

## ABSTRACT

Parametrically Deformable Contour Models for Image Analysis

Lawrence Hamilton Staib

Yale University

1990

A practical system for boundary finding of natural objects in images has been developed. It is based on a new general probabilistic method of boundary finding that allows the incorporation of prior information about the global shape of the target object. Determining the boundaries of objects, and thereby their shape and location, is an important task in computer vision. Segmentation using boundary finding is enhanced both by considering the boundary as a whole and by using model-based global shape information. Previous boundary finding methods have either not used global shape or have designed individual shape models specific to particular shapes. Imperfect image data can be augmented by exploiting the extrinsic information that a model provides. Flexible constraints in the form of a probabilistic deformable model are applied to the problem of segmenting natural objects whose diversity and irregularity of shape makes them poorly represented in terms of fixed features or form. The objects being considered are expected, however, to have a tendency toward some average shape. The parametric model is based on the elliptic Fourier decomposition of the boundary. This is augmented with probability distributions defined on the parameters, which bias the model to a particular overall shape while allowing for deformations. Boundary finding is formulated as a optimization problem using a maximum a posteriori objective function. The best match is found between the boundary, as defined by the parameter vector, and a measure

of image boundary strength derived from the image, as biased by the shape prior probability. A computer implementation was constructed and applied to object delineation problems from a variety of two-dimensional images. Results of the method applied to real and synthetic images are presented. Extensions of this method to three dimensions and temporal sequences are outlined.

Parametrically Deformable Contour Models for Image Analysis  
Yale University  
1990  
Doctoral Dissertation  
Lawrence H. Staib  
Errata

Page 20. In Equation 3.3, the numerators should be  $|x - x_0|^{\frac{2}{q}}$  and  $|y - y_0|^{\frac{2}{q}}$

Page 23. In Equation 3.7 and on Page 25, Equations 3.9 and 3.10,  $v(t)$  should be  $\mathbf{v}(t)$ .

Page 73. On the third line after Equation 5.18, “This method of adjusting” should read “The method of adjusting.”

Page 80. Citation [92] on line 5 should be to: A. P. Witkin, “Scale space filtering”, IJCAI, 1019-1022, 1983.

Page 110. The second summation in Equation 7.5 should be over  $l$ , not  $m$ .

Page 114. The second line should read “example closed surface” not “example tube surface.”

Parametrically Deformable  
Contour Models  
for Image Analysis

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Lawrence Hamilton Staib

Dissertation Director: James Scott Duncan

December 1990

© Copyright by Lawrence Hamilton Staib 1990

ALL RIGHTS RESERVED

## Acknowledgements

This research was fostered and made enjoyable by the congenial, open and egalitarian atmosphere of the Division of Imaging Science. I thank my advisor, James Duncan, for introducing me to the field and providing encouragement, direction, ideas, instruction and support. Gene Gindi asked tough questions and gave many valuable suggestions. I also thank Stelios Orphanoudakis and Vladimir Lumelsky for serving on my committee and providing many helpful comments. I thank P. Anandan, Arnold Smeulders, Carl Jaffe and Perry Miller for their interest and help. Kathy Schlitter provided administrative assistance and encouragement. Financial support for this work was provided, in part, by grant T15LM07056 from the National Library of Medicine.

I thank the following current and erstwhile denizens of the Division of Imaging Science and fellow travelers who helped to make life here enjoyable: Kathy Andriole, Thomas Birkhölzer, Işıl Bozma, D. J. Delorie, Dimitris Gerogiannis, Arthur Gmitro, Robin Greene, Chuck Harrell, Mindy Lee, Sharmila Majumdar, Randy Owen, Tsutomu Shibata, Lisa Simmons, Volker Tresp, Joachim Utans, Elliot Zaret, and George Zubal.

I thank my family for their support, encouragement and for the many good times we have spent together. I will always be indebted to my father for the intellectual influence he had on me. I also thank Eva Kaufman. Without her, I might have been done sooner, but it wouldn't have been as much fun. She is a tremendous source of support and encouragement. Finally, it can be toasted.

# List of Figures

1.1	System overview. . . . .	5
3.1	Contour constructed from ellipses with raw parameters. . . . .	26
3.2	Contour constructed from ellipses with relative parameters. . . . .	33
3.3	Curve for reconstruction error example. . . . .	38
3.4	Reconstruction error example. . . . .	39
3.5	Example mean curve plus and minus one standard deviation. . . . .	42
3.6	Example parameter standard deviations. . . . .	43
5.1	Length of curve in each pixel. . . . .	63
5.2	Example template. . . . .	64
5.3	Powell's method convergence example. . . . .	71
5.4	Continuous gradient ascent convergence example. . . . .	77
5.5	Multiresolution example. . . . .	83
6.1	Synthetic image example. . . . .	88
6.2	Synthetic image open curve example. . . . .	90
6.3	Noise experiment images. . . . .	91
6.4	Sensitivity to noise experiment. . . . .	92

6.5	Parameter sensitivity experiment parameters. . . . .	94
6.6	Sensitivity to initial parameters experiment. . . . .	95
6.7	Bias experiment. . . . .	97
6.8	Left lateral radionuclide angiogram example. . . . .	99
6.9	Long-axis echocardiogram example. . . . .	101
6.10	Magnetic resonance mid-brain sagittal image example. . . . .	102
6.11	Magnetic resonance oblique sagittal cardiac image example. . . . .	103
6.12	Magnetic resonance transaxial cardiac image example. . . . .	105
6.13	Magnetic resonance oblique sagittal cardiac image example. . . . .	106
7.1	Boundary constraint example. . . . .	109
7.2	Torus surface example. . . . .	112
7.3	Open surface example. . . . .	114
7.4	Tube surface example. . . . .	116
7.5	Closed surface example. . . . .	117
7.6	Magnetic resonance oblique sagittal cardiac image motion example. . . . .	119



# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction to the Problem . . . . .	3
1.2 Overview . . . . .	4
1.3 Main Contributions . . . . .	4
<b>2 Related Work in Boundary Finding</b>	<b>7</b>
2.1 Region Methods . . . . .	7
2.2 Boundary Methods . . . . .	8
2.2.1 Edge Detectors . . . . .	8
2.2.2 Grouping . . . . .	9
2.2.3 Pixel Search . . . . .	10
2.2.4 Hough Methods . . . . .	10
2.2.5 Whole-Boundary Methods . . . . .	11
<b>3 Model</b>	<b>14</b>

3.1	Parametrization Design Considerations . . . . .	15
3.2	Alternative Parametrizations . . . . .	16
3.2.1	Direct and Differential . . . . .	16
3.2.2	Extended Gaussian Image . . . . .	18
3.2.3	Splines . . . . .	18
3.2.4	Sweeps . . . . .	19
3.2.5	Superquadrics . . . . .	19
3.2.6	Fourier . . . . .	20
3.3	The Deformable Contour Parametrization . . . . .	22
3.3.1	Closed Curves . . . . .	22
3.3.2	Open Curves . . . . .	32
3.3.3	Number of Harmonics . . . . .	35
3.4	The Parameter Probability Distributions . . . . .	38
<b>4</b>	<b>Matching</b>	<b>44</b>
4.1	Match Measures . . . . .	45
4.2	Boundary Justification . . . . .	46
4.3	Maximum <i>a Posteriori</i> Match . . . . .	47
4.3.1	Bayes Rule . . . . .	47
4.3.2	Likelihood Derivation . . . . .	49
4.3.3	Boundary Formulation . . . . .	50
4.3.4	Independence Assumption . . . . .	53
4.3.5	Noise . . . . .	54
4.4	Boundary Measures . . . . .	55
4.5	Boundary Finding Objective Function . . . . .	56

<b>5</b>	<b>Implementation</b>	<b>58</b>
5.1	Objective Function Calculation . . . . .	58
5.1.1	Boundary Measure Implementations . . . . .	59
5.1.2	Template Generation . . . . .	60
5.1.2.1	Algorithm . . . . .	62
5.2	Optimization Techniques . . . . .	65
5.2.1	Powell’s Direction Set Method . . . . .	68
5.2.2	Continuous Gradient Ascent . . . . .	70
5.2.2.1	Ascent Method . . . . .	72
5.2.2.2	Gradient Formulation . . . . .	76
5.3	Smoothing . . . . .	82
<b>6</b>	<b>Experiments</b>	<b>85</b>
6.1	Evaluation . . . . .	85
6.2	Synthetic . . . . .	87
6.2.1	Varying Noise . . . . .	89
6.2.2	Varying Initial Parameters . . . . .	93
6.2.3	Varying Bias . . . . .	93
6.3	Real Images . . . . .	96
6.3.1	Nuclear Medicine . . . . .	96
6.3.2	Echocardiography . . . . .	98
6.3.3	Magnetic Resonance Imaging . . . . .	100
6.3.4	Summary . . . . .	104
<b>7</b>	<b>Extensions</b>	<b>107</b>
7.1	Boundary Constraints . . . . .	107

7.2	Three dimensions . . . . .	108
7.2.1	Curves in three dimensions . . . . .	110
7.2.2	Surfaces in three dimensions . . . . .	110
7.3	Temporal sequences . . . . .	116
7.4	Multimodality . . . . .	118
<b>8</b>	<b>Summary</b>	<b>120</b>
<b>A</b>	<b>The Gaussian</b>	<b>122</b>
	<b>Bibliography</b>	<b>123</b>

# Chapter 1

## Introduction

The computational formulation of visual tasks, computer vision, is applicable to all fields where the relevant data are expressed pictorially. An important part of computer vision is the analysis of images, the goal of which is the quantitative extraction of information concerning the objects depicted in the image. The objects in the image must be identified or segmented in order to further analyze them. This work is concerned with the segmentation of individual objects from images.

Distinguishing an object from its surroundings in an image requires both the image data and additional information as to how the object distinguishes itself. This additional information is based on a model of the object. Models can range from simple feature criteria (e.g., a light object on a dark background) to more precise specifications (e.g., an exact template).

Marr's view of visual information processing is that top-down, or model-based, information is only of secondary importance, based on psychophysical evidence, especially from stereo and motion [58]. For the purposes of boundary determination, this requires autonomous low-level feature detection followed by organizing or grouping processes.

Unfortunately, grouping processes have been successful, in general, only in conjunction with model-based information.

There are many ways in which objects can be discerned in images. Local properties, such as gray level, texture, or color, act as visual cues. Objects may satisfy some homogeneity criterion in their interior with respect to a local property. Other objects may be best discerned at their boundary by the change of a local property at the object's boundary, or some other local measure that distinguishes the boundary. These properties, however, are often inconsistent and incomplete due to inexact criteria or imperfect viewing conditions. A putative homogeneity may not be satisfied at all points in the object, or the property change may not be discernable at all portions of the boundary. For this reason, these local properties need to be augmented with more global properties.

A powerful distinguishing property is overall shape. Shape can be used to complete the information provided by local properties. A preconception of shape can help resolve ambiguous information provided by local properties. Local shape features, such as curvature, can be useful but they are not as expressive as global shape and, like other local features, are more sensitive to poor viewing conditions. Global shape is too varied to be adequately described by a single shape attribute such as average bending energy or compactness. Attneave's view of the importance of parsimony in the description of visual phenomena, based on ideas from information theory [5], is of general significance for the purposes of formulating a model of shape. A concise representation that removes redundancy allows for computational efficiency.

In order to take full advantage of shape, the problem of object identification will be approached as a process of boundary finding or delineation using a boundary measure and incorporating global shape information. The analysis of shape has been largely limited

to processing after delineation for classification or other purposes. This work deals with the mathematical modeling of shape for use in model-based boundary finding.

## 1.1 Introduction to the Problem

Segmentation by boundary finding using only local information has often been frustrated by poor-contrast boundary regions due to occluding and occluded objects, adverse viewing conditions, and noise. A model-free interpretation is doomed by the underconstrained nature of the problem. Imperfect image data can be augmented with the extrinsic information that a geometric shape model provides. In order to exploit model-based information to the fullest extent, it should be incorporated explicitly, specifically, and early in the analysis. When the model is hidden in algorithm heuristics, it may perform well, but the generality is lost. Some models incorporate generic information such as smoothness or low overall curvature. While this is completely appropriate when no better information is available, the more specific the shape information used, the better. Applying the model too late allows inconsistencies to be created by the low-level processing. In addition, the boundary can be profitably considered as a whole because it tends to result in a more overall consistent solution.

This work is aimed at segmenting natural objects, especially focused on those found in biomedical images, whose diversity and irregularity of shape makes them poorly represented in terms of fixed features or form. The objects, such as organs, cells and other biological structures, are expected to have a tendency toward some average shape. There will be, however, a continuum of possibilities near that average shape. Biomedical images are an important application because automatic quantitative analysis of objects present in these images is needed for both research and clinical applications. In addition, it

provides a rich domain for the study of shape. This model domain is somewhere between the extremes of a completely arbitrary object and a fixed template. This tendency can be taken advantage of by its expression in an appropriately designed shape model.

## 1.2 Overview

A complete system using model-based optimization for the determination of object boundaries has been developed [83, 84]. This system is composed of two main elements: the shape model and the match measure. The shape model, described in Chapter 3, is a parametrization, represented by a parameter vector  $\mathbf{p}$  with associated prior probability distributions,  $P(\mathbf{p})$ . The match measure is developed in Chapter 4 as a way of combining the probabilistic information that the shape model provides with boundary information that the image provides into a single objective function,  $M(\mathbf{p})$ . The information derived from the image,  $i$ , is a boundary measure,  $b(i(x, y))$ . The objective function is based on a maximum *a posteriori* (MAP) formulation that involves the prior and a correlation,  $C$ , between the boundary measure and a template,  $t(\mathbf{p})$ . The template is generated from the current value of the parameter vector. The optimization process starts at an initial point  $\mathbf{p}_0$ , determined by the parameter's prior probability distribution. At each iteration, the parameter vector is updated by the amount  $\Delta\mathbf{p}$ , determined by the optimization method. The template formation and the methods of optimization used are described in Chapter 5. A diagram outlining the approach is shown in Figure 1.1.

## 1.3 Main Contributions

A practical system for boundary finding of natural objects has been designed. A computer implementation has been developed and tested on real and synthetic images. It is based



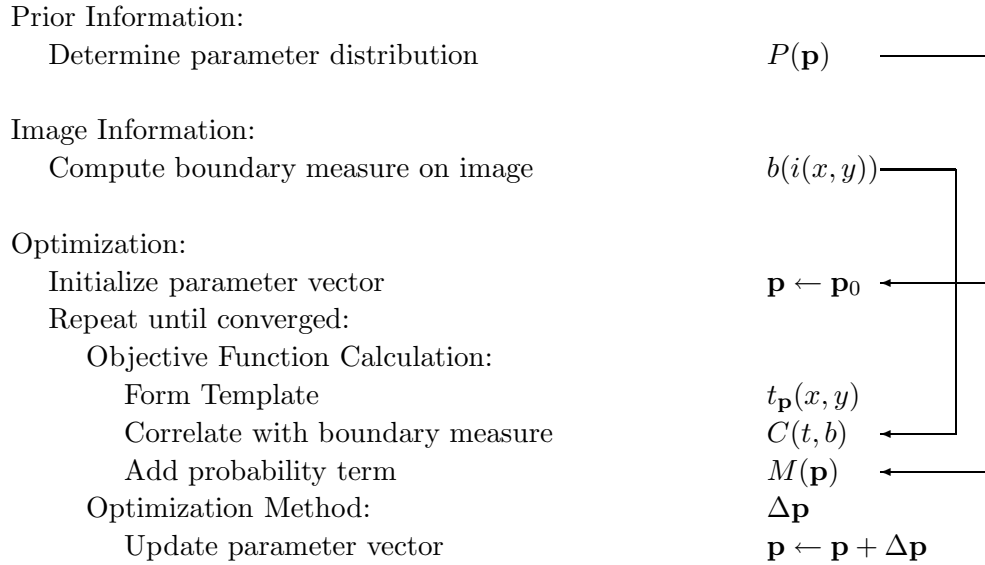


Figure 1.1: System overview.

on a general probabilistic method of boundary finding that allows the incorporation of prior information about the global shape of the target object. Previous boundary finding methods have either not used global shape or have designed individual shape models specific to particular shapes. A parametrization for two-dimensional curves has been developed into a practical shape representation for boundary finding. Probability distributions on the parameters of the representation bias the model to a particular overall shape while allowing for deformations. Boundary finding is formulated as an optimization problem using a maximum *a posteriori* objective function. Extensions of this method to

three dimensions and temporal sequences are outlined.

## Chapter 2

# Related Work in Boundary

# Finding

### 2.1 Region Methods

Considerable work has focused on determining objects in images using region analysis. Typically this involves defining a homogeneity criterion, and perhaps other criteria such as semantic consistency. The segmentation is then determined by region growing methods such as splitting [67], merging [17] or both [44]. These approaches depend on the uniformity of regions in order to determine the salient regions in an image. These methods can be viewed as complementary to boundary methods in that they profit from the spatial coherence in objects. They work reliably in the interiors of regions, but the difficulty in segmentation still lies in deciding where the region ends, that is, its boundary. Region methods can be susceptible to errors at the boundary and often use further heuristics or smoothing to improve the boundaries that result. Region analysis could perhaps benefit from the incorporation of boundary methods. Region analysis is an attempt to compen-

sate for the problems of boundary methods used alone, which are sometimes susceptible to large errors due to spurious or missing edges. The current work is robust to this sort of problem through the use of a prior model and thus can focus on the boundary. Further justification of the boundary approach is presented in Section 4.2. In general, the lack of focus on the boundary in region analysis and the restriction to objects with a well-defined uniformity (often not present), seem to be unwarranted impairments.

## 2.2 Boundary Methods

The major alternative to region analysis is boundary analysis. This approach concentrates on the boundaries of objects as the key distinguishing feature. Given the limitations of region methods, boundary methods are attractive. A number of approaches using boundary information are discussed below. It is not meant to be a comprehensive list, but should give an idea of the range of approaches, while emphasizing those with similarities to the current approach.

### 2.2.1 Edge Detectors

To some, using boundary methods means doing edge detection, that is, calculating a binary edge image. Edge detectors have been used since the earliest work in computer vision and thus many methods have been proposed. One influential method of edge detection based on zero-crossings of the Laplacian of the Gaussian of the image was proposed based on a biological motivation [59]. Canny developed a widely used edge detector based on optimizing certain detection and localization criteria [19].

The problem with edge detectors for boundary finding is that the edges found do not necessarily correspond to boundaries of objects. With the exception of high quality

images from controlled environments, edge detectors produce spurious edges and gaps. For some applications however, such as stereo matching, this does not preclude their use. While there are ways of cleaning up edge images [77], there inevitably will be important information lost. Thus, while boundary information is undoubtedly useful, edge detectors *per se* are of limited use in general and of no use in poor quality images.

The limitations of edge detectors are due in part to their complete reliance on computations made directly on a local neighborhood of pixels in the raw image. By using only image information, any model-based information that may be available is ignored. By computing locally and directly, any higher order organization to the image is also ignored. The decision of edge versus no-edge is made prematurely if this information is not applied.

### 2.2.2 Grouping

Grouping for boundary finding is a way of associating edge elements for the purpose of determining boundaries. The association is done using similarity relations, perhaps in conjunction with model information. A simplistic form of grouping, or linking, is used by Nevatia and Babu [64] for binary edge images where neighboring edge points are grouped if they point in similar directions. Forming a complete boundary is usually accomplished by first associating individual edge elements into edge segments, then associating the segments into a boundary. Grouping has been used for noisy images by grouping edge points using a knowledge-based approach [85]. A probabilistic approach to grouping edge segments based on maximum entropy has also been described [26]. A grouping method for recognition of fixed polyhedral objects was developed by Lowe [55]. Some of this work derives its motivation from the ideas of perceptual organization [91]. Grouping methods, while robust to areas of weak boundary definition, often resort to arbitrary interpolation

in order to form a complete boundary. In addition, it is often difficult to identify and discount spurious edge segments.

### 2.2.3 Pixel Search

Pixel search methods attempt to find an optimal path through an image, based on criteria designed to find boundaries. Most methods try to find the optimum of a function that is a combination of boundary strength and low overall curvature. Montanari [63] and others [8, 34] have used dynamic programming to solve this optimization problem because the function only involves local computations. Dynamic programming is discussed further in Section 5.2.

Martelli [60] formulated the problem in terms of graph searching and used the heuristic search algorithm, A\*, to find the optimal path. This method, which uses heuristic information to speed up the search, reduces the amount of computation needed.

While pixel search methods are powerful, especially when no specific shape information is known, they are limited by their restriction to local shape properties.

### 2.2.4 Hough Methods

An alternative method for boundary analysis is the Hough transform. It was originally formulated for straight lines and simple parametrizable curves such as circles [25]. The technique was extended to general shapes by Ballard [9]. The Hough transform is a mapping from image space to a parameter space. Likely boundary points are first identified in the image. Each of these points has a set of points in the parameter space corresponding to the values of the parameters that could possibly include this boundary point. The accumulation of these correspondences is the Hough transform. Maxima in the Hough space correspond to possible instances of the shape, defined by the associated parameters.

The maxima are found by examining the Hough space. Thus, storage and computational complexity increase exponentially with the number of parameters. Typical parameters are translation, rotation and scale. The advantage of this technique is that it is relatively insensitive to gaps and noise. In fact, under certain conditions, the Hough method is simply a means of implementing matched filtering [80]. The Hough approach is therefore very similar to the current method in that it finds shapes using a parameter space and it is based on template matching. However, the storage and computational complexity of the Hough method are a great disadvantage especially if deformations are envisaged, although alternate formulations have been explored [47]. The advantage of the current technique, as will be seen, is that the entire parameter space does not have to be constructed due to the use of local search in finding the optimal point.

### 2.2.5 Whole-Boundary Methods

Other investigators have considered whole-boundary methods that adjust a tentative boundary in order to match to the image. By considering the boundary as a whole, a structure is imposed on the problem that simplifies the task. Gaps are easily bridged and overall consistency is more likely to result. The method described in this work is a whole-boundary method. One of the first instances of this type of approach is that of Widrow [88], who used parametrized templates called rubber masks to model objects. The parameters are sizes and relationships between subparts. He used this approach to identify chromosomes and the model is specific to this task. Yuille *et al.* [92] used a parametrized template for an eye consisting of a circle bounded by two parabolas. The template was matched to the image by optimizing a match based on morphological features. They developed a similar template for the mouth. Both of these methods have the advantage of describing the overall shape of the structure using very few parameters.

