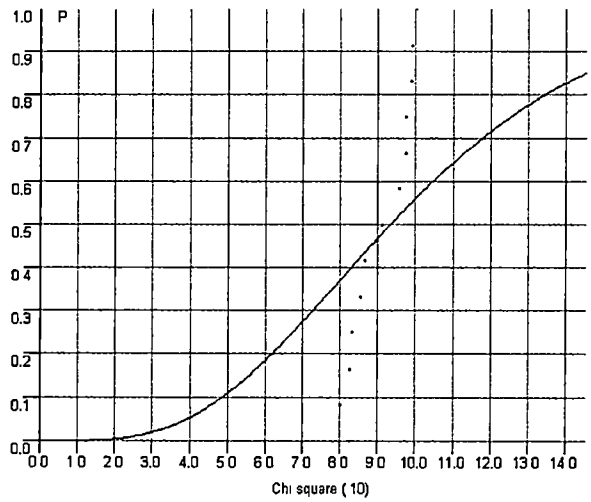
5.3.4 Gaussian Distribution Goodness-of-Fit Tests

During the theoretical development of the PSAM algorithm, multivariate Gaussian functions were used to model both the shape variation and the gray-level profile variation for each LP. Although researchers in this area of statistical shape models often make similar assumptions, the validity of these assumptions is rarely tested. Two separate statistical techniques were used to test the validity of these Gaussian assumptions for the three medical image data sets just presented. These statistical tests were performed with a relatively small number of samples (12 for the skull data, 12 for the heart and lung data, and 14 for the kidney data); therefore, the results must be interpreted with a degree of caution.

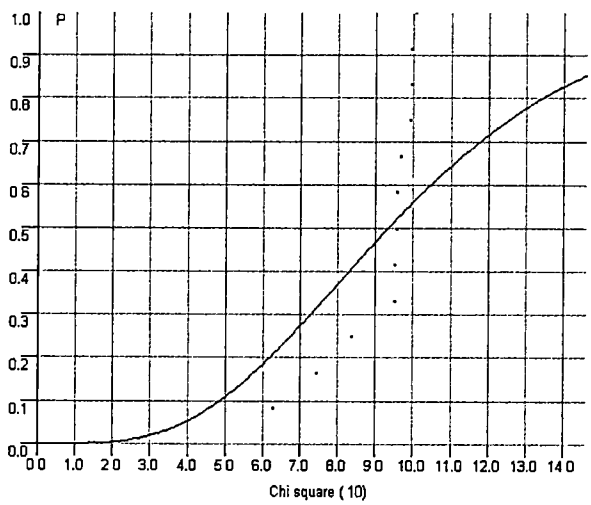
Two well-known multivariate tests were used to determine if the shape and gray-level data in the training set come from a specific (Gaussian, in this case) distribution. Both tests—the Kolmogorov-Smirnov (K-S) goodness-of-fit test [14] and the Anderson-Darling (A-D) test [88]—are based on the fact that for a multivariate Gaussian distribution, the distribution of squared Mahalanobis distances as measured from the training data will follow a chi-square distribution with k degrees of freedom, where k is the number of variables being modeled. The two tests differ in how they measure the difference between the ideal (chi-square) distribution and the empirical (Mahalanobis distance) distribution. The Anderson-Darling test is a more recent variation of the Kolmogorov-Smirnov test and is considered to be both more stringent and more accurate.

The result of these tests applied to the shape vectors, v, as measured from the three different medical image training sets are shown in Fig. 5.27 and Table 5.1. The shape vector

(a)



(b)



(c)

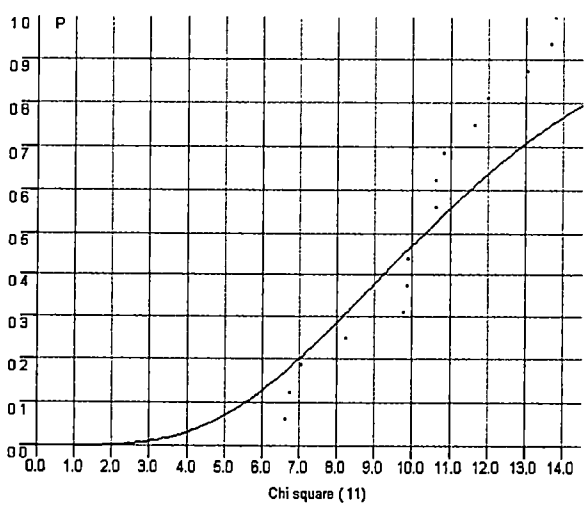


Fig. 5.27 Probability distributions for multivariate Gaussian shape data. Plots show empirical (dots) and theoretical (curve) distributions for skull data (a), heart and lung data (b) and kidney/spine data (c).