However, the object must have sufficient structure to be represented in terms of parts and a new model must be developed for each new object. These part models are related to the representation used by Fischler and Elschlager [29], where components of an object are held together by spring forces. The current method does not rely on a part description and is not specific to a particular object but supplies specific global shape information. Schudy [78] used a spherical harmonic parametrization for boundary finding (in this case, to moving, three-dimensional data representing the heart). This model is fairly general but provides no explicit shape information other than smoothness. All three of these methods match the model to the image data by searching the parameter space for the best fit.

While the above whole-boundary methods optimized in parameter spaces, the following methods optimize in the image space. The boundary finding method of Gritton and Parrish [39] used a flexible bead chain, where the beads are putative boundary points. The beads are adjusted to neighboring locations to improve the match to the image, in this case, image gradient strength. Prior information is incorporated by the use of an initial boundary and by adjusting algorithmic parameters dictating the influence of neighboring beads. Cooper [23, 22] used statistical likelihood as a match metric resulting in a correlation measure. A boundary adjustment scheme similar to the bead chain algorithm [39] is presented to perform the optimization. Kass *et al.* [49] used energy-minimizing snakes that are attracted to image features such as lines and edges, while internal spline forces impose a smoothness constraint. The weights of the smoothness and image force terms in the energy functional can be adjusted for different behavior. In addition, they provide a more general and unified treatment of the optimization process by taking advantage of standard numerical techniques for partial differential equations.



The use of image space representations for the boundary (as opposed to parameter space) makes it difficult to incorporate global shape information and none of these methods do so. The work of Kass *et al.* has been extended to include hard constraints, such as holding the boundary to binary edge data, by using dynamic programming [3]. The idea of energy minimizing models has also been applied to three-dimensional surfaces imposing an additional axial symmetry force [86]. These boundary finding methods are related to elastic matching methods used for the similar problem of registration between images [6, 18]. One image is modeled as an elastic material. Attractive forces deform the elastic image toward similar features in a goal image until they are balanced by the forces due to the stiffness of the material.

All of these approaches take an initial estimate of the contour and adjust it to optimize some measure of fit. Widrow and Yuille *et al.* use explicit global shape information. Their models are best designed for structures with well-defined parts. The other methods described limit their use of shape information to overall smoothness properties and the implicit shape information provided by the initial placement of the contour. The use of prior information ranges from very general (e.g., smoothness constraints) to very specific (e.g., exact templates). The amount of prior information available in a particular domain determines how much can be used. The boundary finding method described in the following chapters is aimed at the situation between the extremes, where there is some prior information about the global shape of the object, but it is not exact.

## Chapter 3

# Model

A model is needed that allows the application of prior information about shape to the problem of boundary determination. The objects being modeled have smooth boundaries that are continuously deformable. Thus, the determination of shape must be achieved through continuous decisions rather than discrete ones. These objects do not necessarily have an obvious decomposition that could be exploited, like a pair of scissors [38] or even an eye [92]. Because overall shape is the only reliable salient feature, a uniform representation that describes the entire shape is needed. The prior information is not a set of hard constraints but a flexible bias towards more likely shapes. This sort of model can be achieved by using a generic parametrization with probability distributions defined on the parameters. That is, the parametrization itself will be expressive enough to represent any potential shape of a given geometric type (for example, closed curves), but the associated probability distributions will introduce a bias towards an expected range of shapes. The spread in the distributions is due to variability among instances of the object. This kind of parametrization represents a stronger use of prior information than methods that use only simple shape characteristics.

### 3.1 Parametrization Design Considerations

Each parametrization has particular properties that suit it for different purposes. The class of shapes that a parametrization can express is important because it represents a limitation of domain. Some restrictions, such as smoothness, can be convenient because they build a necessary constraint directly into the representation. Other restrictions, such as convexity, could represent a design compromise in that the class of problems addressed is limited by the representation. For matching purposes, it is important that there is a one-to-one relationship, or mapping, between the shape and its parametrization. This mapping should also be continuous so that small deformations of the shape always result in small changes in the parametrization. This allows matching in the parameter space. In addition, it is desirable for the parametrization to be concise because that determines the complexity of the matching process. The parametrization should represent the shape economically by removing redundancy.

Certain operations are more convenient with some representations than with others. Some geometric properties such as area or moments are directly available from some representations. The conversion between the parametrization and the shape is usually very important. The conversion can be represented by two functions:  $\mathbf{P}(\mathbf{x})$  and the inverse  $\mathbf{X}(\mathbf{p})$  where  $\mathbf{x}$  is the vector of discrete points on the object in the image space and  $\mathbf{p}$  is the parameter vector. These functions are not always directly computable. For example,  $\mathbf{X}(\mathbf{p})$  is not directly available in implicit representations or region representations.  $\mathbf{P}(\mathbf{x})$  is often only computable through an optimization process as is the case with superquadrics [7].

Some representations, usually implicit, allow the use of an inside-outside function. An inside-outside function is a function of position in the object space that indicates

whether the point is inside or outside of the object and by how much. This can then be used as the basis for an objective function to be optimized [7, 78].

The proper handling of the degrees of freedom of an object due to varying the view angle and position is desirable because often they cannot be controlled. Changing the view results in the corresponding geometric transformations of translation, rotation and scaling. Some representations allow a simple application of the transformation directly to the representation. Some parametric representations can express the view by the individual parameters of the transformations.

## 3.2 Alternative Parametrizations

Boundary representations are the only representations that are relevant to this work because of the focus on shape and need for the boundary to be directly obtainable. Region representations such as quad trees or axis representations such as medial axis transform are impractical because the boundary is not easily accessible. Besl [14] presents a good overview of representations for computer vision. The main alternative parametric boundary representations relevant to this work are presented below.

### 3.2.1 Direct and Differential

Direct representations are those that stay close to the actual coordinates of the object in the object's space. The most general version is just an explicit list of the coordinates representing the object in an arbitrary order. If the object points can be spatially ordered, or indexed, by a spatial parameter, this ordering should be used. Any curve, for example, can be ordered by arclength  $s$ , and represented as  $x(s)$  and  $y(s)$ . This is also sometimes written as a single complex function,  $x(s) + iy(s)$ .

A direct representation can be derived from points of curvature maximum based on Attneave's demonstration of their perceptual importance for curve segmentation [5]. These maxima points are not, however, stable under deformations. The way in which curvature maxima change has, in fact, been used to infer processes that have acted upon the object [52].

Certain classes of curves can be represented as a single function. Curves expressible as a function of one coordinate,  $y(x)$ , are perpendicular deformations of a straight line, that is, deformations of a line only in the direction perpendicular to the line. Curves expressible as a function of angle,  $r(\theta)$ , are radial deformations of a circle.

Surfaces are more problematic in that there is no natural ordering of points on an arbitrary surface. Again, certain classes of surfaces can be represented as a single function. Surfaces expressible as a function of two coordinates,  $z(x, y)$ , are perpendicular deformations of a plane. Surfaces expressible as a function of two angles,  $r(\theta, \phi)$ , are radial deformations of a sphere. If the proper parameters,  $u$  and  $v$ , can be found, an arbitrary surface can be represented by  $x(u, v)$ ,  $y(u, v)$  and  $z(u, v)$ . A surface parametrization based on this representation will be discussed in Chapter 7.

Instead of representing the coordinates, it may be advantageous to represent a differential property of the surface such as tangent or curvature. This is the basis for chain codes (discrete tangent) [31],  $\psi(s)$  (continuous tangent) [10] and the extended Gaussian image (Gaussian curvature) [43]. These differential properties allow rotation and scale changes easily. A potential disadvantage of differential representations is that numerical integration is needed to recover the object.

### 3.2.2 Extended Gaussian Image

The extended Gaussian image [43] surface representation is the reciprocal of Gaussian curvature, or radius of curvature, for each tangent direction. That is:

$$G(\theta, \phi) = \frac{1}{K(u, v)} \quad (3.1)$$

where  $(u, v)$  specifies the point on the surface of the object that has the same orientation as the point on the unit sphere specified by  $(\theta, \phi)$ . In the discrete form, this becomes a surface orientation histogram. Overall scale and orientation of the object are easily handled. This representation, however, is limited to convex objects. Non-convex objects must be segmented into convex pieces in order for the representation to be unique. The two-dimensional analog is the extended circular image [42]. Although it has been used for two-dimensional shape matching [61], it still requires segmentation into convex segments.

### 3.2.3 Splines

Interpolating splines are a useful curve or surface representation. They are piecewise polynomials with specifiable continuity properties. They are most often used in graphics or design. For a particular choice of spline representation, the shape is determined by a set of control points. Moving a control point will change the curve locally. The different kinds of splines are primarily distinguished by their shape and continuity properties, the classes of curves (or surfaces) they can represent and the number of parameters they require. Splines have more in common with direct representations than parametric representations because the parameters represent positions on the object. The spline formalism is a direct representation restricted to certain desired continuity requirements.

It has the same disadvantages of direct representations in that the parameters (control points) do not represent shape, but represent positions. To achieve a change in viewpoint, the control points must be transformed.

Bending energy minimizing splines (or smoothing splines) are a way of interpolating by minimizing the weighted sum of the overall bending energy of a shape and the distance of the shape to the data. They have been used by Kass *et al.* [49], Terzopoulos *et al.* [86] and others. While single points globally change the curve, the effect is still primarily local and thus they behave very similarly to interpolating splines.

### 3.2.4 Sweeps

Sweeps (or generalized cylinders) are a way of representing elongated objects [2, 65]. Typically, this means a one-dimensional curve defines the spine of the object and a two-dimensional cross-section is swept along the spine to define the surface. This cross-section can also be made to vary along the spine. The actual properties of this representation depend on the choices of spine and cross-section. Practical choices usually limit the class of object that is representable.

### 3.2.5 Superquadrics

Superquadrics [11, 7] (in two dimensions, superellipses) are an extension of quadrics using an exponent that varies the squareness of the shape. Superellipses can be expressed parametrically by:

$$\mathbf{x}(t) = \mathbf{x}_0 + \begin{bmatrix} a \cos^q(t) \\ b \sin^q(t) \end{bmatrix} \quad (3.2)$$

The exponent  $q$  controls the squareness as the shape varies from an ellipse ( $q = 1$ ) to a rectangle in the limit ( $q = 0$ ). The implicit form for points on the superellipse is:

$$\frac{|x - x_0|}{a^{2/q}} + \frac{|y - y_0|}{b^{2/q}} = 1 \quad (3.3)$$

One of the advantages of the implicit form is that it can be used as the basis of an inside-outside function. For points inside the superellipse, the left hand side of Equation 3.3 is less than one; for points outside, it is greater than one. These formulas can be extended to three dimensions. A limitation of superquadrics is that the parameters representing the shape cannot be extracted immediately from a direct representation but must be found through an optimization procedure. The basic shape can be altered by such operations as twisting, bending and tapering [12], as can any explicit representation. Using these operations, superquadrics can be thought of as a kind of sweep representation. The main disadvantage of superquadrics is that even with these altering operations, superquadrics are limited by their convex, doubly symmetric cross-section and thus still only represent a very limited family of shapes (without resorting to composition).

### 3.2.6 Fourier

Fourier representations are those that express the curve or surface in terms of an orthonormal basis. The motivation for a basis representation is that it allows us to express any object as a weighted sum of a set of known functions. An orthonormal set is desirable because it makes the parameters (weights) distinct. If two of the basis functions were similar, representing features corresponding to their difference would require their coefficients to be large in magnitude and opposite in sign. This would make the coefficient determination difficult. Representing features corresponding to their average direction



would require their coefficients to be roughly equal. This redundancy is inefficient. Both of these problems can be avoided if the basis functions are as dissimilar as possible, that is, orthogonal.

To express the function  $X(t)$  on the interval  $(a, b)$  in terms of the basis  $\phi_k(t)$ , we write:

$$X(t) = \sum_{k=1}^{\infty} p_k \phi_k(t) \quad (3.4)$$

where:

$$p_k = \int_a^b X(t) \phi_k(t) dt \quad (3.5)$$

The coefficients  $p$ , the projections of the function onto the  $k$  basis functions, are the parameters of the representation. In order to use this representation, however, the sum must be truncated. In most such representations the higher indexed basis functions represent higher spatial variation. Therefore, if the function to be represented is expected to have limited spatial variation, as is the case for most real object boundaries, the series can be truncated and still accurately represent the function. In fact, the truncated Fourier series is optimal in the sense of mean square error. The mean square error between the function and its approximation as a linear combination of orthogonal basis functions is minimized when the coefficients are the Fourier coefficients,  $p_k$  [87].

The usual basis functions are the sinusoids [71], although others, such as those based on orthogonal polynomials, are possible. The sinusoids have the advantage of representing the familiar notion of frequency. The various representations of curves usually differ in the choice of direct representation on which to base the decomposition. For

curves, the common choices are  $r(\theta)$ ,  $\psi(s)$ ,  $[x(s), y(s)]$  and  $x(s) + iy(s)$ , as discussed in Section 3.2.1. The use of Fourier representations of contours has been limited primarily to classification applications such as character recognition [37] and airplane silhouettes [62].

The  $r(\theta)$  representation limits the curves to radial ones.  $\psi(s)$  is not a good Fourier representation because it is differential. Curves reconstructed from a truncated series for a closed curve using the  $\psi(s)$  representation may not be closed. Using a differential representation also exacerbates the problem of corners because they become discontinuities. The  $(x(s), y(s))$  and  $x(s) + iy(s)$  representations are completely expressive and do not have the problems of differential representations.

Spherical harmonics have been used [78, 10] as a type of Fourier surface representation for radial surfaces  $(r(\theta, \phi))$ . The basis functions are based on sinusoids and Legendre polynomials.

### 3.3 The Deformable Contour Parametrization

Fourier parametrizations are the most suitable for this work for a number of reasons. They are a concise boundary representation. They are generic, that is, not limited to a particular class of objects. The conversion between the parametrization and the shape is easily and directly computable. By using a truncated series, we limit the frequency content of the curve and thus enforce a smoothness constraint.

#### 3.3.1 Closed Curves

The standard real Fourier representation is based on Equations 3.4 and 3.5, using the sinusoids or trigonometric functions as the basis functions. The basis can be used, in

conjunction with a direct representation, to form a parametrization for closed curves.

The basis is:

$$\phi = \left\{ \frac{1}{2\pi}, \frac{\cos x}{\pi}, \frac{\sin x}{\pi}, \frac{\cos 2x}{\pi}, \frac{\sin 2x}{\pi}, \frac{\cos 3x}{\pi}, \frac{\sin 3x}{\pi}, \dots \right\} \quad (3.6)$$

Closed curves are useful for representing organs, cells and other objects that are delineated by a complete boundary. A closed curve can be represented by two periodic functions of  $t$ , where  $t$  varies from 0 to  $2\pi$ ,  $x(t)$  and  $y(t)$ . If we then take the Fourier decomposition of these two functions using the sinusoidal basis and write the resulting equations in matrix form, we get the elliptic Fourier representation [35, 51, 53]:

$$v(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \quad (3.7)$$

where:

$$\begin{aligned} a_0 &= \frac{1}{2\pi} \int_0^{2\pi} x(t) dt \\ c_0 &= \frac{1}{2\pi} \int_0^{2\pi} y(t) dt \\ a_k &= \frac{1}{\pi} \int_0^{2\pi} x(t) \cos kt dt \\ b_k &= \frac{1}{\pi} \int_0^{2\pi} x(t) \sin kt dt \\ c_k &= \frac{1}{\pi} \int_0^{2\pi} y(t) \cos kt dt \\ d_k &= \frac{1}{\pi} \int_0^{2\pi} y(t) \sin kt dt \end{aligned}$$

The closed curve is thus represented by the parameters  $a_0, c_0, a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, \dots$ , which will be referred to as the raw parameters. This particular version of Fourier boundary representation has a number of advantages. Both the decomposition and the reconstruction can be calculated efficiently due to the fast Fourier transform. A geometric interpretation, in terms of ellipses, can be developed from this decomposition. The geometric interpretation will allow for better visualization of the effect of the parameters. In addition, from this geometric interpretation, invariance to rotation, scale, translation and starting point can also be achieved. Invariance to rotation, scale and translation is important because these parameters are determined not by the object but by the view of the object, which often cannot be held constant. From Equation 3.7, the relationship between the parameters and the curve can be seen to be continuous and unique, except for the arbitrary starting point of the boundary functions  $x(t)$  and  $y(t)$ . Since the starting point is arbitrary, it will be useful to remove its effect.

In Equation 3.7, the first two coefficients,  $a_0$  and  $c_0$ , determine the overall translation of the shape. Each term in the summation is the parametric form for an ellipse. The degenerate case occurs when  $a_k d_k - b_k c_k = 0$ , in which case each term is the parametric form for a straight line (a degenerate ellipse). In each term, the matrix therefore determines the characteristics of the ellipse. The contour can be viewed as being decomposed into a sum of rotating phasors, each individually defining an ellipse, and rotating with a speed proportional to their harmonic number,  $k$ . This can be seen in Figure 3.1 where a contour is shown constructed from three component ellipses forming a sort of planetary system. The straight lines represent the phasors for each ellipse shown at three different times. Thus, the point  $C_{ij}$  traces out the  $i$ th ellipse at time  $j$ . Each point is the center of the next higher ellipse.  $C_0$  is the center of the first ellipse. Points  $C_{31}, C_{32}$  and  $C_{33}$  are

three different points on the final curve. The elliptic Fourier representation can be seen as a generalization or extension to pure ellipse representations [15].

It is important that the curve representation that is decomposed into Fourier components be both continuous and periodic. While the Fourier series will always converge (except for pathological cases), discontinuities will slow the convergence because of the high frequencies inherent in a step jump. The function values should also match at the endpoints of the interval for the same reason. Both  $x(t)$  and  $y(t)$  are periodic because the contour is closed, and both  $x(t)$  and  $y(t)$  are continuous because the contour is continuous. Thus, both of these properties are always satisfied.

In Equation 3.7, we can make  $t$  correspond to arclength by taking:

$$t(s) = \frac{2\pi s}{S} \quad (3.8)$$

where  $s$  is arclength along the curve from the starting point and  $S$  is the total arclength of the curve. However, when a curve is reconstructed from a truncated series, the  $x$  and  $y$  values will differ from the original values. This difference means that Equation 3.8 no longer holds. This can be seen by noting that from Equations 3.7 and 3.8:

$$\left| \frac{dv}{dt} \right| = \frac{ds}{dt} = \frac{S}{2\pi} \quad (3.9)$$

However, we also know that:

$$\left| \frac{dv}{dt} \right| = \sqrt{\left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2} \quad (3.10)$$

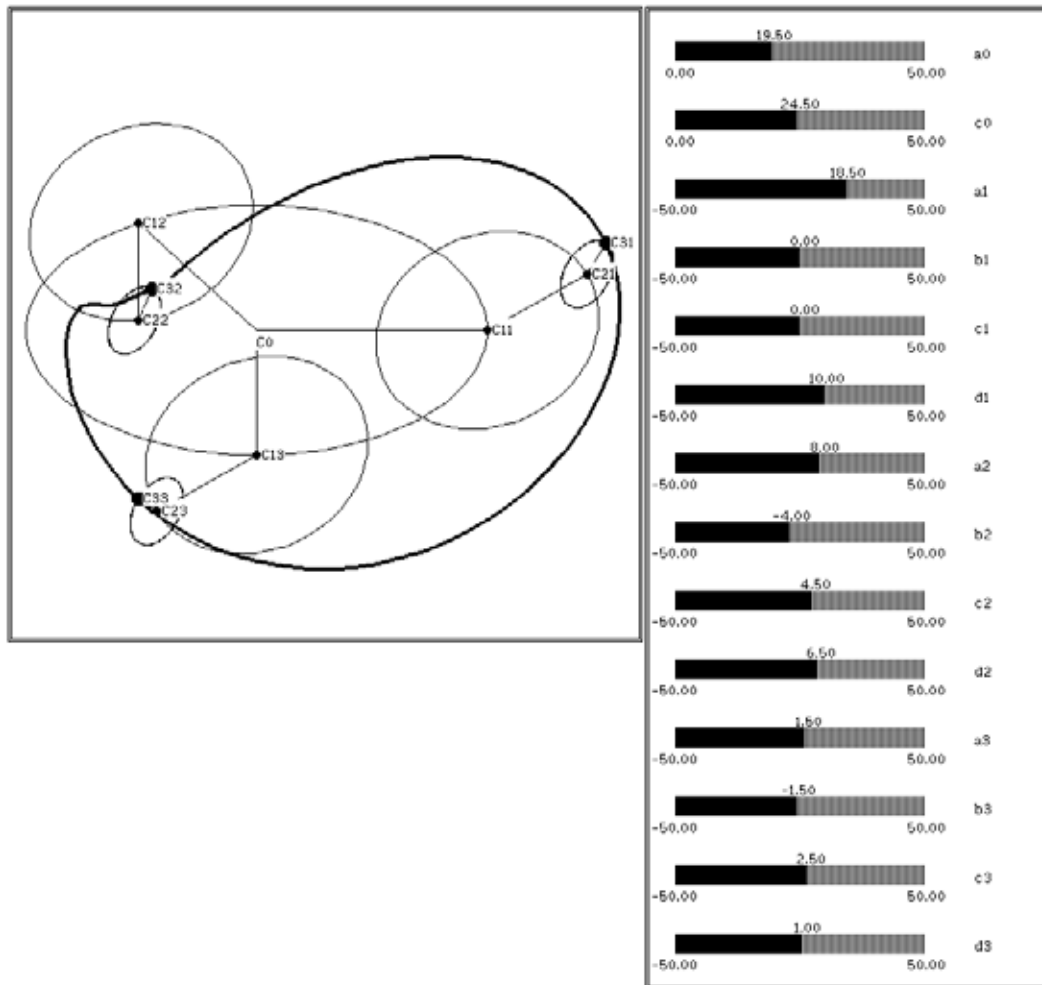


Figure 3.1: The contour (dark line) at the left is constructed from three component ellipses shown at three different times. The associated raw parameter values are shown in the bar graph at the right, where the limits of the scale for each parameter are shown below the bar and the value of the parameter is shown above each bar.

This expression is a complicated function of the parameters and is only equal to  $\frac{S}{2\pi}$  for circles and for the infinite series [71]. Thus, while Equation 3.8 can be used for the purposes of decomposition, it will not be true in general. A numerical approximation to this relationship will be discussed in Section 5.2.

The starting point,  $t = 0$ , is, in general, arbitrary for a closed curve. A different starting point will result in a different parametrization. The starting point can be standardized so that it does not affect the parametrization. This will be discussed in the next section.

The direction of rotation, clockwise versus counterclockwise, must be chosen consistently because the parameters depend upon it. In addition, if direction information is used from the boundary measure (see Chapter 4), the direction of rotation must agree with the boundary direction. This depends, for example, on whether the object of interest is darker or brighter than its surroundings.

### **Ellipse Parameters**

More geometrically meaningful parameters can be derived from this representation that will allow invariance to rotation, scale, translation and starting point. The geometric properties of each of the component ellipses can be derived from the raw elements of each ellipse matrix. Each ellipse can be described by four geometric properties: semi-major axis length, semi-minor axis length, rotation and phase shift. The rotation is the angle from the  $x$ -axis to the major axis of the ellipse, defined from  $-\pi/2$  to  $\pi/2$ . The phase shift is the difference in phase from the major axis to the position of  $t = 0$  (the ellipse starting position), defined from  $-\pi$  to  $\pi$ .

These ellipse properties can be derived as follows. First consider the general form for an ellipse, which is the product of the raw ellipse matrix and the trigonometric basis

function vector:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \quad (3.11)$$

In order to determine the ellipse parameters, consider an ellipse with its major axis aligned with the  $x$ -axis and with no phase shift. The matrix for this ellipse is:

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \quad (3.12)$$

where  $A$  and  $B$  are the major and minor semi-axis lengths, respectively. The phasor moves counterclockwise for  $B$  positive, clockwise for  $B$  negative. The ellipse can be rotated simply by pre-multiplying the ellipse matrix by a rotation matrix. A phase shift of the ellipse by  $\phi_0$  means replacing  $t$  by  $t + \phi_0$ . This is the same as a pre-multiplication of the basis function vector by a rotation matrix, or equivalently, a post-multiplication of the ellipse matrix. Thus, a rotation of this ellipse by  $\theta$  and shift by  $\phi$  can be written as a pre-multiplication and a post-multiplication by rotation matrices:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \quad (3.13)$$

This is equal to:

$$\begin{bmatrix} A \cos \theta \cos \phi - B \sin \theta \sin \phi & -A \cos \theta \sin \phi - B \sin \theta \cos \phi \\ A \sin \theta \cos \phi + B \cos \theta \sin \phi & -A \sin \theta \sin \phi + B \cos \theta \cos \phi \end{bmatrix}$$



This represents a general ellipse and is thus equivalent to the raw ellipse matrix in Equation 3.11. Therefore, to find the ellipse parameters given the values of these matrix elements, solve the following four equations that come from identifying corresponding matrix elements for  $A$ ,  $B$ ,  $\theta$  and  $\phi$ .

$$\begin{aligned}
 a &= +A \cos \theta \cos \phi - B \sin \theta \sin \phi \\
 b &= -A \cos \theta \sin \phi - B \sin \theta \cos \phi \\
 c &= +A \sin \theta \cos \phi + B \cos \theta \sin \phi \\
 d &= -A \sin \theta \sin \phi + B \cos \theta \cos \phi
 \end{aligned} \tag{3.14}$$

This results in:

$$\begin{aligned}
 A^2 &= \frac{\alpha + \sqrt{\alpha^2 - 4\beta^2}}{2} \\
 B^2 &= \frac{2\beta^2}{\alpha + \sqrt{\alpha^2 - 4\beta^2}}
 \end{aligned} \tag{3.15}$$

where

$$\alpha = a^2 + b^2 + c^2 + d^2, \quad \beta = ad - bc \tag{3.16}$$

and

$$\begin{aligned}
 \theta &= \tan^{-1} \frac{Ac + Bb}{Aa - Bd} \\
 \phi &= \tan^{-1} \frac{Ba - Ad}{Ac + Bb}
 \end{aligned} \tag{3.17}$$

By taking  $A$  to be positive and  $B$  to agree in sign with  $j$ , we get a consistent sign convention. This set of parameters,  $a_0, c_0, A_1, B_1, \phi_1, \theta_1, A_2, B_2, \phi_2, \theta_2, \dots$ , represent the shape in terms of the ellipse properties and will be referred to as the refined parameters. This conversion to refined parameters shown in Equations 3.14, 3.15, 3.16 and 3.17 is continuous and unique.

A further conversion can improve this set by making the rotation and shift parameters relative values and normalizing the axes' lengths. This is useful because the refined parameters all represent absolute quantities as opposed to quantities relative to the preceding harmonic. Converting to relative quantities will allow the isolation of an overall rotation parameter. In addition, this conversion will enable the removal of the overall phase shift,  $\phi_1$ , which is determined by the start of the parametrization along the curve and is thus arbitrary. The use of relative rotation and phase parameters means that changing any one of these parameters will not change the relative relationships of any of the other parameters. Normalizing the axes' lengths creates an overall scale parameter.

First consider rotation. A rotation of  $\theta_0$  to the object results in a rotation of  $\theta_0$  on each ellipse and therefore  $\theta_k$  becomes  $\theta_k + \theta_0$ . This is because each rotation,  $\theta_k$ , represents an absolute rotation from the  $x$ -axis. In order to convert to relative rotations, take the difference between successive rotations:

$$\theta'_k = \theta_k - \theta_{k-1} \tag{3.18}$$

To convert back, use:

$$\theta_k = \sum_{l=1}^k \theta'_l \tag{3.19}$$

In this way, one parameter,  $\theta'_1$ , determines the overall rotation of the object and each rotation value,  $\theta'_k$  represents the rotation relative to the preceding harmonic. Now a rotation of  $\theta_0$  to the object changes  $\theta_1$  to  $\theta_1 + \theta_0$  and leaves the other parameters unchanged.

The conversion of the phase shifts to relative values is done in two parts. First, the overall phase shift is removed, since it is arbitrary, and then the relative values are calculated. A phase shift of  $\phi_0$  applied to the boundary is equivalent to replacing  $t$  by  $t + \phi_0$ . For the  $k$ th ellipse, this results in a shift of  $k\phi_0$ . The overall phase shift of the object can thus be removed by subtracting its effects from the other harmonics and then setting it to zero. That is:

$$\begin{aligned}\phi_k^* &= \phi_k - k\phi_1 \\ \phi_1^* &= 0\end{aligned}\tag{3.20}$$

The parameters  $\phi_k^*$  represent the absolute phase shifts with the overall phase set to zero. Thus, a phase shift of  $\phi_0$  applied to the object has no effect on the values of  $\phi_k^*$ . These values can now be converted to and from relative values in the same way as for the rotations:

$$\begin{aligned}\phi'_k &= \phi_k^* - \phi_{k-1}^* \\ \phi_k^* &= \sum_{l=1}^k \phi'_l\end{aligned}\tag{3.21}$$

The parameters  $\phi'_k$  represent the relative phase shifts.

The axes lengths can be normalized to the first major axis in order to isolate a

single parameter that determines the overall scale:

$$\begin{aligned} A'_k &= \frac{A_k}{A_1} \quad \text{for } k \neq 1 \\ B'_k &= \frac{B_k}{A_1} \end{aligned} \tag{3.22}$$

This set of parameters,  $a_0, c_0, A'_1, B'_1, \theta'_1, A'_2, B'_2, \phi'_2, \theta'_2, \dots$ , expresses the boundary in terms of the relative ellipse properties. The further conversion to relative parameters shown in Equations 3.18, 3.19, 3.21 and 3.22 is both continuous and unique, except that the starting point ambiguity has been removed. We have explicit equations for the conversion between the raw coefficients of the Fourier expansion and the refined and relative ellipse parameters. Figure 3.2 shows a contour constructed from three component ellipses defined from their relative parameters (as in Figure 3.1).

We can treat the above parameter set as a single vector  $\mathbf{p}$  and will distinguish between the raw, refined and relative parametrizations, using  $K$  harmonics, by referring to them as follows:

$$\mathbf{p}_{\text{raw}} = (a_0, c_0, a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, \dots, a_K, b_K, c_K, d_K) \tag{3.23}$$

$$\mathbf{p}_{\text{ref}} = (a_0, c_0, A_1, B_1, \theta_1, \phi_1, A_2, B_2, \theta_2, \phi_2, \dots, A_K, B_K, \theta_K, \phi_K) \tag{3.24}$$

$$\mathbf{p}_{\text{rel}} = (a_0, c_0, A'_1, B'_1, \theta'_1, A'_2, B'_2, \theta'_2, \phi'_2, \dots, A'_K, B'_K, \theta'_K, \phi'_K) \tag{3.25}$$

### 3.3.2 Open Curves

The elliptical Fourier descriptors can be used for open curves. Open curves are useful for representing objects or parts of objects that do not have a complete boundary such as organs with openings, or blood vessels. The curve can be represented as before by two

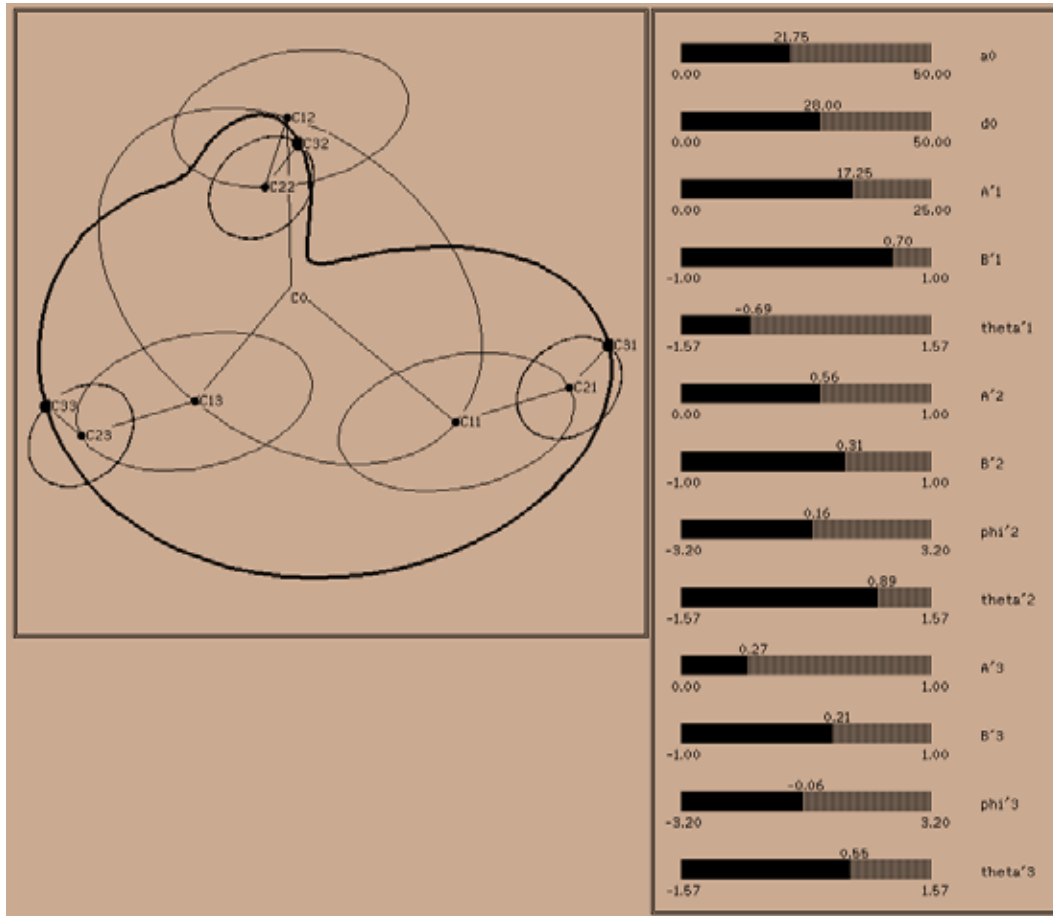


Figure 3.2: The contour (dark line) at the left is constructed from three component ellipses shown at three different times. The associated relative parameter values are shown in the bar graph at the right, where the limits of the scale for each parameter are shown below the bar and the value of the parameter is shown above each bar.

functions,  $x(t)$  and  $y(t)$ , but since the curve is open, a straightforward representation of the curve would result in a discontinuity.

Analogously to Persoon and Fu [71], this discontinuity can be avoided by having the parameter  $t$  start at one end of the line, trace along the contour to the other end, and then retrace the curve in the opposite direction to create a closed path. That is:

$$x(t) = x(2\pi - t) \tag{3.26}$$

$$y(t) = y(2\pi - t) \tag{3.27}$$

This results in functions  $x(t)$  and  $y(t)$  that are even and thus their Fourier sine terms,  $b_k$  and  $d_k$ , are zero. The converse, namely that any elliptic Fourier expansion with  $b_k$  and  $d_k$  equal to zero for all  $k$  results in an even function and thus describes an open curve, is also true.

We can thus represent an arbitrary even function in terms of a sinusoidal basis because this basis can be divided into even and odd functions. In order to represent even functions, we restrict the basis functions to include only even ones.

$$\phi_{\text{even}} = \left\{ \frac{1}{2\pi}, \frac{\cos x}{\pi}, \frac{\cos 2x}{\pi}, \frac{\cos 3x}{\pi}, \dots \right\} \tag{3.28}$$

This parametrization can then be used for image features best described as curved line segments. This representation can be thought of as decomposing the line into degenerate ellipses (flattened down to two coincident lines). The equations for the corresponding

ellipse parameters are then just:

$$\begin{aligned} A^2 &= a^2 + c^2 \\ B^2 &= 0 \end{aligned} \tag{3.29}$$

and

$$\begin{aligned} \theta &= \tan^{-1} \frac{c}{a} \\ \phi &= 0 \end{aligned} \tag{3.30}$$

The ellipses are all degenerate with a fixed starting point at one end, thus forcing both the minor semi-axis length,  $B$ , and the starting point,  $\phi$ , to be zero. The relative transformations for  $\theta_k$  and  $A_k$  of the previous section can also be applied.

Considering these parameters as a single vector  $\mathbf{p}$ , we have:

$$\mathbf{p}_{\text{openraw}} = (a_0, c_0, a_1, c_1, \dots, a_K, c_K) \tag{3.31}$$

$$\mathbf{p}_{\text{openref}} = (a_0, c_0, A_1, \theta_1, \dots, A_K, \theta_K) \tag{3.32}$$

$$\mathbf{p}_{\text{openrel}} = (a_0, c_0, A'_1, \theta'_1, \dots, A'_K, \theta'_K) \tag{3.33}$$

### 3.3.3 Number of Harmonics

The summation in Equation 3.7 must, in practice, be truncated. This truncation limits the number of parameters and thus is necessary for a concise representation. Also, by eliminating the higher harmonics, the curve is smoothed. The truncation also, however,

decreases the accuracy of the representation.

Smoothing is necessary in order to reduce noise. Limiting the number of harmonics limits the representation to smooth objects and thus constrains the boundary to be smooth by excluding shapes with higher frequency variation. This is analogous to regularization for ill-posed problems [73]. In regularization, a functional is devised that incorporates a smoothness constraint. Here, the solution space is directly restricted to allow only smooth solutions. Smoothing by reconstructing a truncated elliptic Fourier representation is, in general, a good method for smoothing curves that eliminates the problem of shrinkage [56] caused by direct filtering methods. Using too many harmonics could unnecessarily increase the computational complexity. The number of harmonics necessary for an accurate representation increases with the true high frequency content of the contour. Harmonics higher than the true frequency content of the contour would be due to noise. The probability distributions associated with their coefficients would be clustered about zero, thus minimizing their contribution. The choice of number of harmonics is thus a tradeoff between desired accuracy, conciseness and degree of smoothing. Many biological forms are relatively smooth and unconvoluted and thus are well represented by a small number of components.

Giardina and Kuhl [35] derive a bound on the error in representing a contour. Let  $v = (x, y)$  be the true values for the coordinates of the curve and  $v_K = (x_K, y_K)$  be the approximation to the coordinates associated with the series truncated at  $K$  components. The error is defined to be the maximum of the individual coordinate errors:

$$\epsilon = \max \left[ \sup_t |x(t) - x_K(t)|, \sup_t |y(t) - y_K(t)| \right] \quad (3.34)$$



The bound on the error that they derive for the series truncated at  $K$  harmonics can be written as:

$$\epsilon \leq \frac{2}{N\pi} \max \left[ \int_0^{2\pi} \left| \frac{d^2 x(t)}{dt^2} \right| dt, \int_0^{2\pi} \left| \frac{d^2 y(t)}{dt^2} \right| dt \right] \quad (3.35)$$

The important features of this bound on the error are that it is inversely proportional to the number of harmonics and proportional to the integral of the absolute value of the second derivative. Empirically, Kuhl and Giardina found this bound to be too conservative on a variety of shapes by at least a factor of two [51]. In fact, the more complex the boundary, the more conservative the bound. For this reason this bound is not a practical method for determining the number of harmonics needed to accurately represent a shape.

Instead, the appropriate number of harmonics to use for the representation can be determined directly because of the prior information about the shape. As will be discussed in Section 3.4, a sample set of boundaries must be obtained. Each boundary is reconstructed varying the number of harmonics, and the error from the boundary is measured. From this procedure, the number of harmonics necessary to reconstruct these curves within a fixed error bound is determined. For most of the examples considered, between four and six terms of the expansion have been used. Figure 3.3 shows a typical curve. The error associated with limiting the number of harmonics is plotted in Figure 3.4. The error is defined by Equation 3.34. For this shape, four harmonics are enough to make the error less than one pixel.

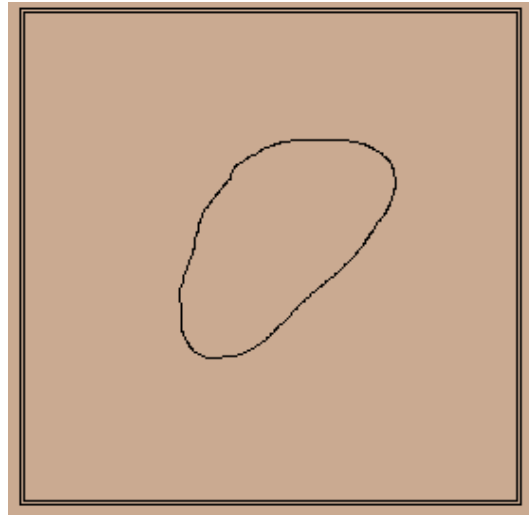


Figure 3.3: Curve for reconstruction error example.

### 3.4 The Parameter Probability Distributions

The probability distributions associated with the parameters are intended to bias the model towards a particular range of shapes. This prior knowledge comes from experience with images of the object being delineated. The images will differ due to variability in the object shape and the view of the object.

While consistency of object shape is often a reliable assumption, leading to peaked distributions for the governing parameters, consistency of view may be harder to come by. Arbitrary translation and rotation are often unavoidable. This will lead to very broad distributions for the translation and rotation parameters and will make the optimization process more difficult, as will be seen in Chapter 5.

Prior information may not be available at all. In these cases, uniform distributions are used for the prior probabilities of the parameters. It will be necessary, however, to

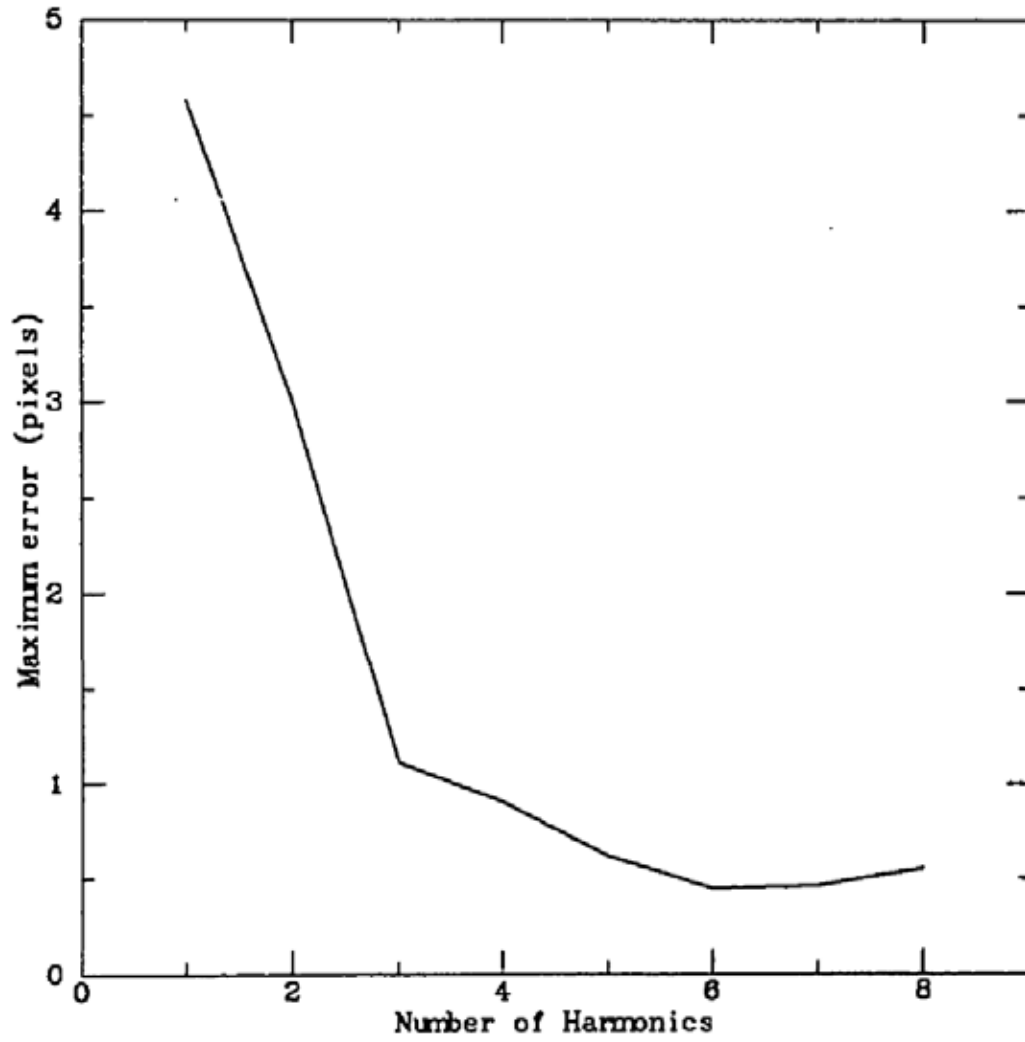


Figure 3.4: Reconstruction error example.

supply an initial estimate of the boundary for the optimization process, described in Chapter 5.