Table 5.1: Results of K-S and A-D Tests on Training Shape Vectors, v

Image Set	K-S Test Result	A-D Test Result
Skull	Fail	Fail
Heart/Lung	Marginal	Fail
Kidneys/Spine	Pass	Fail

distributions pass the K-S test on two of the three data sets (one passes only marginally), and that they all fail the more stringent A-D test. This is an interesting result given that the PSAM algorithm performs well on all three data sets. Recall that the PSAM algorithm also optimizes with respect to the gray-level information, so we will now examine how well the gray-level profile distributions for each LP fit the assumed multivariate Gaussian distribution.

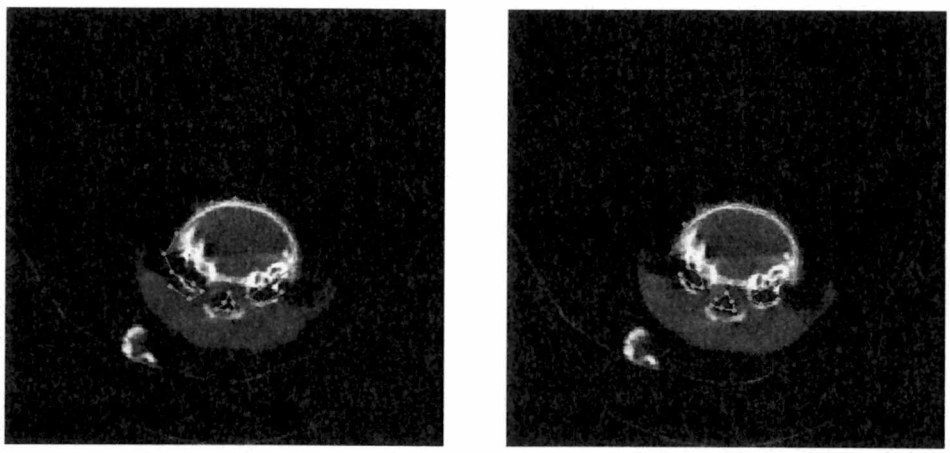
The results of the K-S and A-D tests as applied to the gray-level profile distributions indicate a better fit (as compared to the shape vectors) to a multivariate Gaussian distribution. For the K-S test, 100% of the gray-level profile distributions pass the multivariate Gaussian test. For the more stringent A-D test, 63 to 68% of the gray-level profile distributions were found to fit a multivariate Gaussian distribution. The lowest percentage corresponds to the heart and lung data set. This is the same data set in which the Gaussian-based GLM confidence metric, c_3, did not correlate reasonably well with the PSAM boundary error (see Fig. 5.20).

5.4 LSM Experiments

Many sets of experiments were performed on the synthetic and real data to test the effectiveness of the new LSM introduced in Chap. 4. Some of the tests on synthetic data

(star images) have already been presented at the beginning of this chapter. These showed that the LSM influences the solution in the manner expected. The tests carried out using LSM on real data have shown that when the LSM is used in conjunction with the GSM, both weighted at full strength ($w_1 = 1$, $w_2 = 1$), only in a few instances is segmentation accuracy improved over using the GSM alone. One of those cases is illustrated in Fig. 5.28, where two LSMs (size = 5 LPs) were included, one around each ear canal. The GLM weight, w_3, is was set to 1 for all experiments described here. In the majority of cases, the inclusion of the LSM along with the full influence of the GSM (w_1=1) did not significantly affect the segmentation result, but in the cases where the LSM did influence the result, it improved accuracy. The LSM seems to have a more positive impact in situations where the GSM may be ill-defined, as described next.

In segmentation applications where the GSM is ineffective because of a combination of few training samples and substantial global boundary shape variability, using LSMs



(a) (b)

Fig. 5.28 Effect of LSM influence on segmentation on mouse skull. (a) No LSM influence ($w_2 = 0$) and (b) full LSM influence ($w_2 = 1$).

independent of the GSM can be beneficial. A good example of this situation is an ivy leaf data set. A few examples of this data set are shown in Fig. 5.29. There is a large degree of variability in the overall global shape, but the local shapes of the tips of the leaves and the corners between leaf tips are more predictable and can be modeled effectively using LSMs. The image in Fig. 5.30 shows where the centers of six LSMs were defined during training. Each of the six LSMs included three LPs. The PSAM algorithm was used to segment the ivy leaf images first using the GSM and the GLM (no LSM), and second using only the LSM and the GLM (no GSM). An example of the results is shown in Fig. 5.31. Note the improved segmentation result using LSMs instead of the GSM, particularly around the areas where the LSM was defined. This improvement was typical for most leaf images in the training set.