Prior information can be derived from a sample of images of instances of the object, when such a sample is available. The prior probability distributions can be estimated from this sample. The distributions for each of the parameters of the model can be estimated from the shapes determined from the sample by decomposing the boundaries into their model parameters and collecting statistics. In order to calculate these statistics, the boundaries of the objects must be determined. Manual segmentation is one way of determining the shape for the sample. This is a good option when such segmentation has already been performed. Any errors in the manual segmentation will be averaged over the sample and therefore, unless there is significant bias, their effect will be small. Alternatively, this method can be run on a set of exemplar images with manual initialization and uniform distributions.

If a particular distribution is known to govern the parameters, it can be used as the prior probability, although if it is not unimodal it will make the optimization difficult (see Chapter 5). Otherwise, if mean and variance information is known, a multivariate Gaussian can be used for the  $N$  parameters:

$$P(\mathbf{p}) = \left( \frac{1}{(2\pi)^{N^2} |\mathbf{C}_{pp}|} \right)^{1/2} e^{-(1/2)(\mathbf{p}-\mathbf{p}_0)^T \mathbf{C}_{pp}^{-1} (\mathbf{p}-\mathbf{p}_0)} \quad (3.36)$$

The vector  $\mathbf{p}_0$  is the mean of  $\mathbf{p}$  and  $\mathbf{C}_{pp}$  is the covariance matrix of  $\mathbf{p}$ .

The Gaussian is the natural form for a probability density. It was first discussed by De Moivre in 1733 as the limit of the binomial distribution [24]. The Central Limit Theorem shows that additive random variables converge to a Gaussian distribution under

fairly general conditions [69]. The use of the Gaussian may be understood in terms of information theory because, among probability density functions having a given variance, the Gaussian is the one with the maximum entropy [20, 79]. The Gaussian density follows directly from knowing no information other than a mean and a variance. For this work, the parameters are taken to be independent. The above distribution simplifies to:

$$P(\mathbf{p}) = \prod_{i=1}^N P(p_i) = \prod_{i=1}^N \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(p_i - m_i)^2}{2\sigma_i^2}} \quad (3.37)$$

Here,  $m_i$  is the mean of  $p_i$  and  $\sigma_i^2$  is the variance. The parameters in the model may not be independent. For example, a particular object could tend to be smoother when it is larger. This would lead to a correlation between some of the parameters. This effect will not be exploited in this work although it could easily be incorporated by accumulating the appropriate statistics to construct  $\mathbf{C}_{pp}$ .

An example distribution is shown in Figures 3.5 and 3.6. The curve corresponding to the mean parameter values is the middle curve shown in Figure 3.5. Above and below it are the curves corresponding to the mean parameter values plus and minus one standard deviation. The standard deviation for each parameter is shown in Figure 3.6.

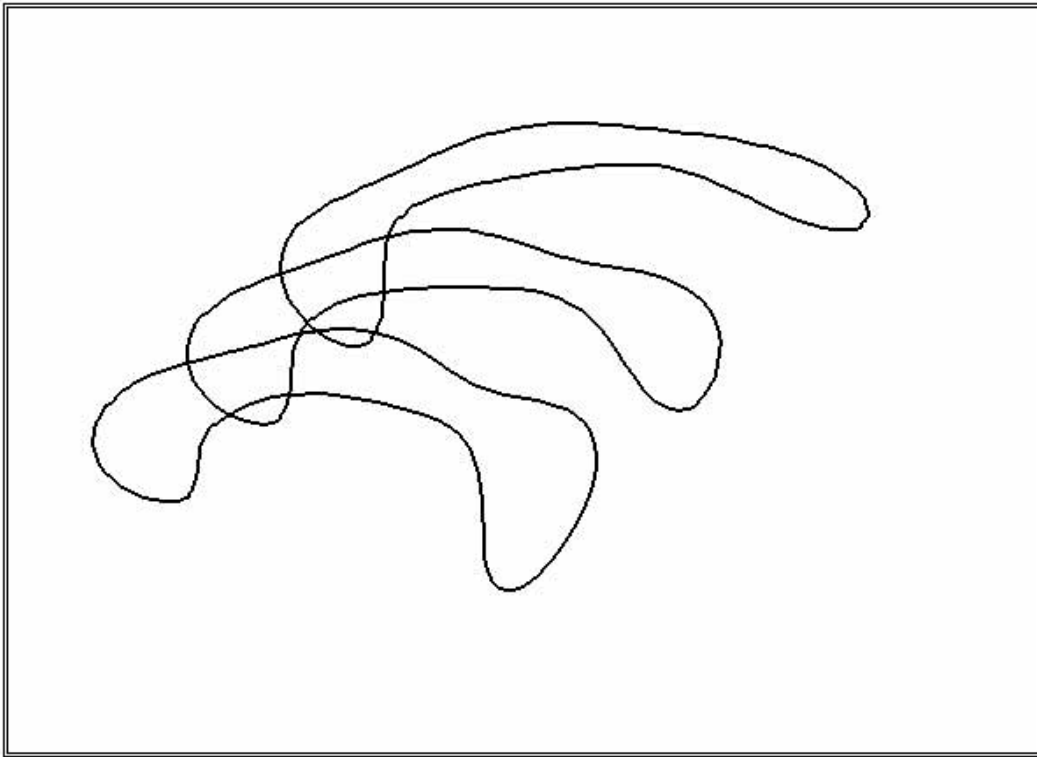


Figure 3.5: Example mean curve, shown with curves corresponding to parameters plus and minus one standard deviation.

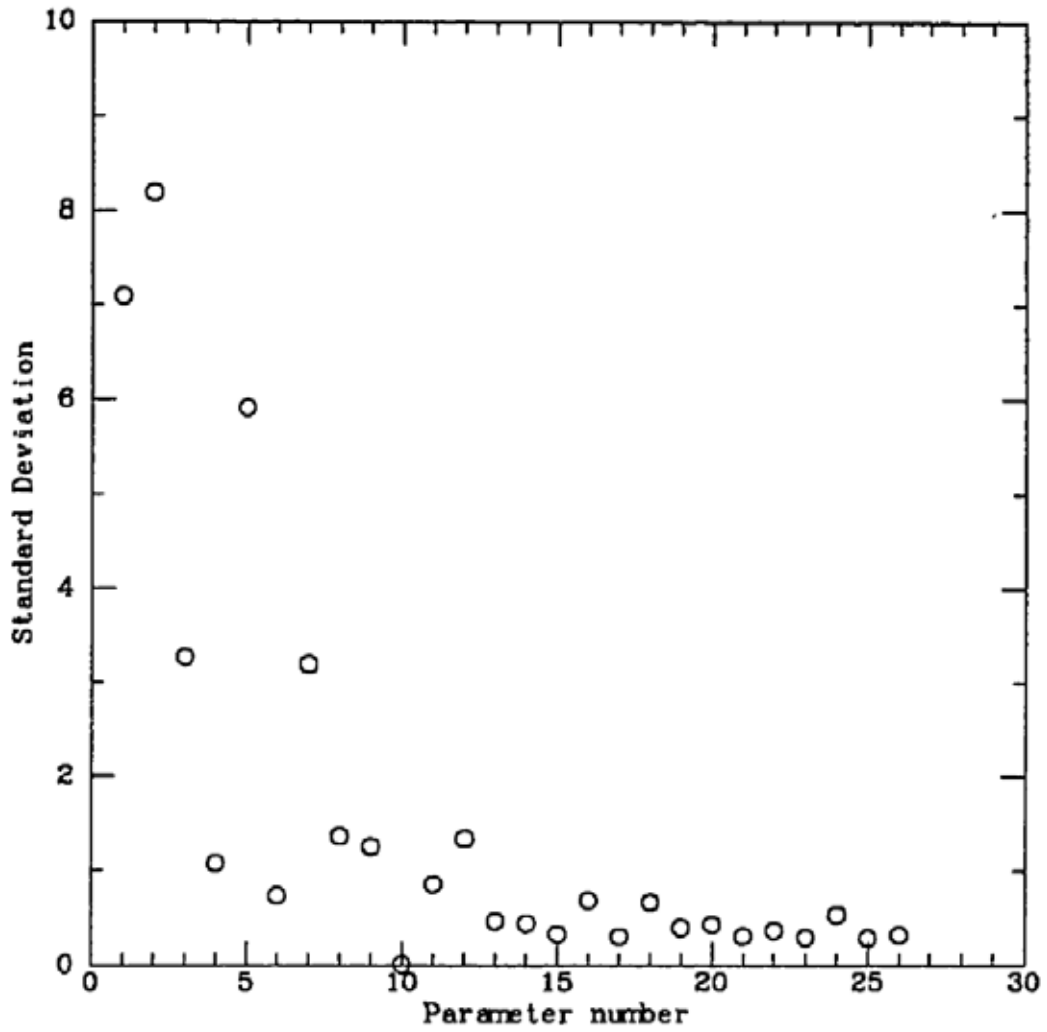


Figure 3.6: Example parameter standard deviations.

## Chapter 4

# Matching

Matching in computer vision is the process of relating representations by bringing them into correspondence [10]. In geometric modeling, we need both a representation for information derived from the image data and a representation for prior knowledge of the image domain. These representations will be designed such that they express the relevant prior and image-derived information. They must also be commensurable so that we can determine the correspondence between them. The correspondence established represents an interpretation for the image by reconciling it with prior information. In this work, the correspondence is determined by optimization. This means either minimizing a measure of mismatch or maximizing a measure of match between the two representations. For a parametric model, the correspondence is achieved by varying the parameters.

We need an appropriate measure derived from the image that can be compared to our model. This will be discussed in Section 4.4. We also need a measure of match between the image measure and the geometric model. This will be discussed in Section 4.1. Chapter 5 will discuss the optimization method.



## 4.1 Match Measures

There are many ways in which to measure the degree of match between the model and the image. In the case where the measure is just intended to compare the function values over the region of the image, it usually involves some type of correlation or sum of squared differences (SSD) measure [76]. The sum of squared differences mismatch measure for two functions  $t$  and  $b$  over an area  $\mathcal{A}$  can be written as:

$$\begin{aligned} \text{SSD}(t, b) &= \iint_{\mathcal{A}} (t(x, y) - b(x, y))^2 dx dy \\ &= \iint_{\mathcal{A}} (t^2(x, y) - 2t(x, y)b(x, y) + b^2(x, y)) dx dy \end{aligned} \quad (4.1)$$

The corresponding correlation match measure is:

$$C(t, b) = \iint_{\mathcal{A}} t(x, y)b(x, y) dx dy \quad (4.2)$$

If  $\iint_{\mathcal{A}} t^2(x, y) dx dy$  and  $\iint_{\mathcal{A}} b^2(x, y) dx dy$  are constant, the above two measures are equivalent. When  $t$  is a template smaller than  $b$  whose position in  $b$  is sought, then  $\iint_{\mathcal{A}} b^2(x, y) dx dy$  will not be a constant as  $\mathcal{A}$  varies over different portions of  $b$ . In this case the correlation is sometimes normalized by either the mean of  $b$  in the area or the variance of  $b$ . In the present case, where  $t$  is derived from the model and  $b$  is derived from the image,  $\iint_{\mathcal{A}} b^2(x, y) dx dy$  is constant because  $\mathcal{A}$  represents the entire image but  $\iint_{\mathcal{A}} t^2(x, y) dx dy$  varies with the length of the contour. It will be shown in Section 4.3 that even in this case, these measures are essentially equivalent.

The above measures generalize directly to vector valued functions. Scalar differences become vector differences and scalar products become dot products.

## 4.2 Boundary Justification

To match the geometric model to the image, we can use either the area of the object under consideration or the boundary. In this work, the feature of interest is shape and thus the boundary of the object is the goal and the focus of the analysis. The object is expected to be distinguished from the background by some measure of boundary strength, which must be computed from the image. Only the boundary of the shape is modeled and the matching process only considers the boundary. When the entire area of the object is considered, the internal structure of the object can be included as part of the match. While the internal gray level structure of some objects is a reliable and distinguishing feature, more often it is either textural or trivially related to the boundary (e.g., a bright border area).

The boundary value will be considered constant along the entire boundary. This means that differences in the degree to which the object differs from the background will be ignored. While differences in contrast with the background will occur, they are not considered consistent enough to be used as salient criteria. The boundary measure will be assumed to be proportional to the evidence for a boundary. Thus, the stronger the measure, the more likely the boundary at that point. Using a constant boundary value model weighs all the image boundary strength equally.

Focusing on the boundary has the advantage of reducing the computation of the match because the comparison is needed only along the boundary. This computational difference can be seen in the difference between Equation 4.2 and the one-dimensional version for boundaries:

$$C(t, b) = \int_{\mathcal{C}} t(x, y)b(x, y) ds \quad (4.3)$$

where  $\mathcal{C}$  is the path of integration. In addition, when the model is in terms of the boundary, the correspondence to the boundary of the object is simpler and more direct in that the interior of the boundary does not have to be determined. Using a boundary measure has the advantage of improving the contrast between good matches and close matches. This is because slightly mismatched areas match much better than slightly mismatched boundaries. In addition to boundary strength, boundary direction may also be used, when it is available. Using boundary direction helps to distinguish properly oriented boundary strength.

### 4.3 Maximum *a Posteriori* Match

In order to apply the prior knowledge of shape to the problem of boundary determination, we can formulate the problem using a maximum *a posteriori* (or minimum error) criterion using Bayes rule.

#### 4.3.1 Bayes Rule

Consider the problem of boundary determination as one in which the data is an image,  $b(x, y)$ , which could be depicting any one of a set of objects, where  $t_i(x, y)$  is an image template corresponding to the  $i$ th possible object. The goal is to determine which object is depicted. In order to determine this we should find the most probable one based on both the prior information and the image information.

In terms of probabilities, if we want to decide which template,  $t_i$ , an image,  $b$ , corresponds to, we need to evaluate the probability of the template given the image,

$P(t_i|b)$ , and find the maximum over  $i$ . This can be expressed using Bayes rule, where:

$$P(t_{\text{map}}|b) = \max_i P(t_i|b) = \max_i \frac{P(b|t_i)P(t_i)}{P(b)} \quad (4.4)$$

Here,  $t_{\text{map}}$  is the maximum *a posteriori* solution,  $P(t_i)$  is the prior probability of template  $t_i$  and  $P(b|t_i)$  is the conditional probability, or likelihood, of the image given the template. Since  $P(b)$ , the prior probability of the image data, will be equal for all  $i$ , it can be eliminated and it suffices to maximize:

$$P(b|t_i)P(t_i) \quad (4.5)$$

This can be simplified further, due to the monotonicity of the logarithm, by writing it as:

$$M(b, t_{\text{map}}) = \max_i M(b, t_i) = \max_i [\ln P(t_i) + \ln P(b|t_i)] \quad (4.6)$$

The function  $M$  is the general form of the objective function that will be optimized to find the maximum *a posteriori* solution. This basic form shows the tradeoff or compromise that will be made between prior information,  $P(t_i)$ , and image-derived information,  $P(b|t_i)$ . The prior probability density used will be either uniform, in which case there is no prior information, or estimated, as described in Section 3.4. For a uniform prior, this formulation reduces to the maximum likelihood solution. The likelihood,  $P(b|t_i)$ , can be derived from the image, as described below.

### 4.3.2 Likelihood Derivation

The likelihood of obtaining  $b$  given that a particular template,  $t_i$ , is present is  $P(b|t_i)$ . In this section, the likelihood will be derived and substituted into the expression for the objective function from the last section. Consider the image  $b$  to be a noise-corrupted version of one of these templates with noise that is independent and additive:  $b = t_i + n$ . This assumption is discussed in Section 4.3.4. Then,  $P(b|t_i)$  is equivalent to  $P(b = t_i + n)$  or  $P(n = b - t_i)$ . The noise at each pixel,  $n(x, y)$ , equals  $b(x, y) - t_i(x, y)$  and is governed by the probability density  $P_n$ . These events are independent for each point, so the probability for the noise over the entire area  $\mathcal{A}$  is just the product of the individual probabilities. Thus, the conditional probability of obtaining  $b$  given that it arises from  $t_i$  is the product of the noise probabilities at each pixel. That is:

$$P(b|t_i) = \prod_{\mathcal{A}} P_n(n(x, y)) \quad (4.7)$$

We make the further assumption that the noise is Gaussian with zero mean and standard deviation  $\sigma_n$ . This assumption is discussed in Section 4.3.5. This gives:

$$P(b|t_i) = \prod_{\mathcal{A}} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(b(x, y) - t_i(x, y))^2}{2\sigma_n^2}} \quad (4.8)$$

Taking the logarithm, we get:

$$\ln P(b|t_i) = \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \sum_{\mathcal{A}} \frac{(b(x, y) - t_i(x, y))^2}{2\sigma_n^2} \quad (4.9)$$

Now, by substituting this result into Equation 4.6, we can expand the match equation to get:

$$M(b, t_i) = \ln P(t_i) + \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \sum_{\mathcal{A}} \frac{(b(x, y) - t_i(x, y))^2}{2\sigma_n^2} \quad (4.10)$$

The first term is the logarithm of the prior probability. The second term is a constant. The third term is the SSD calculation of Equation 4.1. As before, because  $\sum_{\mathcal{A}} b^2(x, y)$  is a constant, this is approximately equivalent to a correlation if  $\sum_{\mathcal{A}} t_i^2(x, y)$  does not vary much. A similar interpretation for correlation as likelihood is given by Rosenfeld and Kak [76] and others.

Equation 4.10 can then be expanded to get:

$$\begin{aligned} M(b, t_i) = \ln P(t_i) + \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \\ \frac{1}{2\sigma_n^2} \sum_{\mathcal{A}} (b^2(x, y) - 2b(x, y)t_i(x, y) + t_i^2(x, y)) \end{aligned} \quad (4.11)$$

This equation is the maximum *a posteriori* function for images with the assumption of independent Gaussian noise at each pixel.

### 4.3.3 Boundary Formulation

As discussed in Section 4.2, the focus of the match will be directed to the boundary. The objective function will now be specialized to match boundaries. For this case, the object template,  $t_i(x, y)$ , represents the boundary of the object. The ideal boundary is one-dimensional, but it can be embedded into a two-dimensional image. This is done by making  $t_i(x, y)$  constant along the boundary of the object it represents and zero

everywhere else. This two-dimensional template thus represents the ideal boundary response. In order to match this template with the image, consider  $b(x, y)$  to be a boundary measure applied to the raw image data,  $b(x, y) = b(i(x, y))$ . Thus, both  $t_i$  and  $b$  are two-dimensional functions that represent boundaries. They are summed (or integrated), however, only along a contour, as will be explained below. The construction of the object boundary template and the boundary measure calculation will be discussed in Chapter 5.

Because the template has support only along the boundary, it is not necessary to sum over the entire image area for terms involving the template, but only over the curve represented by the template.

Equation 4.11 can be rewritten:

$$M(b, t_i) = \ln P(t_i) + \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \frac{1}{2\sigma_n^2} \left( \sum_{\mathcal{A}} b^2(x, y) + \sum_{\mathcal{C}_i} (-2b(x, y)t_i(x, y) + t_i^2(x, y)) \right) \quad (4.12)$$

where  $\mathcal{C}_i$  is the curve defined by the boundary in template  $t_i$ . This can be simplified because  $t_i(x, y)$  is constant over the curve that it defines:

$$M(b, t_i) = \ln P(t_i) + \sum_{\mathcal{A}} \left( \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \frac{b^2(x, y)}{2\sigma_n^2} \right) + \frac{1}{2\sigma_n^2} \sum_{\mathcal{C}_i} (2b(x, y)k - k^2) \quad (4.13)$$

where  $k$  is the magnitude of the template at any point, taken to be constant as explained in Section 4.2 and chosen to be the maximum boundary response. The function  $M$  can be simplified further by removing the terms that do not depend on the different possible templates, since we want the maximum over  $i$ . We can also remove the  $k^2$  term because

it is a function only of the length of the curve, which we can approximate as a constant because it will not vary significantly.

$$M(b, t_i) = \ln P(t_i) + \frac{1}{\sigma_n^2} \sum_{C_i} b(x, y)k \quad (4.14)$$

This equation is the maximum *a posteriori* objective for boundary templates. The first term is the bias due to the prior probability. The second term is simply a correlation of a boundary template with the boundary strength in the image, and is thus a kind of matched filter [76].

We can also consider the boundary to be a vector-valued quantity where the magnitude of the vector represents the strength of the boundary, and the direction of the vector is the direction of the tangent to the boundary. This means that  $t_i(x, y)$  has a constant magnitude along the boundary of the object it represents and a direction equal to the tangent to the boundary. The corresponding  $\mathbf{k}$  is now a function of position along the curve:

$$\mathbf{k}(x, y) = k \begin{bmatrix} \frac{\partial x(\mathbf{p}, s)}{\partial s} \\ \frac{\partial y(\mathbf{p}, s)}{\partial s} \end{bmatrix} \quad (4.15)$$

The boundary measure  $\mathbf{b}$  is a measure of both boundary magnitude and direction. Equation 4.14 can be interpreted as vector valued and rewritten using the dot product. This is equivalent to assuming that the two components of the vector boundary measure are independent and governed by the same probability density. Thus, the maximum *a posteriori*



objective for the vector-valued boundary template is:

$$M(b, t_i) = \ln P(t_i) + \frac{1}{\sigma_n^2} \sum_{c_i} \mathbf{b}(x, y) \cdot \mathbf{k}(x, y) \quad (4.16)$$

#### 4.3.4 Independence Assumption

The boundary measure at each pixel is calculated from a neighborhood of pixels and therefore the values are correlated. In addition, the original image pixel values are most likely correlated. The original pixel correlation is in general unknown, although it may be possible to devise a reasonable model using Gibbs distributions [33]. However, the correlation of the boundary measure at each position with neighboring values, caused by the boundary measure calculation, can in principle be determined explicitly. That is, the boundary measure value at each point is a known function of the pixels in the neighborhood. The correlation of the boundary measure with neighboring boundary values due to this effect can thus be determined explicitly.

The problem with including the correlation is that it excessively increases the complexity of the problem. In order to account for the correlation between boundary measure values, a covariance matrix for all of the boundary values would have to be constructed and inverted. The equation analogous to Equation 4.8 would be:

$$P(b|t_i) = \left( \frac{1}{(2\pi)^{A^2} |\mathbf{C}_{nn}|} \right)^{1/2} e^{-(1/2)\mathbf{n}^T \mathbf{C}_{nn}^{-1} \mathbf{n}} \quad (4.17)$$

where  $A$  is the number of pixels in the image,  $\mathbf{n}$  is the concatenation of values of  $b(x, y) - t_i(x, y)$  and  $\mathbf{C}_{nn}$  is the corresponding covariance matrix. To avoid this complication, we invoke an assumption of independence. Cooper [22] made the same in-

dependence assumption for a maximum likelihood approach to boundary finding. He compared the probability of error with and without the independence assumption for some simple examples and found that it was only slightly worse using the independence assumption. His results suggest that the assumption does not alter the performance significantly. The independence assumption amounts to ignoring information, not assuming additional information, by the principle of maximum entropy or least information [50].

### 4.3.5 Noise

The noise, as described by the formulation in Equation 4.8, should be thought of as not just the degradation of the signal due to the imaging process, but also the combined effect of many factors such as other objects, occlusion and boundary measurement. These factors are, in general, impossible to model explicitly. Instead, we assume it can be described by a Gaussian density with zero mean. As explained in Section 3.4, the Gaussian is the probability density with the maximum entropy among probability density functions having a given variance. A non-zero mean would only change the expression by a constant. For the variance of the noise,  $\sigma_n^2$ , we could estimate it as we did the model parameters, that is, measure it on solutions obtained either manually or with this method using a uniform prior. While the performance depends on  $\sigma_n^2$ , it is not too sensitive to it. As can be seen in Equation 4.14,  $\sigma_n^2$  represents the weight of the image information and thus the relative importance of the prior information and the likelihood. The greater the noise, the more important the prior.

## 4.4 Boundary Measures

To compare the image to the boundary model, we need a boundary measure that can be calculated from the image. A natural candidate for many images is the gray-level gradient. Objects often contrast with their backgrounds by a difference in gray level resulting in a strong gradient at the boundary. The gradient has the advantage of allowing the use of direction information. The direction of the gradient, however, is not always consistent around an object, such as when the object overlaps both lighter and darker objects. An example is shown in Figure 6.7. Often objects do not differ significantly in terms of gray level from the background, but are delineated by a border of a different gray level. Measures that respond to line strength are useful in these cases. The gray level itself (or the negative) is one measure that works as a line indicator when it is relatively high (or low) at the boundary. A more general line detector is the Laplacian. It works as a line detector by acting as a non-directional template for a line in that it has a low center and a high surround.

Any measure that indicates a change in some property that distinguishes the object from the background could be used. The proper one must be chosen with reference to the particular character of the images involved. All of these measures benefit from smoothing. Smoothing reduces the effect of noise in the image and produces a smoothly varying boundary measure. This will be beneficial in the optimization process (see Section 5.2). The implementations of these measures will be discussed in Chapter 5. Examples will be shown in Chapter 6.

## 4.5 Boundary Finding Objective Function

The templates under consideration actually form a continuum and each has a corresponding value of a parameter vector  $\mathbf{p}$  as described in Chapter 3. Thus, Equation 4.6 can be written as:

$$\max_{\mathbf{p}} M(b, \mathbf{p}) = \max_{\mathbf{p}} [\ln P(\mathbf{p}) + \ln P(b|\mathbf{p})] \quad (4.18)$$

From this and Equation 4.14 we can write the term to be maximized as:

$$M(b, \mathbf{p}) = \ln P(\mathbf{p}) + \frac{1}{\sigma_n^2} \sum_{\mathcal{C}_{\mathbf{p}}} b(x, y)k \quad (4.19)$$

where  $\mathcal{C}_{\mathbf{p}}$  is the curve defined by  $(x(\mathbf{p}), y(\mathbf{p}))$ . This is equivalent to the continuous version:

$$M(b, \mathbf{p}) = \ln P(\mathbf{p}) + \frac{k}{\sigma_n^2} \int_{\mathcal{C}_{\mathbf{p}}} b(x, y)ds \quad (4.20)$$

This line integral can be written as a regular definite integral:

$$M(b, \mathbf{p}) = \ln P(\mathbf{p}) + \frac{k}{\sigma_n^2} \int_0^S b(x(\mathbf{p}, s), y(\mathbf{p}, s))ds \quad (4.21)$$

where  $s$  is the arclength along the curve and  $S$  is the total arclength.

If a uniform prior were used, the first term would be a constant, and thus inconsequential in the maximization. We can, instead, expand the first term using the Gaussian

distribution shown in Equation 3.37.

$$\begin{aligned}
 \ln P(\mathbf{p}) &= \ln \left[ \prod_{i=1}^N \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(p_i - m_i)^2}{2\sigma_i^2}} \right] \\
 &= \sum_{i=1}^N \left[ \ln \left( \frac{1}{\sigma_i \sqrt{2\pi}} \right) - \frac{(p_i - m_i)^2}{2\sigma_i^2} \right]
 \end{aligned} \tag{4.22}$$

Now, from Equations 4.21 and 4.22 we get:

$$\begin{aligned}
 M(b, \mathbf{p}) &= \sum_{i=1}^N \left[ \ln \left( \frac{1}{\sigma_i \sqrt{2\pi}} \right) - \frac{(p_i - m_i)^2}{2\sigma_i^2} \right] + \\
 &\quad \frac{k}{\sigma_n^2} \int_0^S b(x(\mathbf{p}, s), y(\mathbf{p}, s)) ds
 \end{aligned} \tag{4.23}$$

The corresponding vector-valued version of this equation, from Equation 4.16, is:

$$\begin{aligned}
 M(b, \mathbf{p}) &= \sum_{i=1}^N \left[ \ln \left( \frac{1}{\sigma_i \sqrt{2\pi}} \right) - \frac{(p_i - m_i)^2}{2\sigma_i^2} \right] + \\
 &\quad \frac{k}{\sigma_n^2} \int_0^S \mathbf{b}(x(\mathbf{p}, s), y(\mathbf{p}, s)) \cdot d\mathbf{s}
 \end{aligned} \tag{4.24}$$

Equations 4.23 and 4.24 are the objective functions expressed in terms of the parameters that are used for boundary finding, and are discretized as will be explained in Chapter 5. The first term of this objective function is the contribution of the prior probability of the parameter vector. The influence of this term is controlled by the variance of the prior probability. The greater the variance of the prior, the smaller the influence of this term. The prior also determines the starting point for the optimization process, as described in Section 5.2. The second term is the contribution of the image information.

## Chapter 5

# Implementation

The system was implemented in Pascal on a DEC Vaxstation II using VMS. This chapter discusses the important implementation issues for the system including the objective function calculation, optimization techniques used and smoothing issues.

### 5.1 Objective Function Calculation

The objective function, Equation 4.23 or 4.24, consists of two terms. The first term, due to the prior probability, is a straightforward calculation given the current value of the parameter vector. The second term is a line integral of the boundary measure image along the boundary defined by the current parameter vector. The boundary measure calculation will be discussed in the next section. The integration is accomplished by first constructing a discrete template corresponding to the boundary as defined by the current value of the parameter vector. This template can then be directly correlated with the boundary measure image. Template generation will be discussed in Section 5.1.2.

### 5.1.1 Boundary Measure Implementations

The gray level gradient can be calculated by first smoothing with a Gaussian to reduce the effect of noise and simplify the image, followed by a finite difference approximation to the partial derivatives. This allows smoothing to be controlled independently of the differentiation.

As a smoothing filter, the Gaussian has many advantages. The two-dimensional Gaussian is the only separable, rotationally symmetric two-dimensional operator. The Gaussian has the same form as its Fourier transform and has the smallest space-frequency width product. It is the optimally localized function that is smooth in both space and frequency.

Composing the convolutions would require more computation since a larger mask would be needed to achieve the same accuracy. A good local approximation to the gradient that has  $x$  and  $y$  components that are unbiased for the same point is described by Horn [41]. This can be done by the following convolutions:

$$\frac{\partial i}{\partial x}(x + \frac{1}{2}, y + \frac{1}{2}) \approx \frac{1}{2}i(x, y) ** \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad (5.1)$$

$$\frac{\partial i}{\partial y}(x + \frac{1}{2}, y + \frac{1}{2}) \approx \frac{1}{2}i(x, y) ** \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad (5.2)$$

where  $**$  is two-dimensional convolution. This gives the average of two finite difference approximations for a point midway between pixels. The result is a gradient computation with variable smoothness. The amount of smoothing used will be discussed in Section 5.3. The magnitude of the gradient is simply the square root of the sum of the squares

of the  $x$  and  $y$  components of the gradient. The perpendicular to the gradient is used in order to compare it with the curve tangent. The gray-level Laplacian is calculated after smoothing using a similar discrete approximation:

$$\nabla^2 i(x, y) \approx \frac{1}{6} i(x, y) ** \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} \quad (5.3)$$

The Gaussian convolution was implemented using the ideas outlined by Hummel and Lowe [45]. We take advantage of the separability of the Gaussian and convolve with two one-dimensional Gaussians. Kernel values of the Gaussian are computed using block averages. Since the image is defined only on a finite grid, a rule is needed for points outside the grid. We repeat the edge values outside of the border rather than taking them equal to zero. Repeating preserves the global mean and makes the normal derivative at the border zero.

### 5.1.2 Template Generation

To facilitate the integration in Equations 4.23 and 4.24, we must generate a template corresponding to the current value of the parameter vector,  $\mathbf{p}$ . Constructing the spatially discrete template corresponding to a continuous curve is known variously as scan conversion, rasterization and discretization [30]. In addition to achieving an accurate discretization, the template construction should also try to limit the number of points in the template in order to limit the computational burden of the correlation. The overall computational burden of the construction itself must also be limited. The implementation of this procedure is crucial because the template generation is needed for every objective



function evaluation, as we will see in Section 5.2. A computationally intensive template generation would cripple the entire method. The standard approach in computer graphics is to discretize a blurred version of the curve in order to create a more visually pleasing, less jagged rendition of the line. We do not want to blur the template, however, because the boundary should be accurately located. Any blurring should more properly be done to the image. In addition, this would increase the number of points in the template.

The ideal template is the curvilinear Dirac delta function that follows the curve:  $\delta(x - x(t), y - y(t))$ . This is the most accurately localized description of the contour and consequently allows for the most accurate match. Template accuracy is particularly important in low-resolution images. The actual template can, of course, only be a discrete approximation to the delta function. The question is which discrete points are part of the template and what is their value.

To determine the form of the template, consider the problem as numerical line integration. The line integral:

$$\int_{\mathcal{C}: (x(s), y(s))} f(x, y) ds \quad (5.4)$$

can be written as the definite integral:

$$\int_{s=0}^{s=S} f(x(s), y(s)) ds \quad (5.5)$$

where  $s$  is the arclength and  $S$  is the total arclength. This is discretized to be:

$$\sum_{j=0}^{N_{\text{tem}}} f(x_j, y_j) l(x_j, y_j) \quad (5.6)$$

where  $N_{\text{tem}}$  is the number of pixels that the curve crosses,  $x_j$  and  $y_j$  are the coordinates of the  $j$ th pixel and  $l$  is the length of the portion of the curve that crosses the  $j$ th pixel. This approximation considers the function to be constant over the square area of the pixel. It is therefore effectively the rectangle rule for line integrals. The values  $l(x_j, y_j)$  are very important because if they were taken to be constant, lines at different angles would be weighted differently. This effect is not ameliorated in higher resolution images.

### 5.1.2.1 Algorithm

The template generation procedure must calculate  $x_j$ ,  $y_j$  and  $l(x_j, y_j)$ . The computation of the right-hand side of Equation 3.7 is done using a real-valued fast Fourier transform. This yields values of  $x$  and  $y$  at discrete values of  $t$ . The number of points must be large enough so that the discrete values are close enough to make linear interpolation sufficient (approximately 3 pixels). Too many points will unduly increase the computation. The fast Fourier transform is most efficient when applied to a sequence whose length is a power of 2. The number of points in the reconstructed curve is chosen to be a fixed power of 2 based on the length of the initial curve. This will then be sufficient throughout the optimization process.

The algorithm is a variation of Bresenham's [30]. The length of the line through the pixel depends on the slope,  $m$ , and the cross-over point between the two adjacent pixels,  $x_c$ , as shown in Figure 5.1 for a slope less than one and greater than zero. Other slopes are handled similarly. Bresenham's algorithm keeps track of the distance from the center of the pixel,  $t$ , and from that we can determine  $x_c$ :

$$x_c = \min(1, \max(0, \frac{2t - 1 + m}{2m})) \quad (5.7)$$

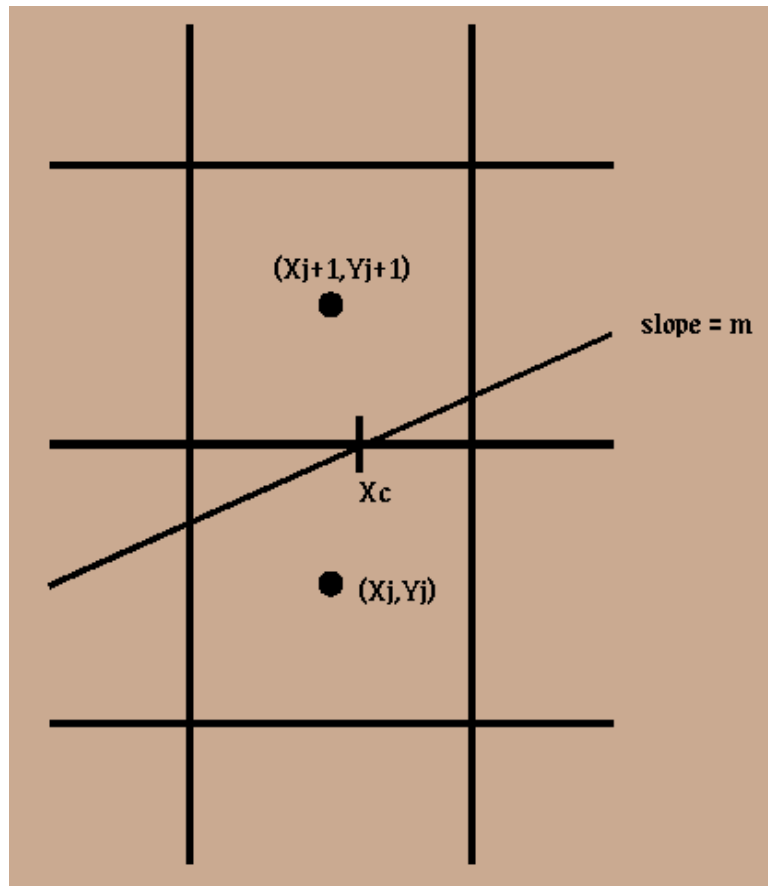


Figure 5.1: Length of curve in each pixel.

where it is bounded by zero and one in order to handle the border cases. The curve length in the two adjacent pixels is then:

$$\begin{aligned}
 l(x_j, y_j) &= x_c \sqrt{m^2 + 1} \\
 l(x_{j+1}, y_{j+1}) &= (1 - x_c) \sqrt{m^2 + 1}
 \end{aligned} \tag{5.8}$$

An example template is shown in Figure 5.2. The darkness of the square areas indicating

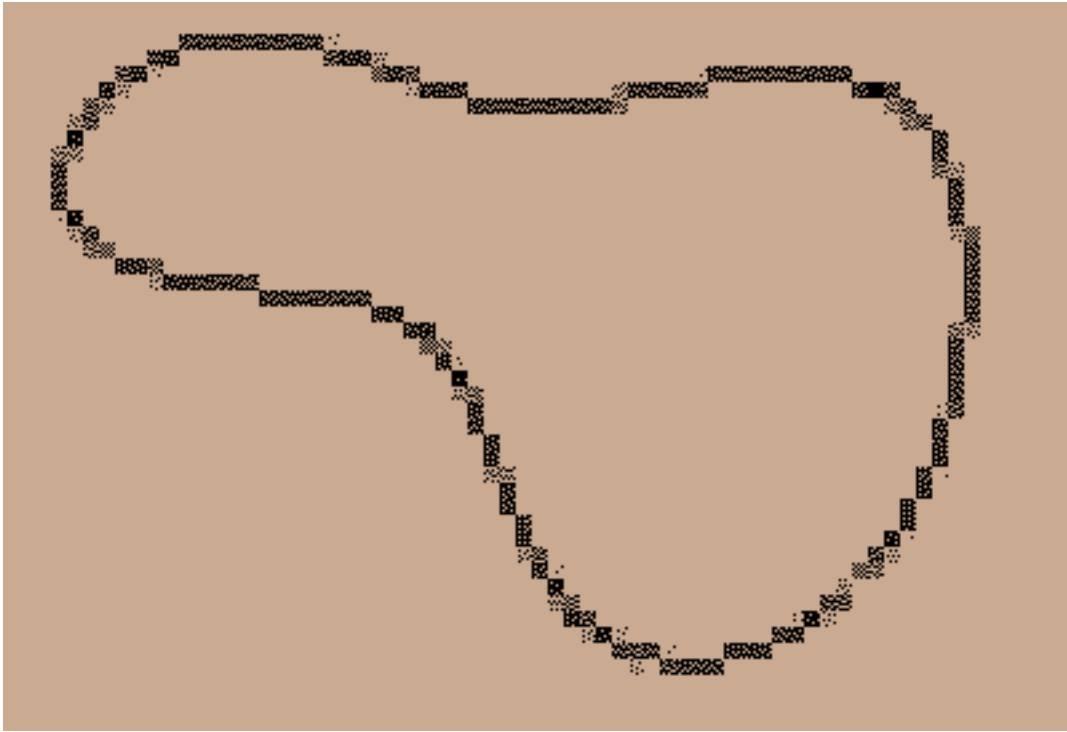


Figure 5.2: Example template.

each pixel corresponds to the length of the curve that passes over it.

The complexity of this algorithm is proportional to the length of the boundary. The boundary length is related to the size of the object and the resolution of the image. In order to speed the calculation, the template generation could be simplified by making the template binary or weighting the template simply according to whether the boundary is straight or diagonal at each pixel. The resulting inaccuracy might not degrade the performance too much, especially for higher resolution images. The template generation, and the subsequent correlation, could also be parallelized fairly straightforwardly by dividing the boundary up between processors.

## 5.2 Optimization Techniques

The problem to be solved is that of maximizing the objective function  $M(\mathbf{p})$ , as defined in Chapter 4. In order to simplify the formulation, we will consider this optimization problem to be completely unconstrained, that is, all values of  $\mathbf{p}$  are considered feasible. Any  $\mathbf{p}$  corresponding to an object outside the limits of the image, however, is clearly invalid. For this reason, we consider the value of the boundary measure to be zero outside of the image area, thus making the invalid values of  $\mathbf{p}$  nonoptimal.

A central concern of optimization is the distinction between local and global solutions. If the objective function is convex, then there is only one relative optimum, and that is the global optimum [57]. The objective function we are solving is not in general convex, but depends ultimately on the gray-level surface shape of the image. This means that there will be local maxima. To choose an optimization method we must first decide between local and global methods. The main concerns are appropriateness to the problem and feasibility as a solution. Global methods are naturally much more computationally intensive and thus can only be justified when they are absolutely essential. Apart from exhaustively searching the space, which is only practical in low-dimensional problems, other alternatives are dynamic programming [8, 63] and simulated annealing [33]. Dynamic programming requires that the objective can be formulated as a sum of local terms, as discussed in Chapter 2, with exponentially increasing complexity as the neighborhood enlarges. This is a powerful technique when the objective is of the proper form. Simulated annealing is a more general technique for finding a global optimum using a stochastic relaxation method, but unfortunately it still involves a heavy computational burden.

For this problem, the objective has been formulated in such a way as to lessen the

need for a global solution. Note that the prior probability term in the objective function, shown in Equation 4.23, is quadratic. On the tails of the distributions, this quadratic term dominates. Thus, the probability distributions make distant points in the space non-optimal because they are on the tails of the distributions. The starting point for the optimization will be taken to be the maximum of the prior distributions. The global optimum probably will be near the starting point and thus a local optimum is likely to be a global optimum. The degree to which this is true depends on the strength of the probabilistic information. The narrower the distributions, the quicker the quadratic term will dominate the rest of the expression and thus the more localized the optimal point. Since a local optimization is likely to find a global optimum, although there is still the possibility of converging to a poor local minimum, the excessive computation involved in finding a global optimum is deemed not necessary. Poor convergence can be identified by a corresponding low objective function value and verified visually. The case of wide distributions for translation and rotation caused by widely varying viewpoints could be problematic because the initial estimates of those parameters would not necessarily be very good. An additional process to determine better starting points could be incorporated using an initial exhaustive search over just those parameters or using other low-level features as a guide.

Dynamic programming is not possible here because the objective cannot be broken into local terms. The representation, and thus the overall objective function, is deliberately defined in terms of non-local shape characteristics in order to better characterize shape. Dynamic programming is only possible when the objective is limited to local properties such as continuity or low curvature.

Variational calculus methods formulate the optimization problem as a system

of partial differential equations representing the necessary conditions for an optimum [81, 41]. This results in the Euler-Lagrange equations, and for all non-trivial problems they must be solved by numerical methods. These methods typically find a local solution by iterative means [75]. Because they solve for the necessary conditions, variational methods require higher order derivatives. While this may be a disadvantage due to the greater instability of higher derivatives, this method has been widely used. In addition, since these conditions are necessary but not sufficient, unless the objective function is convex, the solution is not in general guaranteed to be an optimal point.

Instead of solving the necessary conditions, a local optimum can be found directly from the objective function. The main computational burden of optimization, for all but the simplest of objectives, is function evaluations (and, if used, gradient evaluations). Most numerical optimization methods require that the gradient of the objective function be computable. The gradient gives the direction of greatest increase in the function value [72]. It therefore provides the best direction to move in the space in order to maximize the objective function. An analytic form for the gradient must be derived in order to make use of these methods. A discrete divided difference approximation, such as the forward-difference, can of course always be computed directly from the function values.

$$\nabla f(\mathbf{p}) \approx \frac{f(\mathbf{p} + \Delta\mathbf{p}) - f(\mathbf{p})}{\Delta\mathbf{p}} \quad (5.9)$$

The discrete approximation is sensitive to errors in that the appropriate step to take,  $\Delta\mathbf{p}$ , is not obvious and the wrong choice can lead to inaccuracies. It is also computationally intensive in that it requires  $N$  function evaluations for each gradient evaluation, where  $N$  is the dimensionality of the parameter space. The options are therefore to either use

an optimization method that does not require gradient information, or to formulate a gradient for the objective function. Methods that do not require gradient information have the advantage of allowing greater flexibility in formulating the objective function and in choosing the parameter space in that they free us of the restriction of differentiability.

Two optimization methods were used for the implementation: one computed with gradient information and one without. One of the few practical ways to optimize without the use of gradient information is with direction set methods [75]. The use of a method of this type, developed by Powell, will be described in Section 5.2.1. The optimization method of continuous gradient ascent was also implemented. While the particular form of our objective function cannot be directly differentiated, it can be formulated in such a way as to enable the gradient calculation. The gradient calculation and the continuous gradient ascent method are described in Section 5.2.2.