Recall from the development of the shape model in Sec. 4.1.2 that an LSM reward term was added to the shape component of the objective function. Simply adding this term implied an assumption of independence between the LSMs and GSM. The quality of this assumption depends in part on the size of the LSM (which is a subset of the GSM) relative

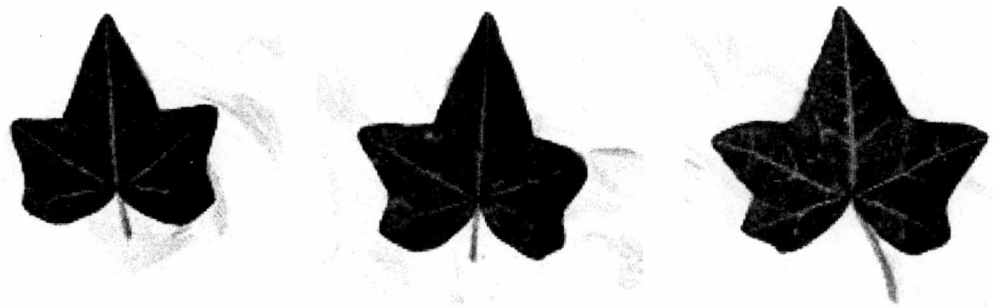


Fig. 5.29 Three examples from the ivy leaf data set. Note the large variations in global shape of the leaves.

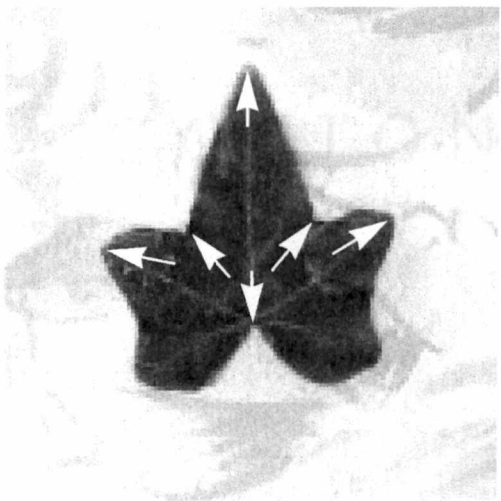
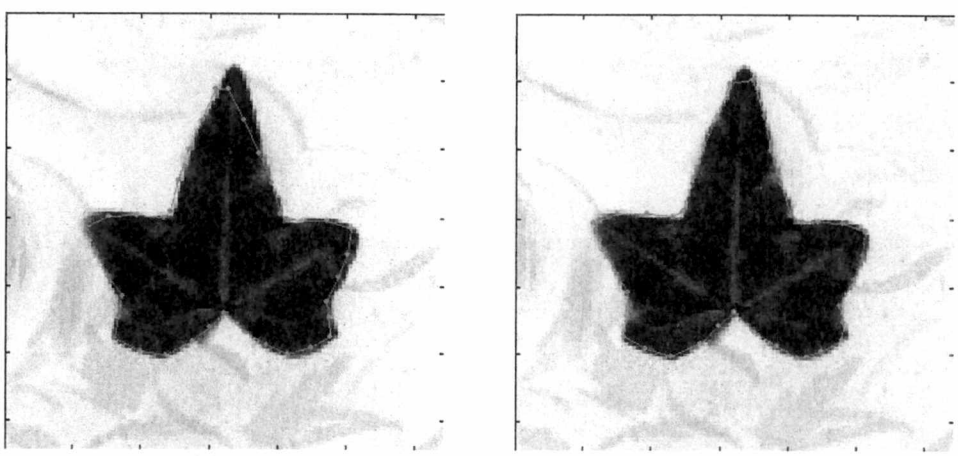


Fig. 5.30 Ivy leaf example. Arrows show the location of centers for six LSMs



(a) (b)

Fig. 5.31 Samples of PSAM segmentation on ivy leaf images. The image in (a) was segmented with $w_1 = 1$, $w_2 = 0$, and $w_3 = 1$ (no LSM). The image in (b) was segmented with $w_1 = 0$, $w_2 = 1$, and $w_3 = 1$ (no GSM).

to the GSM. If the LSM is small enough compared to the GSM, this independence assumption will approximately hold. As the LSM size approaches that of the GSM, the independence assumption falls apart. In practice, there may be a level of dependence between the GSM and LSM that needs to be included in the theoretical development of the PSAM approach. It has been demonstrated (e.g., Fig. 5.28) that including LSM influence with the GSM during optimization will in some cases increase the boundary segmentation accuracy. It is also clear from experimentation that the inclusion of the LSMs has a larger and more positive impact if done independently of any GSM influence (e.g., Fig. 5.31). This last observation supports the fact that there is a level of dependence between the LSM and the GSM that probably needs to be theoretically developed in future research for maximum effectiveness.