### 5.2.1 Powell's Direction Set Method

Without gradient information, an optimization method needs some other basis for choosing which direction to move in the parameter space. By moving through the parameter space, experience can be built up to determine which direction results in the greatest increase. In general, direction set methods choose a set of directions to move in and then alter the directions based on the progress made. This is repeated until no further change in the function is made.

For each direction, a line maximization is done. That is,  $\mathbf{p}$ , the parameter vector, is updated for the direction  $\mathbf{d}$  according to:

$$\mathbf{p}' = \mathbf{p} + \lambda \mathbf{d} \tag{5.10}$$



where  $\lambda$  is found from:

$$\min_{\lambda} f(\mathbf{p} + \lambda \mathbf{d}) \tag{5.11}$$

where  $f$  is the function to be maximized. This one-dimensional maximization can be solved by standard means such as golden-section search or parabolic interpolation. The problem is that these methods require a known interval in which the maximum lies. This interval is not, in general, known. One end of the interval is  $\lambda = 0$ . The other end of the interval can be found by taking a large step in the direction that the function increases. For a unimodal function, larger and larger steps can be taken until the function decreases again. If the function is multimodal, this approach can be dangerous because a large step could fall into an adjacent local maximum. The difficulty results because there is no basis for choosing the size of the step taken in order to guarantee remaining in the neighborhood of one local maximum. Cautious small steps will result in many function evaluations. This disadvantage is true of all maximization methods that rely on line maximization.

Each iteration of a direction set method is thus a set of line maximizations. In Powell's method [74], the direction set is initialized to be the individual parameter directions. While this set is sufficient for convergence, it can be inefficient. Powell's method attempts to improve upon this set by replacing the direction of largest increase in the set with the average direction moved after maximizing in all directions. The rationale is that the average direction moved is likely to be a good direction, but the direction of largest increase is likely to point in a similar direction. This update to the directions therefore adds a likely good new direction while avoiding the possibility that all of the directions

will point in nearly the same direction. The direction set must continue to span the entire  $N$ -dimensional space or the optimum point found will only be valid in a subspace.

The use of this method is fairly straightforward; all that is needed is function evaluations. For this method we use the relative parameters,  $\mathbf{p}_{\text{rel}}$  from Equation 3.25, because we are not concerned with the complexity of the gradient. In spite of the difficulties of line maximizations, this method works well, but the number of function evaluations is large.

Figure 5.3 shows the function value for each function evaluation taken using Powell's method for a typical case (shown in Figure 6.3, noise standard deviation of 0.2). Note the large number of iterations required for convergence. Also, the plot is quite jagged. This is partially due to searches in directions where there is no possible improvement. It is also a general property of the line maximization process which can take large steps beyond the range of improvement in a particular direction. The function value for each full iteration of Powell's method, however, converges steadily. Results using Powell's method will be shown in Chapter 6. This optimization method took an average of 30 minutes CPU time on a Vaxstation II for the range of problems shown in Chapter 6.

### 5.2.2 Continuous Gradient Ascent

The main problem with Powell's method is its slow speed of convergence. This is due to both the lack of gradient information and the line maximizations. Gradient methods are very powerful in that exploratory moves of the sort that Powell's method makes are made unnecessary because the gradient provides that information. Most gradient methods use line maximization. The simplest method of this sort is steepest ascent, which at each iteration performs a line maximization in the direction of the gradient. This method is inefficient because each step is perpendicular to the previous one and

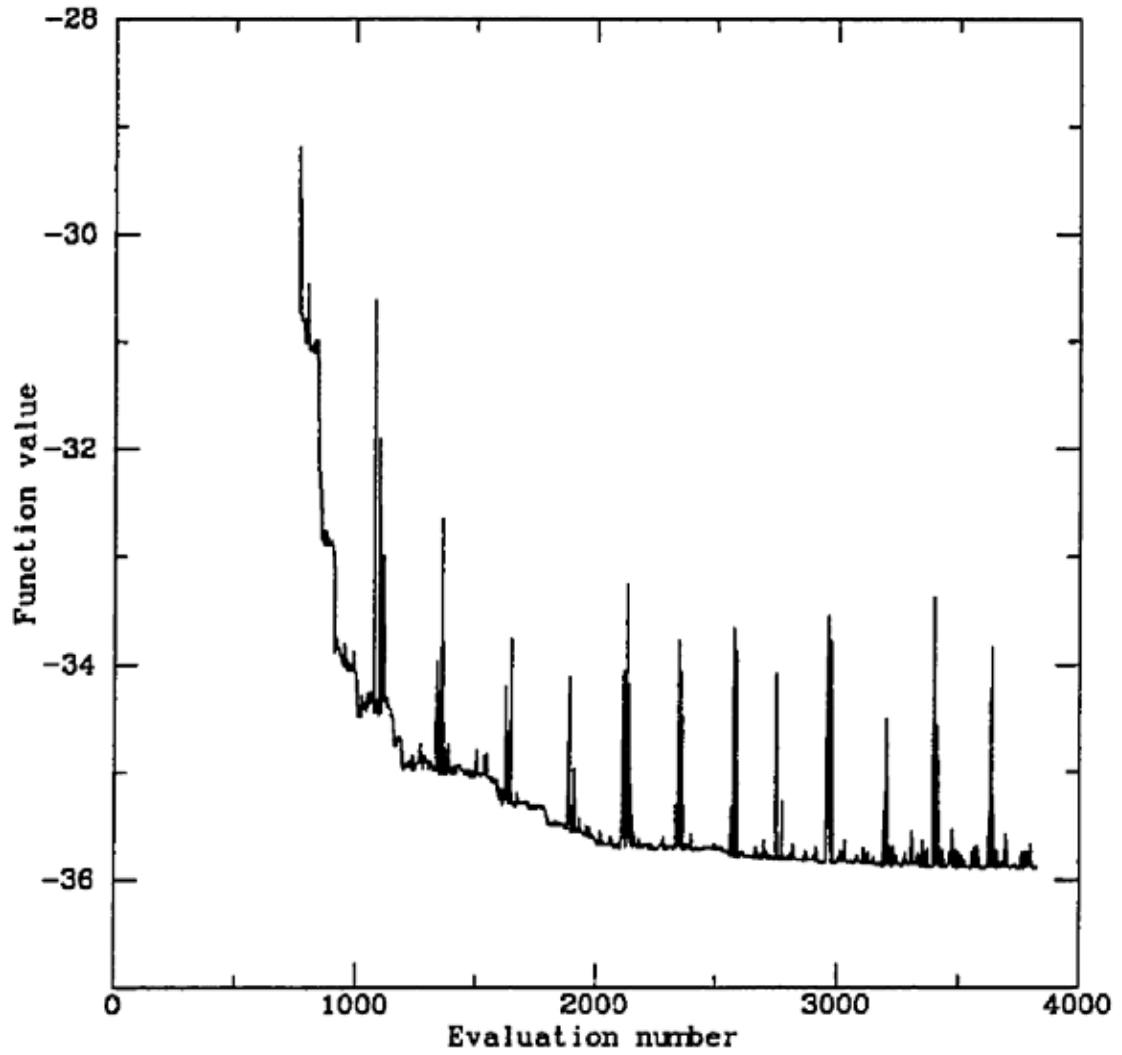


Figure 5.3: Powell's method convergence example.

this results in a zigzagging path to the optimal point. Other methods, such as conjugate gradient methods, attempt to overcome this problem. The rationale for line maximization is to make the most out of each gradient computation because of the computational cost. The problem is that the gradient direction is only the steepest direction, and therefore the best, in the small neighborhood about the point of evaluation where the gradient is effectively constant. The cost due to the many function evaluations required for line maximization can outweigh the cost of additional gradient evaluations. Also, there is still the problem of obtaining a local optimum using line maximization with a multimodal objective.

An alternative approach which avoids the use of line maximizations is continuous gradient ascent [72, 46, 27]. This method takes small steps in the direction of the gradient. The truly continuous version of this method, using infinitesimal steps, is only appropriate for an analog computer. A discrete approximation is therefore used here. The goal of the method is to move always in the direction of greatest increase because this will save many function evaluations. The gradient must be relatively easy to compute, however, in order to make this method practical. If the gradient computation is comparable to the function computation, for example, the overall computation could be greatly reduced.

#### 5.2.2.1 Ascent Method

The ascent method follows the gradient, taking small steps until the optimal point is reached. The initial vector  $\mathbf{p}_0$  is updated by the following:

$$\mathbf{p}_{j+1} = \mathbf{p}_j + \omega \nabla M(\mathbf{p}_j) \tag{5.12}$$

Function evaluations are needed at each step to check for convergence. The gradient weighting factor,  $\omega$ , must be chosen small so as not to overstep the maximum. If  $\omega$  is too small, the number of iterations will be excessive. Thus,  $\omega$  starts at a predetermined small value. If  $\mathbf{p}_{j+1} < \mathbf{p}_j$ , then  $\omega$  is decreased by a set factor and  $\mathbf{p}_{j+1}$  is recalculated. This avoids overstepping due to too large a value of  $\omega$ . The algorithm stops when  $M(\mathbf{p}_{j+1}) - M(\mathbf{p}_j) = 0$  or  $\omega = 0$  (within machine precision).

In order to show that this algorithm converges, we can apply the Global Convergence Theorem [57]. The objective function is an ascent function for the algorithm as long as  $\omega$  is small enough. For the sake of this analysis, we will consider  $\omega$  to be fixed to a small constant. The technical conditions of closure of the mapping and compactness of the generated points are satisfied due to continuity of the algorithm. Thus the algorithm converges to a solution point.

In order to determine the rate of convergence, we base our derivation on a similar derivation for the rate of convergence for steepest descent described by Luenberger [57]. Consider the following quadratic objective:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{x}^T\mathbf{b} \quad (5.13)$$

where  $\mathbf{Q}$  is a negative definite symmetric  $n \times n$  matrix. If we define  $\mathbf{y}_k = \mathbf{x}_k - \mathbf{x}^*$  where  $\mathbf{x}^*$  is the maximum point of  $f$  we can consider the equivalent problem (differing only by a constant) of maximizing:

$$E(\mathbf{x}) = \frac{1}{2}\mathbf{y}^T\mathbf{Q}\mathbf{y} \quad (5.14)$$

Consider the ratio of  $E$  at successive iterations of the algorithm (the convergence ratio):

$$\begin{aligned}
\frac{E(\mathbf{x}_{j+1})}{E(\mathbf{x}_j)} &= \frac{\frac{1}{2}\mathbf{y}_{j+1}^T \mathbf{Q} \mathbf{y}_{j+1}}{\frac{1}{2}\mathbf{y}_j^T \mathbf{Q} \mathbf{y}_j} \\
&= \frac{\frac{1}{2}(\mathbf{y}_j + \omega \mathbf{g}_j)^T \mathbf{Q} (\mathbf{y}_j + \omega \mathbf{g}_j)}{\frac{1}{2}\mathbf{y}_j^T \mathbf{Q} \mathbf{y}_j} \\
&= 1 + \frac{2\omega \mathbf{g}_j^T \mathbf{Q} \mathbf{y}_j + \omega^2 \mathbf{g}_j^T \mathbf{Q} \mathbf{g}_j}{\mathbf{y}_j^T \mathbf{Q} \mathbf{y}_j}
\end{aligned} \tag{5.15}$$

where  $\mathbf{g}_j = \nabla f(\mathbf{x}_j)$ . Since  $\mathbf{g}_k = \mathbf{Q} \mathbf{y}_k$ , we get:

$$\frac{E(\mathbf{x}_{j+1})}{E(\mathbf{x}_j)} = 1 + \frac{2\omega \mathbf{g}_j^T \mathbf{g}_j + \omega^2 \mathbf{g}_j^T \mathbf{Q} \mathbf{g}_j}{\mathbf{g}_j^T \mathbf{Q}^{-1} \mathbf{g}_j} \tag{5.16}$$

With the appropriate rotation,  $\mathbf{Q}$  becomes diagonal with diagonal elements equal to its eigenvalues (all negative) denoted in decreasing order:  $(\lambda_1, \lambda_2, \dots, \lambda_N)$ . Using this representation we have:

$$\frac{E(\mathbf{x}_{j+1})}{E(\mathbf{x}_j)} = 1 + \frac{2\omega \sum_{i=1}^N g_{ji}^2 + \omega^2 \sum_{i=1}^N \lambda_i g_{ji}^2}{\sum_{i=1}^N g_{ji}^2 / \lambda_i} \tag{5.17}$$

This expression can be bounded from above by:

$$\frac{E(\mathbf{x}_{j+1})}{E(\mathbf{x}_j)} \leq 1 + 2\omega \lambda_1 + \omega^2 \lambda_1 \lambda_N \tag{5.18}$$

Thus, if  $0 < \omega < -2/\lambda_N$ , the method converges linearly with a convergence ratio no greater than  $1 + 2\omega \lambda_1 + \omega^2 \lambda_1 \lambda_N$ . The best value for  $\omega$  is  $-1/\lambda_N$  for which the convergence ratio is  $1 - \lambda_1/\lambda_N$ . This method of adjusting  $\omega$  can be seen as a way to keep it within the bounds required for convergence. If  $\omega$  is too large, it will overshoot and  $\omega$  will be

adjusted. Note that while the corresponding bound for steepest descent is  $\left(\frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1}\right)^2$ , this is for an entire line maximization which requires many function evaluations. If we consider just four iterations of continuous gradient ascent at a time and the best ratio, the convergence ratio is better than that of steepest descent. That is:

$$\left(1 - \frac{\lambda_1}{\lambda_N}\right)^4 < \left(\frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1}\right)^2 \quad (5.19)$$

A typical line maximization will use many more than four function evaluations. Continuous gradient descent is therefore comparable to, and potentially much better than, steepest descent. The above argument extends to the case of a general objective function where  $\mathbf{Q}$  is identified with the Hessian of the objective at the solution point [57].

To use continuous gradient ascent, both function evaluations and gradient evaluations are needed. The gradient calculation will be described in Section 5.2.2.2. As explained in that section, we will use only  $\mathbf{p}_{\text{raw}}$  and  $\mathbf{p}_{\text{ref}}$  and not  $\mathbf{p}_{\text{rel}}$  (from Equation 3.25) because of the complexity of the gradient with respect to the relative parameters. The principal advantage of this method is the improvement in speed over Powell's method. There is some indication that the refined parameter vector,  $\mathbf{p}_{\text{ref}}$ , is a better space to work in because it isolates the more distinct geometric features of rotation and scale. Thus, some types of delineation problems may work better using  $\mathbf{p}_{\text{ref}}$ , perhaps justifying the increase in computation.

Figure 5.4 shows the function value for each function evaluation taken using continuous gradient ascent for a typical case (shown in Figure 6.11). Note the decrease in the number of iterations compared to Powell's method. The jaggedness in this plot is due to  $\omega$  adjustments, most of which are at the very end. Results using continuous gra-

dient ascent will be shown in Chapter 6. This optimization method took an average of 2 minutes CPU time on a Vaxstation II for the range of problems shown in Chapter 6.

### 5.2.2.2 Gradient Formulation

The specific form of the objective function developed in Section 4.5 makes the gradient calculation difficult. Numerical integration will have to be used because the integrand is a complicated function of the data which cannot be analytically integrated. We cannot treat it as a double sum or integral, as in Equation 4.10, because we do not have a closed-form expression for the two-dimensional template. The line integral formulation of Equation 4.20 is more tractable, but it must be converted to a definite integral in order to be differentiated. The curve of integration is defined by  $(x(\mathbf{p}, t), y(\mathbf{p}, t))$  where  $t$  is not equal to arclength along the curve, as discussed in Section 3.3.1. The definite integral form of a line integral uses the differential element of arclength,  $ds$ , which can be expressed as:

$$ds = \sqrt{\left(\frac{\partial x(\mathbf{p}, t)}{\partial t}\right)^2 + \left(\frac{\partial y(\mathbf{p}, t)}{\partial t}\right)^2} dt \quad (5.20)$$

This expression is impractical because of its complexity, since both  $x$  and  $y$  involve several terms.

A practical gradient calculation may be obtained by using a numerical approximation based directly on the objective function calculation. The template for the objective function calculation is computed using interpolated values. The definite integral formulation of Equation 4.21 can be used by basing it on these interpolated values.



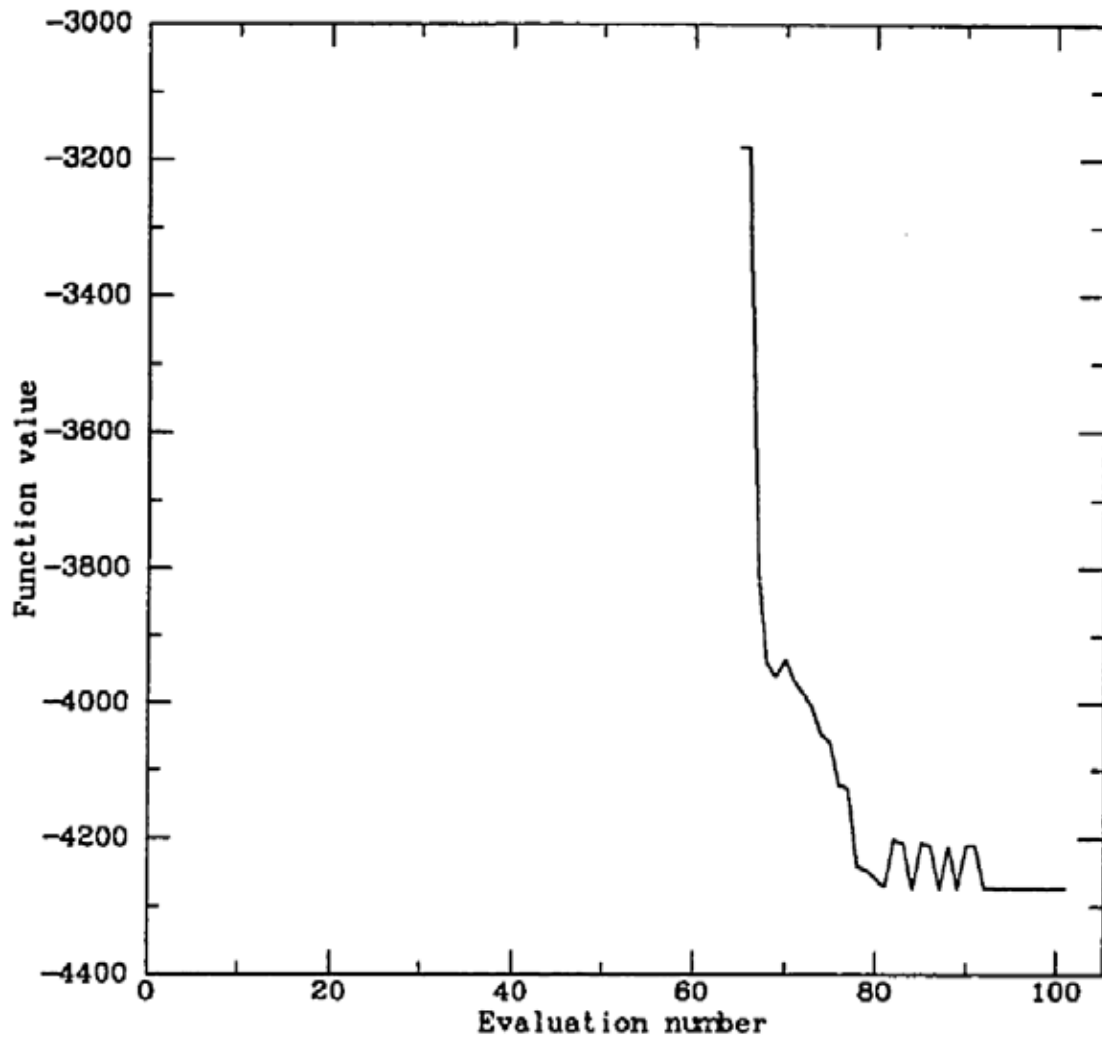


Figure 5.4: Continuous gradient ascent convergence example.

First, differentiate Equation 4.21 to get:

$$\nabla M(\mathbf{p}) = \frac{\partial}{\partial \mathbf{p}} \left( \ln p(\mathbf{p}) - \frac{k}{2\sigma^2} \int_0^S b(x(\mathbf{p}, s), y(\mathbf{p}, s)) ds \right) \quad (5.21)$$

$$\begin{aligned} &= \frac{\partial \ln p(\mathbf{p})}{\partial \mathbf{p}} \\ &\quad - \frac{k}{2\sigma^2} \int_0^S \frac{\partial b(x(\mathbf{p}, s), y(\mathbf{p}, s))}{\partial x} \frac{\partial x}{\partial \mathbf{p}} + \frac{\partial b(x(\mathbf{p}, s), y(\mathbf{p}, s))}{\partial y} \frac{\partial y}{\partial \mathbf{p}} ds \end{aligned} \quad (5.22)$$

In the above equation, we have used the simplifying approximation that  $ds$  is not a function of  $\mathbf{p}$ . This results in a much simpler expression for the gradient because we have avoided differentiating Equation 5.20. An inaccurate gradient could potentially slow the optimization somewhat because movement would not be exactly in the direction of greatest increase. To evaluate the above expression we must be able to calculate all of the subexpressions and then numerically integrate. The first term can be calculated if the distributions are chosen as closed form expressions. In particular, if they are taken to be Gaussian, as in Equation 4.22, this gives:

$$P(\mathbf{p}) = \prod_{i=1}^n P(p_i) = \prod_{i=1}^n \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(p_i - m_i)^2}{2\sigma_i^2}} \quad (5.23)$$

$$\ln P(\mathbf{p}) = \sum_{i=1}^n \left[ \ln \left( \frac{1}{\sigma_i \sqrt{2\pi}} \right) - \frac{(p_i - m_i)^2}{2\sigma_i^2} \right] \quad (5.24)$$

$$\frac{\partial \ln P(\mathbf{p})}{\partial \mathbf{p}} = -\frac{p_i - m_i}{2\sigma_i^2} \quad (5.25)$$

The partials of  $b$ , the boundary measure image, with respect to  $x$  and  $y$ , can be calculated using a discrete divided difference approximation. The central-difference is used here because it is symmetric but still localized. The partials with respect to  $\mathbf{p}$  are more problematic in that we do not have analytic expressions for  $x$  or  $y$  as functions of  $s$ .

We do have expressions for  $x$  and  $y$  as functions of  $t$  in Equation 3.7, which can be differentiated with respect to  $\mathbf{p}$ . We can then use the following:

$$\frac{\partial x(\mathbf{p}, s)}{\partial \mathbf{p}} = \frac{\partial x(\mathbf{p}, t(s))}{\partial \mathbf{p}} \quad (5.26)$$

Since we know  $s$  at all points along the curve simply from partial sums of the arclength obtained during the template generation process, and we know  $t$  at discrete points along the curve from the inverse Fourier transform, we can interpolate to determine  $t(s)$  at all points. That is:

$$t(s) = t_i + \frac{(t_{i+1} - t_i)(s - s_i)}{s_{i+1} - s_i} \quad (5.27)$$

We can derive the partials of  $x$  and  $y$  with respect to the raw parameters,  $\mathbf{p}_{\text{raw}}$ , from the Fourier relation in Equation 3.7:

$$\begin{aligned} \frac{\partial x(t)}{\partial a_0} &= 1 & \frac{\partial x(t)}{\partial a_k} &= \cos kt & \frac{\partial x(t)}{\partial b_k} &= \sin kt \\ \frac{\partial x(t)}{\partial c_0} &= 0 & \frac{\partial x(t)}{\partial c_k} &= 0 & \frac{\partial x(t)}{\partial d_k} &= 0 \end{aligned} \quad (5.28)$$

$$\begin{aligned} \frac{\partial y(t)}{\partial a_0} &= 0 & \frac{\partial y(t)}{\partial a_k} &= 0 & \frac{\partial y(t)}{\partial b_k} &= 0 \\ \frac{\partial y(t)}{\partial c_0} &= 1 & \frac{\partial y(t)}{\partial c_k} &= \cos kt & \frac{\partial y(t)}{\partial d_k} &= \sin kt \end{aligned} \quad (5.29)$$

resulting in extremely simple expressions. If instead we use the refined ellipse parameters,  $\mathbf{p}_{\text{ref}}$  (excluding the relative transformation), we start with the corresponding Fourier

relation, using Equation 3.14:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} A_k \cos \theta_k \cos(\phi_k + kt) - B_k \sin \theta_k \sin(\phi_k + kt) \\ A_k \sin \theta_k \cos(\phi_k + kt) + B_k \cos \theta_k \sin(\phi_k + kt) \end{bmatrix} \quad (5.30)$$

and differentiate to get the correspondingly more complex expressions for the partial derivatives:

$$\begin{aligned} \frac{\partial x(t)}{\partial a_0} &= 1 \\ \frac{\partial x(t)}{\partial c_0} &= 0 \\ \frac{\partial x(t)}{\partial A_k} &= \cos \theta_k \cos(\phi_k + kt) \\ \frac{\partial x(t)}{\partial B_k} &= -\sin \theta_k \sin(\phi_k + kt) \\ \frac{\partial x(t)}{\partial \theta_k} &= -A_k \sin \theta_k \cos(\phi_k + kt) - B_k \cos \theta_k \sin(\phi_k + kt) \\ \frac{\partial x(t)}{\partial \phi_k} &= -A_k \cos \theta_k \sin(\phi_k + kt) - B_k \sin \theta_k \cos(\phi_k + kt) \end{aligned} \quad (5.31)$$

$$\begin{aligned} \frac{\partial y(t)}{\partial a_0} &= 0 \\ \frac{\partial y(t)}{\partial c_0} &= 1 \\ \frac{\partial y(t)}{\partial A_k} &= \sin \theta_k \cos(\phi_k + kt) \\ \frac{\partial y(t)}{\partial B_k} &= \cos \theta_k \sin(\phi_k + kt) \\ \frac{\partial y(t)}{\partial \theta_k} &= A_k \cos \theta_k \cos(\phi_k + kt) - B_k \sin \theta_k \sin(\phi_k + kt) \\ \frac{\partial y(t)}{\partial \phi_k} &= -A_k \sin \theta_k \sin(\phi_k + kt) + B_k \cos \theta_k \cos(\phi_k + kt) \end{aligned} \quad (5.32)$$

Differentiation with respect to the relative parameters,  $\mathbf{p}_{\text{rel}}$ , will result in a much more complex expression because the interdependencies between the parameters greatly increase.

The vector valued version of the objective function can be differentiated similarly to the scalar one. First, from Equations 4.21 and 4.15 we get:

$$M(\mathbf{p}) = \ln p(\mathbf{p}) - \frac{k}{2\sigma^2} \int_0^S \left[ b_x(x(\mathbf{p}, s), y(\mathbf{p}, s)) \frac{\partial x(\mathbf{p}, s)}{\partial s} \right] + \quad (5.33)$$

$$\left[ b_y(x(\mathbf{p}, s), y(\mathbf{p}, s)) \frac{\partial y(\mathbf{p}, s)}{\partial s} \right] ds$$

where  $b_x$  and  $b_y$  are the  $x$  and  $y$  components of  $b$ . To simplify, we can drop the explicit notation of the dependence of  $x$  and  $y$  on  $\mathbf{p}$  and  $s$ . From this we can differentiate to get:

$$\begin{aligned} \nabla M(\mathbf{p}) &= \frac{\partial \ln p(\mathbf{p})}{\partial \mathbf{p}} - \frac{k}{2\sigma^2} \int_0^S \frac{\partial}{\partial x} \left[ b_x(x, y) \frac{\partial x}{\partial s} + b_y(x, y) \frac{\partial y}{\partial s} \right] \frac{\partial x}{\partial \mathbf{p}} + \\ &\quad \frac{\partial}{\partial y} \left[ b_x(x, y) \frac{\partial x}{\partial s} + b_y(x, y) \frac{\partial y}{\partial s} \right] \frac{\partial y}{\partial \mathbf{p}} ds \\ &= \frac{\partial \ln p(\mathbf{p})}{\partial \mathbf{p}} - \frac{k}{2\sigma^2} \int_0^S \left[ \frac{\partial x}{\partial s} \frac{\partial b_x(x, y)}{\partial x} + \frac{\partial y}{\partial s} \frac{\partial b_y(x, y)}{\partial x} \right] \frac{\partial x}{\partial \mathbf{p}} + \quad (5.34) \\ &\quad \left[ \frac{\partial x}{\partial s} \frac{\partial b_x(x, y)}{\partial y} + \frac{\partial y}{\partial s} \frac{\partial b_y(x, y)}{\partial y} \right] \frac{\partial y}{\partial \mathbf{p}} ds \end{aligned}$$

This expression requires the calculation of  $\partial x/\partial s$  and  $\partial y/\partial s$ . These, along with the partials of  $b_x$  and  $b_y$  with respect to  $x$  and  $y$ , can be calculated using a discrete divided difference approximation. The other terms are calculated as above.

To numerically integrate, we use the  $x$   $y$  points generated by the template making procedure along with the associated lengths and  $t(s)$  values. The lengths are the weight for each term in the integration. The template is relatively computationally expensive to generate but it only has to be computed once for each gradient calculation. In fact, the function value and the gradient can be calculated from one template generation. Thus, overall the gradient computation is comparable to objective computation and practical to use.

### 5.3 Smoothing

The amount of smoothing necessary for an image depends on the amount of noise. The distribution of the noise is not usually known. A certain amount of smoothing is necessary to lessen the effect of noise. Further smoothing simplifies the image. Multiresolution or scale-space methods smooth by Gaussians over a range of widths (scales) and avoid choosing one level of smoothing [91]. These methods take advantage of the simplification and organization that the change in scale provides.

For this work, the simplification in the image due to smoothing results in a corresponding simplification in the search space. A smoother and simpler structure in the search space makes the optimum easier to find because the local region of unimodality grows larger. This simplification also affords the possibility of decreasing the dimensionality of the space because the high frequency content of the boundary is blurred out, thus making the boundary describable in terms of fewer harmonics. An example of such an approach is shown in Figure 5.5. The results from each level are passed to the next, where the model is increased by one harmonic and the level of smoothing is reduced.

The shortcomings of this approach are revealed when it is applied to images with more structure due to other objects. Boundaries from a highly smoothed image will not correspond to region boundaries in the unsmoothed image because features are merged or destroyed as the smoothing increases. Further processing can be used to determine which boundaries are salient [13]. Too much smoothing on complex scenes is a mistake once the smoothing goes beyond simplifying individual objects. In general, the full multiresolution approach can be applied only to images with well separated features that will not merge when highly smoothed.

In order to avoid the problems of too much smoothing while still simplifying the

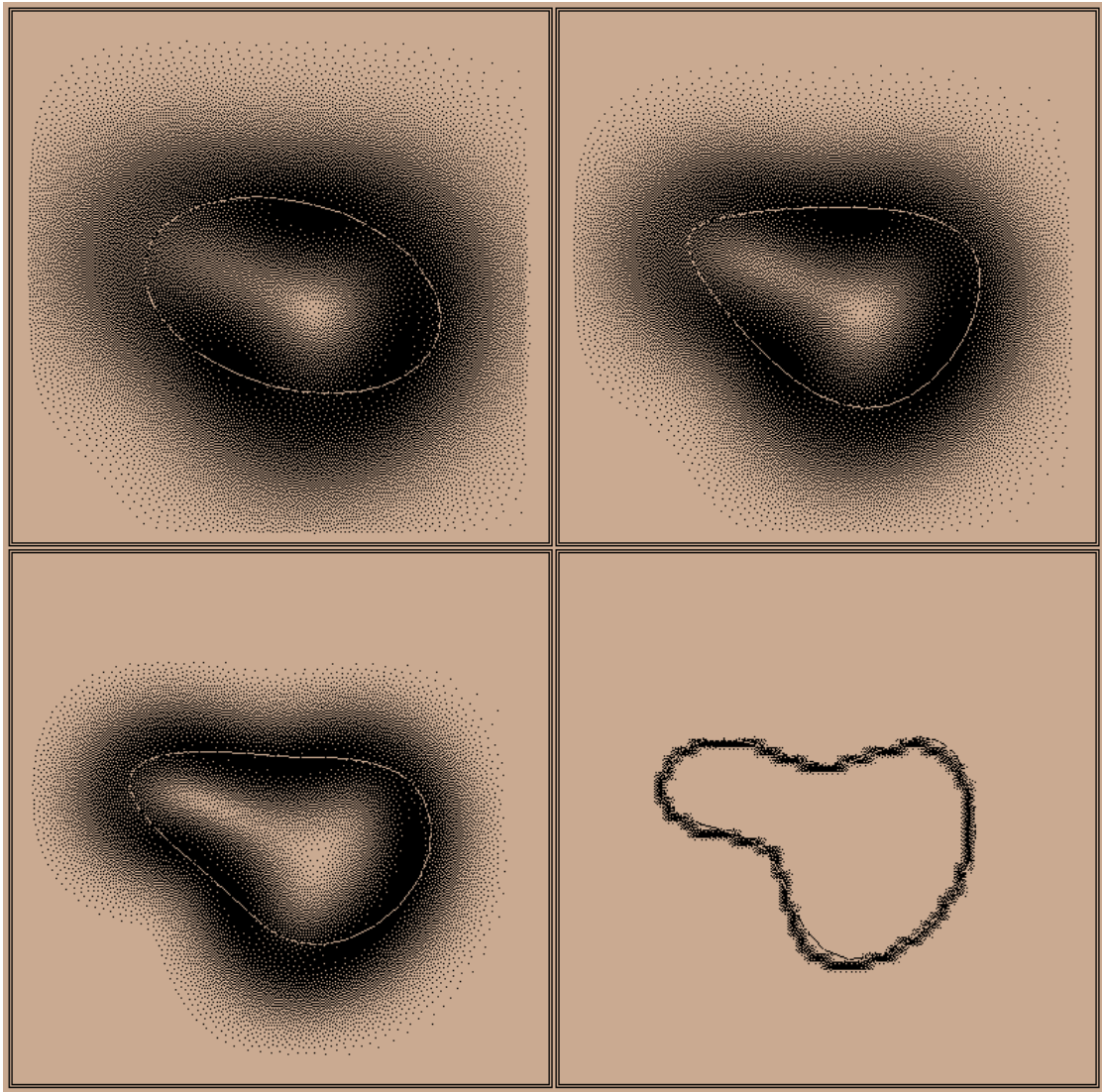


Figure 5.5: Multiresolution example. Four successive stages of a multiresolution strategy applied to a single-object synthetic image. The smoothed gradient of the synthetic image is shown with the final contour at each level. The model increases in complexity from one to four harmonics as the smoothing decreases from  $\sigma = 9.0$  to  $\sigma = 1.0$ .

image somewhat, we have adopted a simple two-level approach. Since the merging of boundaries occurs at high levels of smoothing, the optimization is first done using an intermediate amount of smoothing to calculate the boundary measure. This allows the optimization to approach the appropriate solution with less of a chance of getting stuck in a local maximum. The result of that optimization is then used as the initial estimate for a second optimization, done on a slightly smoothed version, in order to accurately localize the boundary.



## Chapter 6

# Experiments

The system must be evaluated by testing it under a variety of conditions. In particular, we can investigate the dependence of its accuracy on prior information and image quality. The two-dimensional boundary finding system was tested on both real and synthetic images. The advantage of synthetic images is that the true solution is known and thus can be used for quantitative comparison. The results on real images can be evaluated qualitatively.

### 6.1 Evaluation

To evaluate a boundary found by this method we need a good measure of its error from the true boundary. For synthetic images, the true boundary is known and thus a quantitative measure can be devised. For real images, we must rely on qualitative evaluation. While a better way of evaluating such a system might be in terms of some specific measure of interest, such as area or perimeter, that results from further processing, we prefer to evaluate the boundary finding step in isolation. A natural quantitative measure of error

is one that is equal to zero for a perfect match and gets larger as the boundaries become further apart in terms of euclidean distance. Quantitative evaluation is a relatively rare and often neglected component of computer vision. Edge detector performance, a somewhat different problem, has been addressed by Abdou and Pratt [1]. The measure they developed, also used by Gerbrands [34], that is most relevant to this work was based, in part, on the distance between the detected edge and the actual edge. Their measure was limited, however, to straight edges. Cooper [23, 22] used a similar measure which was limited to perturbations of straight lines. All of these approaches simplify the problem by reducing it to one dimension. It should be clear, especially for closed boundaries, that the problem is inherently two-dimensional and should be kept that way. It might be considered reasonable to use some measure of distance between the Fourier descriptors themselves as a measure of error. This, however, begs the question in that this system is attempting, in part, to justify Fourier descriptors as a reasonable representation for shape.

In order to evaluate the error, while still retaining the intrinsic two-dimensionality of the problem, we need to establish a correspondence between the points on the two curves. The appropriate error measure is then the average distance between the corresponding points on the two curves. The correspondence can be determined by finding the offset between the curves that produces the minimum average error. The error,  $\epsilon$ , between the two curves is then defined by:

$$\epsilon(v, v') = \min_{0 \leq t_0 \leq 2\pi} \frac{1}{2\pi} \int_0^{2\pi} \|v(t) - v'(t + t_0)\| dt \quad (6.1)$$

where  $v$  is the true curve,  $v'$  is the measured boundary and  $t_0$  is the offset. In discrete

form, the curves are first made commensurable by resampling them to an equal number of equally spaced points, using simple linear interpolation. The discrete form of the error calculation can be written as:

$$\epsilon(v, v') = \min_{0 \leq t_0 \leq n} \frac{1}{n} \sum_{t=0}^n \|v(t) - v'(t + t_0)\| \quad (6.2)$$

The average error has to be evaluated at all possible offsets in order to determine the minimum. This technique is general and readily extendable to higher dimensions with the corresponding increase in computation. The computational burden is not important, however, since this is only done for evaluation purposes and not in actual use.

## 6.2 Synthetic

The image shown in Figure 6.1 is a simple synthetic Mondrian image where the target object (the darkest) is partially occluded by one object on the right and overlaps another. The extraneous objects are potential sources of choice or confusion because alternate boundaries are plausible, especially at the right side of the target and at the center of the top boundary. The probability distributions for the parameter vector  $\mathbf{p}$  were derived from a set of manually traced contours. The initial curve superimposed is defined by the mean parameter vector. The starting boundary only roughly agrees with the target in terms of shape and location. The final curve accurately delineates the target with approximately one half pixel average error. The curve correctly avoids both the overlapped and the occluding object both because of the limit on the number of harmonics (four) and the bias due to the probability distributions on the parameters.

Another synthetic image is shown in Figure 6.2. The target is the s-shaped portion

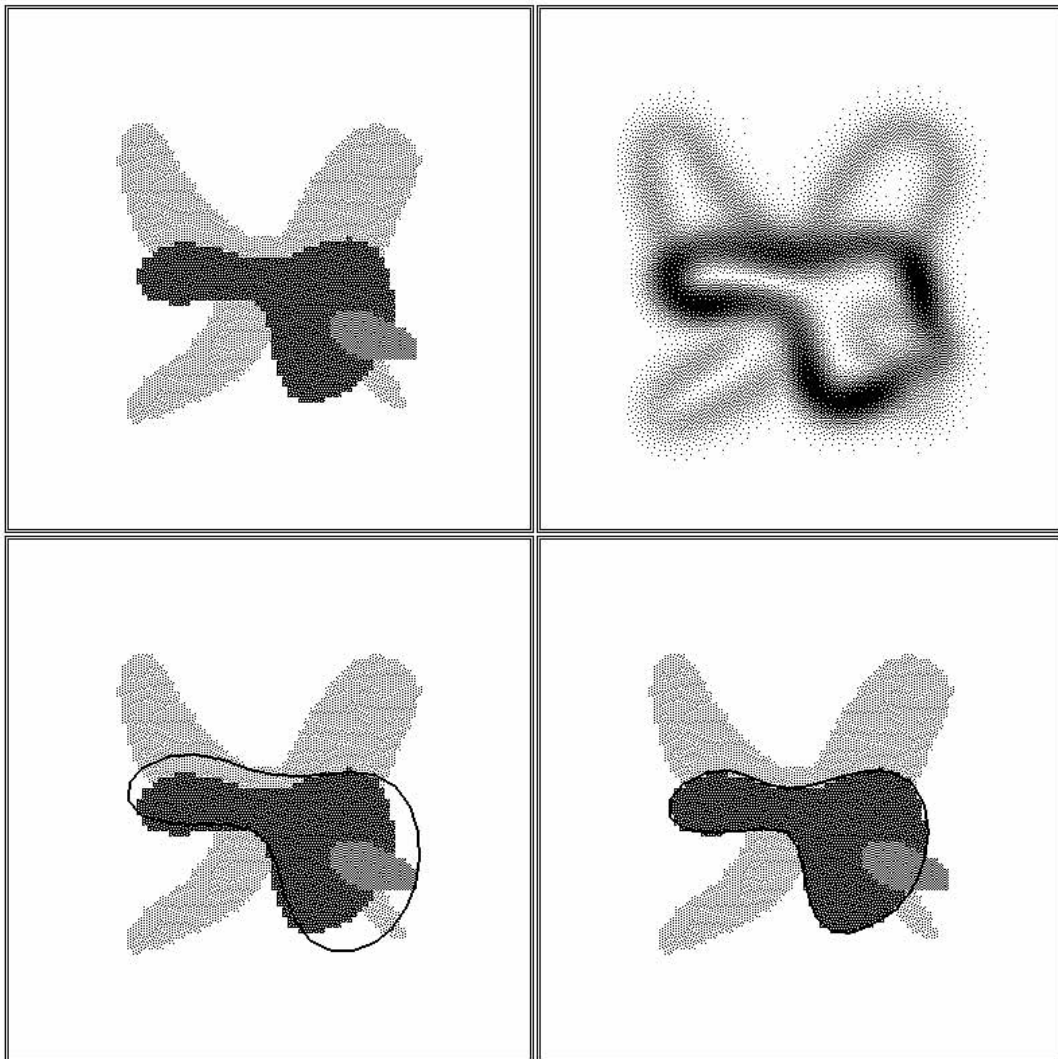


Figure 6.1: Synthetic image example. Top left: Synthetic image ( $96 \times 96$ ). Top right: Gray level gradient magnitude ( $\sigma = 3.0$ ). Bottom left: Initial contour (4 harmonics). Bottom right: Final contour on target shape.

of the boundary of the large object. This structure is described by an open curve model. The other objects in the image are potential sources of confusion, as in the previous example. Note the large number of harmonics needed due to the relative complexity of the curve and because each harmonic is described by just two parameters as opposed to four for closed curves. The final curve accurately describes the target and avoids the other objects.

### 6.2.1 Varying Noise

The effect of noise on the performance of the system can be investigated by adding different amounts of noise to a synthetic image and comparing the resulting boundaries to the true boundary. The synthetic image shown in Figure 6.1 was altered by the addition of Gaussian noise of zero mean and varying standard deviation. Examples of the noisy images are shown in Figure 6.3. The accuracy of the results of running the optimization on the noisy images, using the same distributions for each, is shown in Figure 6.4. Signal-to-noise ratio (SNR) is defined here as:

$$\text{SNR} = \frac{\text{Gray level contrast between object and background}}{\text{Standard deviation of Gaussian noise}} \quad (6.3)$$

The error shown is from the boundary from which the image was constructed. The accuracy of the resulting contours is good ( $\sim 0.5$  pixels) for  $\text{SNR} > 1$  and steadily worsens as the noise increases. Most of the error seems to be due to confusion with the overlapping object on the right.