5.5 Experimental Results Summary

Experiments with synthetic images showed that the influence of the three model components (GSM, LSM, and GLM) can be controlled through setting of the three weight parameters w_1, w_2, and w_3. The effect of this procedure was shown in the LP boundary segmentation results as well as in the confidence metrics for each component. Experiments using real data demonstrated that the PSAM algorithm is superior to the standard ASM approach in terms of segmentation accuracy. It was also shown that in some cases (two out of the three medical cases) the GLM confidence metric may be effectively used as a predictor of segmentation accuracy. It was also demonstrated that in spite of the fact that the Gaussian distribution assumption is not ideal for the anatomic shape variations in the medical images, PSAM was able to perform the object segmentation quite well. This is

probably due, at least in part, to the fact that the gray-level profiles were shown to follow a multivariate Gaussian distribution in most instances. Finally, it was shown that the use of LSMs in conjunction with the GSM will generate more accurate segmentation for a few cases. However, the LSM has more influence if it can be used independently from the GSM. Future work needs to be done to further assess the effectiveness of the LSMs and to incorporate them more fully into the theoretical development of the PSAM algorithm.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter summarizes PSAM development and segmentation performance. It concludes by describing opportunities for future work based on this research.

6.1 PSAM Development and Performance

The new probabilistic shape and appearance modeling (PSAM) algorithm has many advantages over existing boundary-finding techniques. The PSAM technique is based on compound Bayesian decision theory, and the formulation is general enough that can be used as a starting point to derive some of the other previously developed probabilistic boundary-finding techniques. The motivation for developing this new PSAM algorithm arose from a need to segment semirigid objects from complex backgrounds in cases where the object might have faint and/or missing edge information, as is common in medical imaging applications.

The PSAM algorithm contains three specific model components: (1) a global shape model (GSM), (2) a local shape model (LSM), and (3) a gray-level model (GLM). The GLM formulation is flexible enough to allow a variety of image features to be extracted, but the image feature on which this research focused was gradient-based gray-level profiles normal to the boundary through each LP. All three of the PSAM components are optimized simultaneously when performing boundary searches within new images. PSAM is formulated in a convenient manner so that the influence of each of these components on

the final boundary position can be controlled by the system operator through adjustments of individual model component weight parameters. This formulation allows the same PSAM algorithm to be used in applications with predictable global shape and relatively poor object edge strength, as well as applications where global shape is unpredictable but object edges are prominent and/or LSMs are applicable.

The PSAM algorithm formulation provides confidence metrics for each of the three model components to indicate how well each PSAM component of the final boundary fits within the distribution of each component as derived from the training data. These confidence metrics can be monitored to alert an operator to any boundary results in which one or more model components was found to be "out of bounds" relative to the training data. Furthermore, as was demonstrated in this research, the GLM confidence metric can be used a a predictor of segmentation accuracy for some applications.

The performance of the PSAM algorithm is remarkable for applications of varying degrees of difficulty. In three cases of real medical image data segmentation the PSAM algorithm performed well and, in fact, outperformed the existing ASM algorithm by a substantial margin. In particular, PSAM has a much larger degree of success compared to ASM on the most difficult cases (e.g., the mouse kidney images).

Finally, it was shown that the new local shape modeling formulation in this work is a positive addition to the PSAM boundary-based segmentation algorithm. When GSM and LSM are used simultaneously to optimize a boundary, the result is occasionally better than results using only the GSM for shape. It was also shown that in an application where global shape is highly variable, to the point where it maybe difficult to predict given the train-

ing samples, the use of LSMs in place of the GSM can improve the PSAM algorithm performance.

In summary, the strengths of new PSAM algorithm are the following:

- PSAM simultaneously optimizes with respect to boundary shape and gray-level characteristics using a priori information derived from a training set. In addition, PSAM simultaneously optimizes with respect to the boundary pose parameters (translation, scale, and rotation).

- PSAM provides control over the relative influence of each of the three model components (GSM, LSM, and GLM) via the adjustment of component weight parameters.

- The algorithm optimizes over pose parameters by modeling acceptable translation, scale, and rotation as measured from the training set.

- It incorporates local shape modeling for more controlled segmentation around critical LPs.

- It generates confidence metrics that can used to ascertain how well the final PSAM boundary shape and gray-level characteristics fit those measured in the training data; in some cases, the GLM confidence can be used to predict segmentation accuracy.

- PSAM performs well on a wide variety of semirigid object segmentation applications and significantly outperforms similar shape-based segmentation approaches.

This was demonstrated to be true especially for the more difficult segmentation problems.

- It provides a general theoretical framework for previously developed probabilistic boundary-finding techniques.

6.2 Future Research

There are many opportunities for future research using this PSAM algorithm as a starting point. Some involve improving the performance of the algorithm on 2D segmentation applications. Other opportunities involve extending this approach to 3D object segmentation.

Formulating a shape-modeling component that includes a theoretical dependence between the LSM and the GLM may improve the performance of PSAM when these two shape models are used simultaneously for boundary-based segmentation. The use of other image-derived features beyond gray-level profiles may have a positive impact on the performance of this approach; for example, using local texture measurements near the current boundary could provide additional useful information. Also, using LPs that are not necessarily located directly on the object boundary but placed on interesting features within the interior to the object of interest may improve the boundary-finding capability of the PSAM algorithm. Finally, alternative optimization approaches beyond gradient descent should be explored to determine if improved convergence is possible.

Another substantial opportunity for new research is in the extension of this approach to 3D object segmentation. There are two primary challenges in completing an extension

of the PSAM algorithm. The first is the theoretical development of the objective function in a 3D volumetric application, although the theoretical basis of PSAM supports direct 3D extension. The second challenge of extending the approach to 3D applications is providing the operator with a robust, easy-to-use training procedure to allow placement of LPs on the surface of an object embedded within a 3D volume. This is a challenging task and will involve both unique data visualization tools and some level of semiautomatic LP placement to ensure that corresponding LPs are similarly placed on the surface of each object in the training set.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Y. Amit, U. Grenander, and M. Piccioni, "Structural image restoration through deformable templates," *J. Amer. Stat. Assoc.*, vol. 86, no. 414, pp. 376-387, June 1991.

[2] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 12, no. 9, pp. 8555-8567, 1990.

[3] M. Atkins and B. Mackiewich, "Fully automatic segmentation of the brain in MRI," *IEEE Trans. on Med. Imaging*, vol. 17, no. 1, pp. 98-107, Feb. 1998.

[4] N. Ayache, I. Cohen, and I. Herlin, "Medical image tracking," A. Blake, A. Yuille (Eds.), *Active Vision*, chap. 17, pp. 285-301, 1992.

[5] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Patt. Rec.*, vol. 13, no. 2, pp. 111-122, 1981.

[6] C. Barillot, G. Le Goualher, P. Hellier, and B. Gibaud, "Statistical analysis of brain sulci based on active ribbon modeling," *Proc. SPIE Conf. on Med. Imaging*, pp. 312-321, 1998.

[7] B. Bascle and R. Deriche, "Region tracking through image sequences," *Proc. Int. Conf. Comp. Vis.*, pp. 302-307, June 1995.

[8] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *Int. J. Comp. Vis.*, vol. 11, no. 2, pp. 127-1458, 1993.

[9] C. Brechbühler, G. Gerig, and O. Kübler, "Parametrization of closed surfaces for 3-D shape description," *Comp. Vis. and Image Understanding*, vol. 61, no. 2, pp. 154-170, 1995.

[10] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematique 66*, pp. 7-37, 1993.

[11] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IEEE Proc. ICCV*, pp. 694-699, 1995.

[12] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int. J. of Comp. Vis.*, vol. 22, no. 1, pp. 61-79, 1997.

[13] A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding influenced by region homogeneity," *IEEE Proc. Conf. Comp. Vis. and Patt. Rec.*, pp. 624-627, 1994.

[14] A. Chakravart, M. Laha, and S. Roy, *Handbook of Methods of Applied Statistics,* Vol. I, John Wiley, pp. 392-394.

[15] S. Chandran and A. Potty, "Energy minimization of contours using boundary conditions," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, no. 5, pp. 546-549, May 1998.

[16] R. Chin and C. Dyer, "Model-based recognition in robot vision," *ACM Comp. Surveys*, vol. 18, pp. 67-108, 1986.

[17] T. Cho, "A knowledge-based machine vision system for automated industrial web inspection," *Ph.D. diss.*, Virginia Polytechnic Institute, Blacksburg, 1991.

[18] P. Ciarlet, *Introduction to Numerical Linear Algebra and Optimisation*, Cambridge University Press, Cambridge, 1989

[19] L. Cohen, "Note on active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211-218, Mar. 1991.

[20] I. Cohen, L. Cohen, and N. Ayache, "Using deformable surfaces to segment 3-D images and infer differential structures," *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 242-263, Sept. 1992.

[21] L. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1131-1147, 1993.

[22] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models — Their training and application," *Comp. Vis. and Image Understanding*, vol. 61, no. 1, pp. 38-59, Jan. 1995.

[23] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Proc. European Conference on Comp. Vis.*, vol. 2., pp. 484-498, 1998.

[24] T. Cootes and C. Taylor, "Active shape models," University of Manchester Technical Report, http://www.wiau.man.ac.uk, 1998.

[25] I. Cox, S. Rao, and Y. Zhong, "'Ratio regions: a technique for image segmentation," *Proc. Int. Conf. Patt. Rec.*, pp. 557-564, 1996.

[26] D. DeCarlo and D. Metaxas, "Shape evolution with structural and topological changes using blending," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, no. 11 pp. 1186-1205, Nov. 1998.

[27] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 19, no. 2, pp. 121-132, 1997.

[28] E. Di Mauro, T. Cootes, C. Taylor, and A. Lanitis, "Active shape model search using pairwise geometric histograms," *Proc. BMVC96*, 1996.

[29] M. Dubuisson and A. Jain, "Contour extraction of moving objects in complex outdoor scenes," *Int. J. Comp. Vis.*, vol. 14, pp. 83-105, 1995.

[30] M. Dubissson-Jolly, S. Lakshmanan, and A. Jain, "Vehicle segmentation using deformable templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 3, pp. 293-308, 1996.

[31] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, 1973.

[32] R. Duda and P. Hart, "Use of the Hough transforms to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11-15, 1972.

[33] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images using an improved knowledge based active shape model," *Proc. IPMI '97 Conf. Lecture Notes in Comp. Sci.*, pp. 375-380, 1997.

[34] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images: An improved active shape model," *IEEE Trans. on Med. Imaging*, vol. 17, no. 6, pp. 1049-1062, Dec. 1998.

[35] M. Figueiredo and J. Leitao, "Bayesian estimation of ventricular contours in angiographic images," *IEEE Trans. Med. Imaging*, vol. 11, no. 3, pp. 416-429, 1992.

[36] J. Fessler and A. Hero, "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms," *IEEE Trans. Im. Proc.*, vol. 4, no. 10, pp. 1417-29, Oct. 1995.

[37] M. Figuiredo, J. Leitao, and A. Jain, "Adaptive B-splines and boundary estimation," *Proc. IEEE Conf. Comp. Vis. Patt. Rec.*, pp. 724-730, 1997.

[38] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comp.*, vol. 22 no. 1, pp. 67-92, 1973.

[39] D. Fritsch, S. Pizer, L. Yu, V. Johnson, and E. Chaney, "Segmentation of medical objects using deformable shape loci," *IPMI '97, Lecture Notes in Comp. Sci.*, vol. 1230, pp127-140, 1997.

[40] P. Fua and C. Brechbuhler, "Imposing hard constraints on soft snakes," *Proc. Eur. Conf. on Comp. Vis.*, vol. 2, pp. 495-506, 1996.

[41] J. Gauch, H. Pien, and J. Shah, "Hybrid boundary-based and region-based deformable models for biomedical image segmentation," *SPIE Mathematical Methods in Med. Imaging III*, vol.2299, pp. 72-83, 1994.

[42] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 3, pp. 294-302, 1993.

[43] S. S. Gleason, H. Sari-Sarraf, M. J. Paulus, D. K. Johnson, and M. A. Abidi, "Deformable Model-based X-ray CT Image Segmentation for Automatic Phenotype Identification in Laboratory Mice," The Second Conference on Biomedical Engineering, Vanderbilt University, Nasheville, Tenn., April, 1999.

[44] S. Gleason, H. Sari-Sarraf, M. Paulus, D. Johnson, and M. Abidi, "Automatic screening of polycystic kidney disease in x-ray CT images of laboratory mice," *Proc. of the SPIE Med. Imaging Conf.*, February, 2000.

[45] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. of the Royal Statistical Society B*, 53(2):285-339, 1991.

[46] U. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford University Press, Oxford, 1993.

[47] U. Grenander and D. Keenan, "Towards automated image understanding," *Advances in Applied Statistics: Statistics and Images*, vol. 1, chap. 6, pp. 89-103, 1993.

[48] W. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA, 1990.

[49] C. Gritton and E. Parrish, Jr., "Boundary location from an initial plan: the bead chain algorithm," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. Pami-5, no. 1, pp. 8-13, Jan. 1983.

[50] R. Grzeazczuk and D. Levin, "'Brownian strings': segmenting images with stochastically deformable contours," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 19, no.10, pp. 1100-1114, Oct. 1997.

[51] A. Hill, A. Thornham, and C. Taylor, "Model-based interpretation of 3D medical images," *4th British Machine Vision Conference*, pp. 339-348, Sept. 1993.

[52] A. Hill, T. Cootes, C. Taylor, and K. Lindley, "Medical image interpretation: A generic approach using deformable templates," *J. of Med. Informatics*, vol. 19, no. 1 pp. 47-59, 1994.

[53] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *Proc. Eur. Conf. on Comp. Vis.*, vol. 1, pp. 343-356, 1996.

[54] A. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 3, pp. 267-278, 1996.

[55] A. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1386-1390, 1997.

[56] A. Jain, Y. Zhong, and M. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, vol. 71, no. 2, pp, 109-129, Dec. 1998.

[57] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. of Comp. Vis.*, vol. 1, pp. 321-331, 1988.

[58] A. Kelemen, G. Székely, and G. Gerig, "Three-dimensional model-based segmentation," *SPIE Proc. Workshop on Biomedical Image Anal.*, pp. 4-13, 1998.

[59] C. Kervrann and F. Heitz, "Robust tracking of stochastic deformable models in long image sequences," *Proc. Int. Conf. on Image Processing*, vol. 3, pp. 88-92, 1994.

[60] C. Kervrann and F. Heitz, "A hierarchical statistical framework for the segmentation of deformable objects in image sequences," *Proc. IEEE Conf. Comp. Vis. Patt. Rec.*, pp. 724-728, 1994.

[61] C. Kervrann and F. Heitz, "A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes," *Graphical Models and Image Processing*, vol. 60, no. 3, pp. 173-195, May 1998.

[62] K. F. Lai, "Deformable contours: Modeling, extraction, detection and classification," Ph.D. diss., Univ. of Wisconsin-Madison, 1994.

[63] A. Lanitis, C. Taylor, and T. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 19, no. 7, July 1997.

[64] F. Lefebvre, G. Berger, and P. Laugier, "Automatic detection of the boundary of the calcaneus from ultrasound parametric images using an active contour model; clinical assessment," *IEEE Trans. on Med. Imaging*, vol. 17, no. 1, pp. 45-52, Feb. 1998.

[65] F.Leymarie and M. Levine, "Tracking deformable objects in the plane using as active contour model," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 6, pp. 617-634, 1993.

[66] S. Lobregt and M. Viergever, "A discrete dynamic contour model," *IEEE Trans. on Med. Imaging*, vol. 14, no. 1, pp. 12-24, Mar 1995.

[67] L. Lorigo, O. Faugeras, W.E.L. Grimson, R. Keriven, and R. Kikinis, "Segmentation of bone in clinical knee MRI using texture-based geodesic active contours," *Proc. 1st Int. Conf. on Med. Image Computing and Comp.-Assisted Intervention (MICCAI)*, pp. 1195-1204,1998.

[68] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(2):158-175, Feb. 1995.

[69] K. Mardia, W. Qian, D. Shah, and K. de Souza, "Deformable template recognition of multiple occluded objects," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 19, no. 9, Sept. 1997.

[70] J. Marques and A. Abrantes, "A class of probabilistic shape models," *Proc. IEEE Conf. Comp. Vis. Patt. Rec.*, pp. 1054-1059, 1997.

[71] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91-108, 1996.

[72] T. McInerney and D. Terzopoulos, "Medical image segmentation using topologically adaptable surfaces," *Proc. CVRMed '97*, pp. 1-10, Mar 1997.

[73] A. Mirhosseini, H. Yan, and K. Lam, "Adaptive deformable model for mouth boundary detection," *Society of Photo-Optical Instrumentation Engineers*, vol. 37, no. 3, pp. 869-875, Mar 1998.

[74] M. Moshfeghi, S.ranganath, and K. Nawyn, "Three-dimensional elastic matching of volumes," *IEEE Trans. Image Proc.*, vol. 3, no. 2, pp. 128-138, 1994.

[75] W. Niessen, B. ter Haar Romeny, and M. Viergever, "Geodesic deformable models for medical image analysis," *IEEE Trans. on Med. Imaging*, vol. 17, no. 4, Aug. 1998.

[76] W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler, "Making snakes converge from minimal initialization," *Proc. Int. Conf. Patt. Rec.*, vol. A, pp. 613-615, 1994.

[77] T. O'Donnell, M. Dubuisson-Jolly, and A. Gupta, "Cooperative framework for segmentation using 2D active contours and 3D hybrid models as applied to branching cylindrical structures," *Proc. Int. Conf. Comp. Vis.*, pp. 454-459, 1998.

[78] M. Paulus, H. Sari-Sarraf, S. Gleason, M. Bobrek, J. Hicks, D. Johnson, J. Behel, and L. Thompson, "A new x-ray computed tomography system for laboratory mouse imaging," *IEEE Trans. on Nucl. Sci.*, July, 1999.

[79] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, no. 7, pp. 715-729, July 1991.

[80] C. Poon, M. Braun, R. Fahrig, A. Ginige, and A. Dorrell, "Segmentation of medical images using an active contour model incorporating region-based image features," *SPIE Proc. Third Conf. On Visualization in Biomedical Eng.*, vol. 2359, pp. 90-97, 1994.

[81] M. Revow, C. Williams, and G. Hinton, "Using generative models for handwritten digit recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 6, pp. 592-606, 1996.

[82] R. Ronfard, "Region-based strategies for active contour models," *Int. J. of Comp. Vis.*, vol. 13, no. 2, pp. 229-251, 1994.

[83] A. Rosenfeld and A. Kak, *Digital Picture Processing*, 2nd Ed., vol. 2, Academic Press, Orlando, 1982.

[84] G. Sapiro, "Vector-valued active contours," *Proc. IEEE Comp. Vis. Patt. Rec.*, pp. 680-685, 1996.

[85] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 6, pp. 545-561, 1995.

[86] P. Smyth, C. Taylor, and J. Adams, "Automatic measurement of vertebral shape using active shape models," *Proc. BMVC96*, pp. 176-180, 1996.

[87] L. Staib and J. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, no. 11, Nov. 1992.

[88] M. Stephens, "EDF Statistics for Goodness of Fit and Some Comparisons," *J. Am. Stat. Assoc.*, vol. 69, pp. 730-737.

[89] G. Storvik, "A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 10, pp. 976-986, 1994.

[90] G. Szekely, A. Keleman, C. Brechbuhler, and G. Gherig, "Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models," *Med. Image Anal.*, vol. 1, no. 1, pp. 19-34, 1996.

[91] H. Tagare, "Deformable 2-D template matching using orthogonal curves," *IEEE Trans. on Med. Imaging*, vol. 16, no. 1, pp. 108-117, Feb. 1997.

[92] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: deformable superquadrics," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, no. 7 pp. 703-714, July 1991.

[93] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 6, pp. 569-579, 1993.

[94] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable modes: Recovering 3D shape and nonrigid motion," *Artificial Intell.*, vol. 36, pp. 91-123, 1998.

[95] A. Vailaya, Y. Zhong, and A. Jain, "A hierarchical system for efficient image retrieval," *Proc. Int. Conf. Patt. Rec.*, Vienna, Austria, pp. 356-360, 1996.

[96] Y. Wang and L. Staib, "Boundary finding with correspondence using statistical shape models," *IEEE Proc. Comp. Vis. and Patt. Rec.*, pp. 338-345, 1998.

[97] Y. Wang and L. Staib, "Boundary finding with prior shape and smoothness models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 7, pp. 738-743, July 2000.

[98] R. Whitaker, "A level-set approach to 3D reconstruction from range data," *Int. J. of Comp. Vis.*, vol. 29, no. 3, pp. 203-231, 1998.

[99] B. Widrow, "The 'rubber-mask' technique — I. Pattern measurement and analysis," *Patt. Rec.*, vol. 5, pp. 175-197, 1973.

[100] B. Widrow, "The 'rubber-mask' technique — II. Pattern storage and recognition," *Patt. Rec.*, vol. 5, pp. 199-211, 1973.

[101] C. Xu and J. Prince, "Gradient vector flow: A new external force for snakes," *IEEE Proc. Conf. Comp. Vis. Patt. Recog.*, pp. 66-71, 1997.

[102] A. Yuille, P. Hallinan, and D. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comp. Vis.*, vol. 8, no. 2, pp. 133-144, 1992.

[103] Y. Zhong and A. Jain, "Object localization using color, texture and shape," *Proc. Workshop on Energy Minimization Methods in Comp. Vis. and Patt. Rec., Venice,* Springer, Berlin, pp. 279-294, 1997.

[104] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884-899, Sept. 1996.

VITA

Shaun Scott Gleason was born in Portland, Maine on May 2, 1967. He moved to Knoxville, Tennessee in 1975, and he has resided there ever since. He graduated from Farragut High School in 1985, and went on to receive a B.S. in electrical engineering from the University of Tennessee, Knoxville, in 1989. While attending the University of Tennessee, he was a member of Tau Beta Pi and Eta Kappa Nu honor societies.

In June of 1989, he began his career at Oak Ridge National Laboratory (ORNL) in the Image Science and Machine Vision group within the Instrumentation and Controls Division. While working at ORNL he earned an M.S. in electrical engineering in 1992. He currently maintains a part-time position at ORNL, where his research interests are computer-aided diagnosis for medical images, tomographic image reconstruction, and vision systems for monitoring manufacturing processes.

Since April of 1998, he has also been vice president of ImTek, Inc., a manufacturer of high-resolution small-animal imaging systems located in Knoxville, Tennessee. His research efforts there include X-ray computed tomography system development and volumetric data reconstruction/analysis.

He is married to the former Caroline Johnson of Knoxville, Tennessee. They have two daughters, Palmer, born in 1994, and Paige, born in 2000.