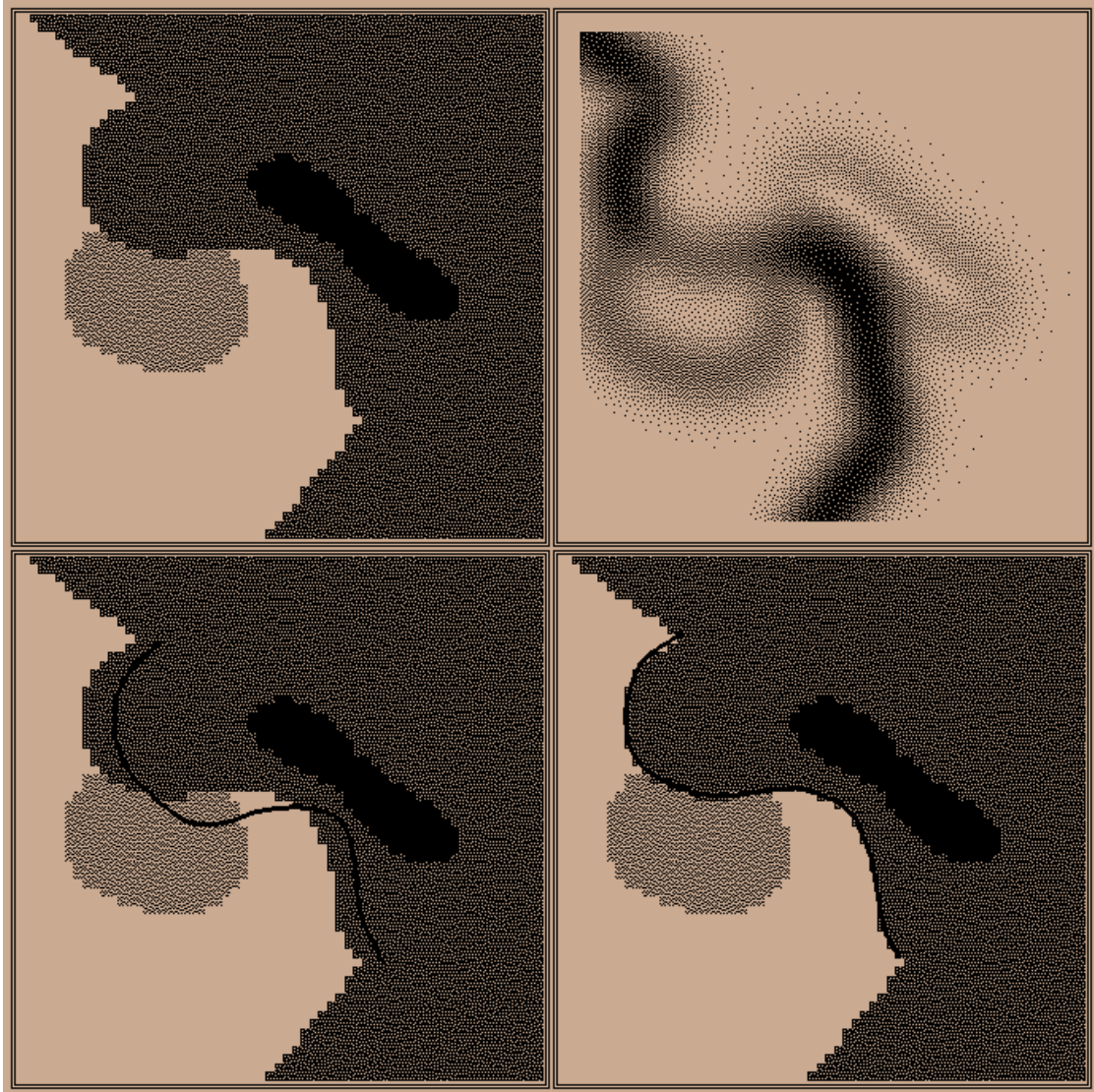


Figure 6.2: Synthetic image open curve example. Top left: Synthetic image ( $64 \times 64$ ). Top right: Gray level gradient magnitude ( $\sigma = 3.0$ ). Bottom left: Initial contour (12 harmonics). Bottom right: Final contour on target shape.

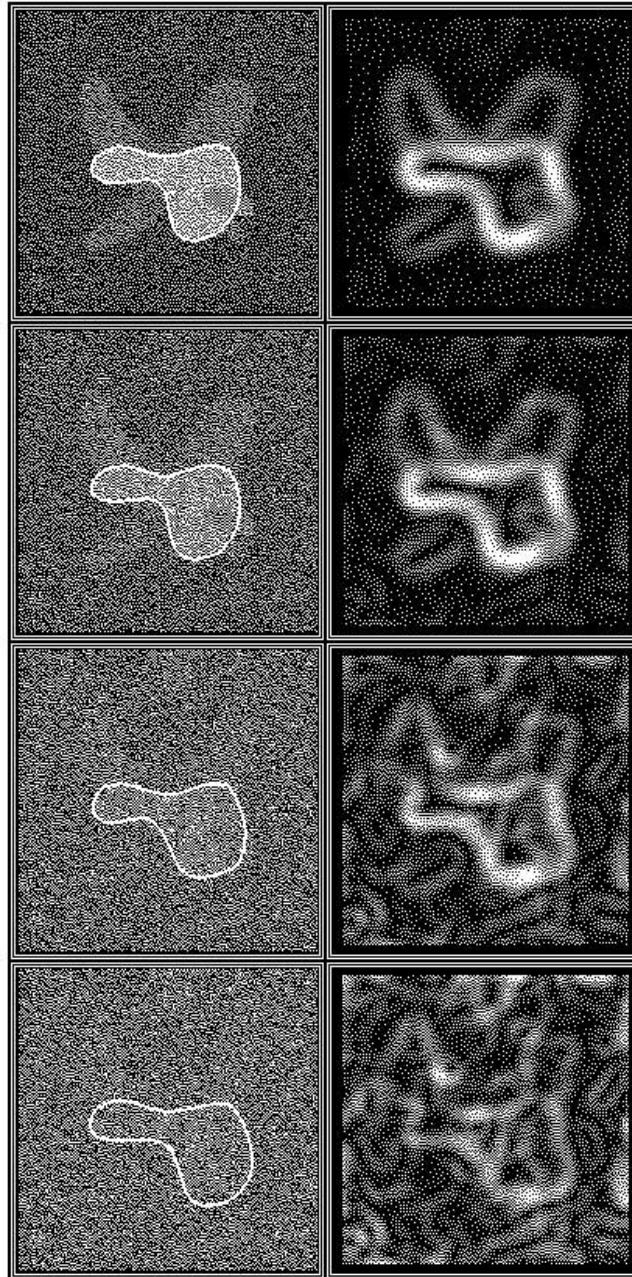


Figure 6.3: Noise experiment images. Left, top to bottom: Image from Figure 6.1 with Gaussian noise added with SNR of 5.0, 2.5, 1.0 and 0.5. Each shown with final contour. Right, top to bottom: Corresponding gray level gradient magnitude ( $\sigma = 3.0$ ).

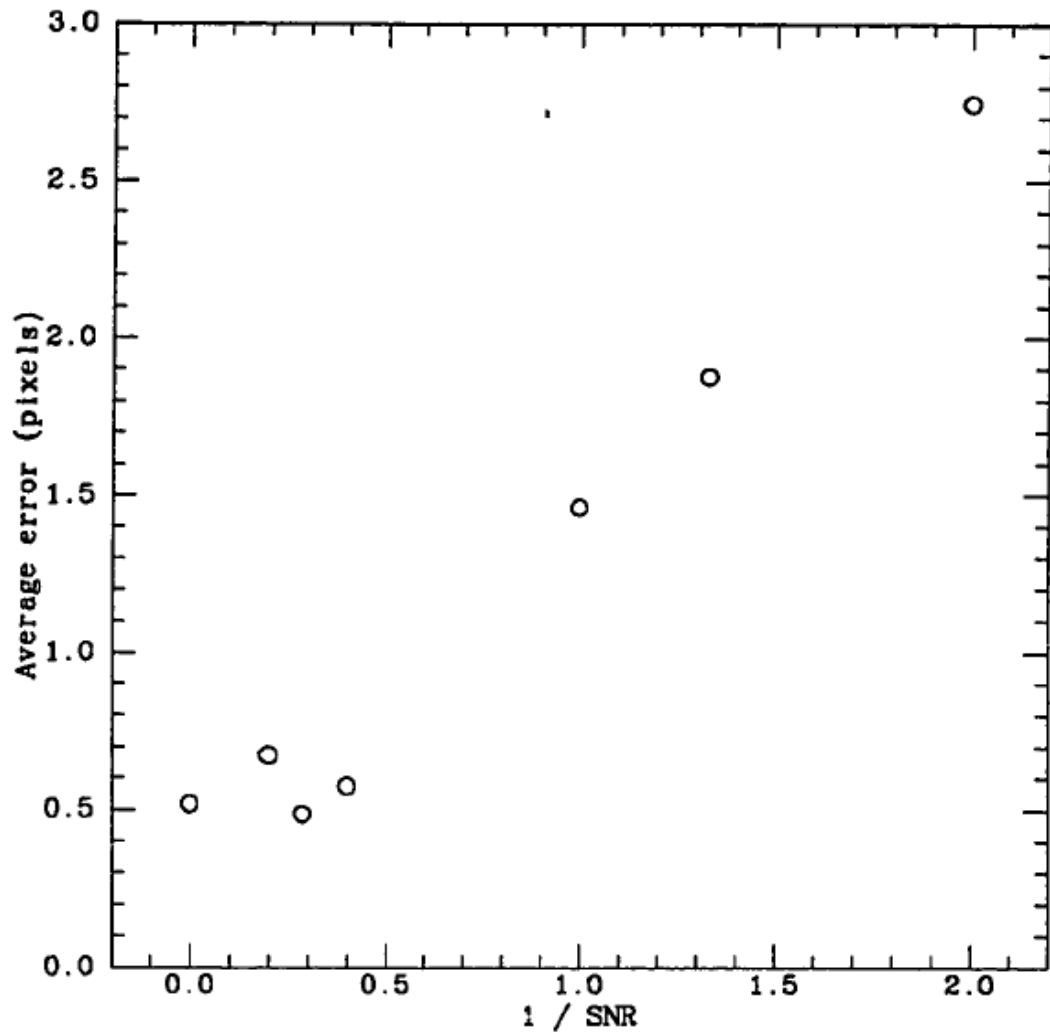


Figure 6.4: Sensitivity to noise experiment.



### 6.2.2 Varying Initial Parameters

The performance is also affected by the values of the initial parameters, that is, the quality of the initial guess. This effect can be investigated in a similar way by running the same problem from different starting points and examining the results. Here we used the same synthetic image with  $\text{SNR} = 2.5$  constructed as above. In this experiment, the parameters varied were vertical translation, scale and rotation. The other parameters were held constant while each of the above three parameters were varied individually. The range of initial curves for the parameters tested are shown in Figure 6.5 with the image used. This image, while synthetic, has a reasonable amount of noise and complication. The resulting optimized boundaries using these different starting points are shown in Figure 6.6. Each parameter has a range within which the solution can be found reliable. Once the parameters are varied beyond that range, the result will converge to nearby features or get stuck far from the proper solution.

### 6.2.3 Varying Bias

This experiment is designed to demonstrate the effect of different prior probability distributions on the optimization process. The purpose of this experiment is to have a demonstrable difference result from two different prior distributions applied to the same image. To show that the difference is due to the bias caused by the probability term in the objective function and not simply the starting point, the distributions will have the same mean value, but different variances. A synthetic Mondrian image, shown in Figure 6.7, was designed containing two similar objects. The light object corresponds to a rotation of the mean prior curve. The dark object underneath it is a scaled version of the mean curve. The prior distribution can be biased towards finding the light object

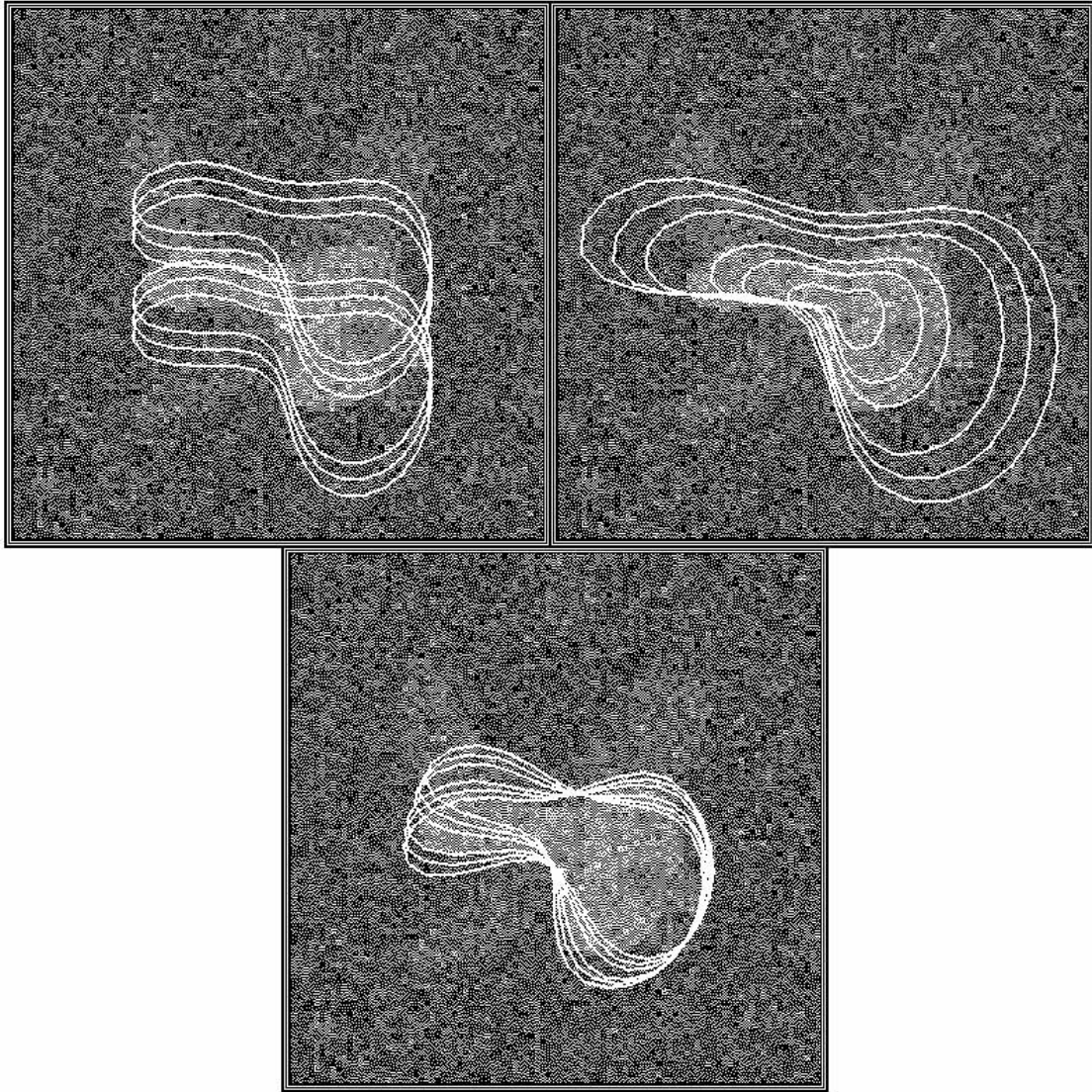


Figure 6.5: Parameter sensitivity experiment parameters. Top left: Image from Figure 6.1 with Gaussian noise added ( $\text{SNR} = 2.5$ ) shown with initial curves for range of vertical translation tested. Top right: Image with initial curves for range of scale tested. Bottom: Image with initial curves for range of rotation tested.

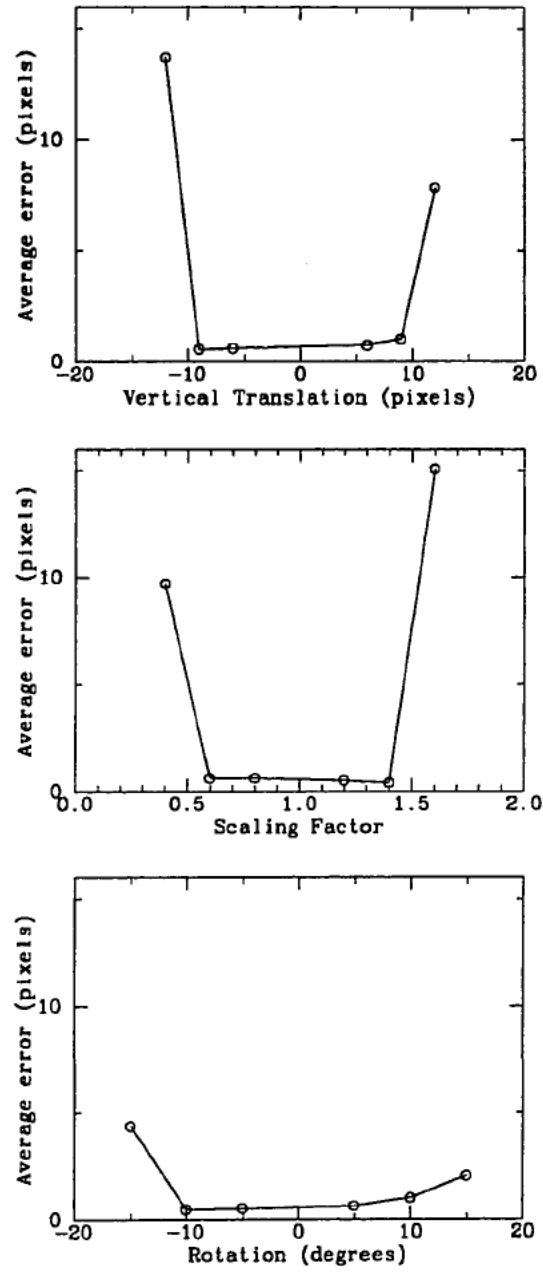


Figure 6.6: Sensitivity to initial parameters experiment.

by having a wide distribution on the rotation parameter and narrow distributions on the others. Conversely, a wide distribution on the scale parameter and narrow distributions on the others will bias the optimization to the dark object. These two different prior distributions were applied to the image and the results are shown in Figure 6.7.

## 6.3 Real Images

The deformable object boundary finding method has been applied to a variety of objects from real images, with an emphasis on heart and brain images. There has been a long history of interest in the automatic analysis of cardiac images and an increasing interest, due to new imaging techniques, in images of the brain [4]. The successful application of this boundary finding technique to cardiac problems has been described [82].

### 6.3.1 Nuclear Medicine

In cardiac imaging, it is necessary to isolate the boundary of the left ventricle of the heart in order to quantify cardiac wall motion and shape. An equilibrium radionuclide angiogram [21] acquired in the left lateral view is shown in Figure 6.8. The gray level in these images is gamma-ray intensity corresponding to a projection in the plane of the viewer of the volume of radioactively-labeled blood at each pixel position. The primary difficulties in finding the left ventricle here are the low spatial resolution of the image ( $48 \times 48$ ) and the fact that the left ventricle overlaps another structure, the right ventricle, which causes an ambiguity in that portion of the boundary. The upper right of Figure 6.8 shows the result of the gray-level gradient operation. The gradient magnitude is displayed as gray level intensity. The initial contour derived from the mean curve of a set of manually traced contours, seen from the same view, is shown in the lower left.

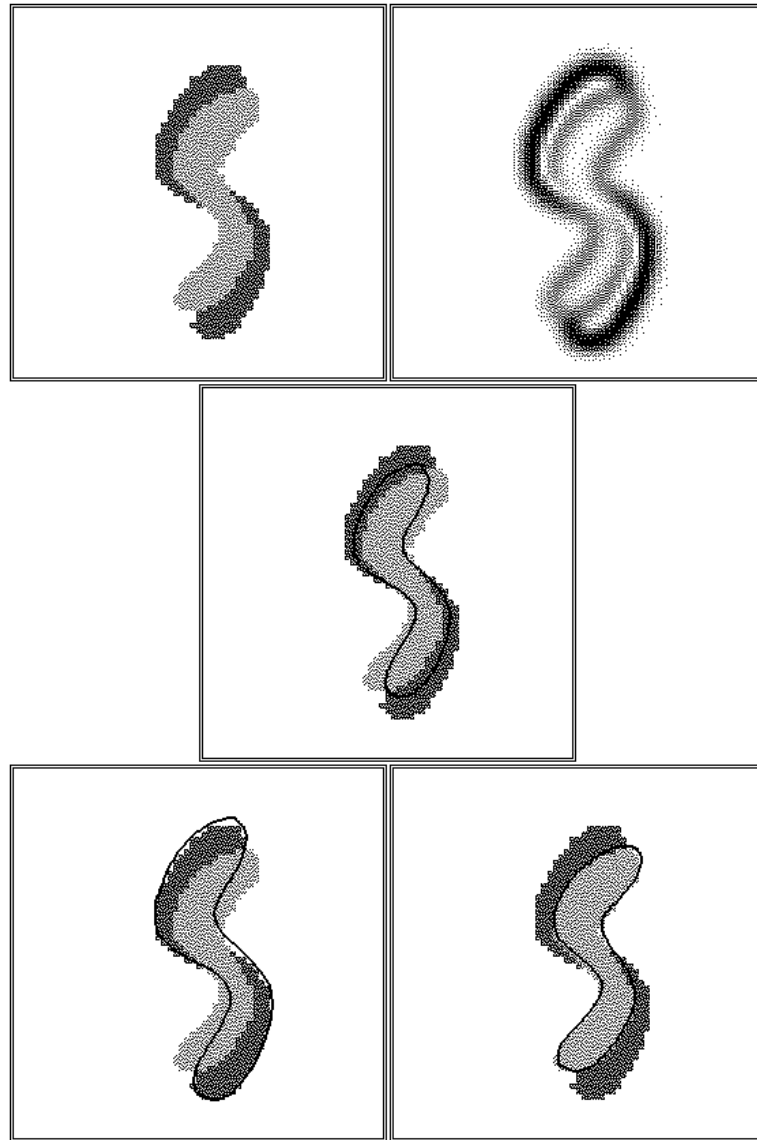


Figure 6.7: Bias experiment. Top left: Synthetic image ( $64 \times 64$ ). Top right: Gray level gradient magnitude ( $\sigma = 1.5$ ). Middle: Initial contour (6 harmonics). Bottom left: Final contour, biased to scaled target shape. Bottom right: Final contour, biased to rotated target shape.

Note that the left-most portion of the initial boundary is blunter and shorter than the edge magnitude image indicates. The final boundary is shown in the lower right. Here, the apex has been extended out to its true location and the regions surrounding the apex more accurately represent the true border. However, the boundary has adhered to a strong gradient at the top of the ventricle that may be spurious. This illustrates its tendency (due to the prior probability term in the objective function) to not move too far from the original contour.

### 6.3.2 Echocardiography

In two-dimensional echocardiography [21], the boundary of an object is rarely indicated by complete, closed groups of edge gradients. Instead, objects are identified by subjective contours [48], that is, contours identified by subjectively grouping incomplete contour information. Gray level in these images represents the strength of echoes of ultrasound waves at each pixel location due to reflection from tissue interfaces and scattering from within tissues. Echocardiograms are particularly difficult to analyze because structures are subject to a high level of noise and tend to have a very scattered and disjointed response. Object boundaries are thus largely made up of organized blobs of signal intensity. An example echocardiogram boundary finding problem is shown in Figure 6.9. The left ventricle, the object to be delineated, is shown in the apical long axis view at end-diastole along with the corresponding magnitude of the intensity gradient. The ultrasound transducer is just above the top center of the image. The probability distributions were generated from a sample of manually traced contours. Note that although this is almost a plausible shape for this boundary, many of the strongest gradients such as those near the lower left part of the contour are completely missed. The final converged solution of the algorithm attains a more acceptable shape in this region, while the

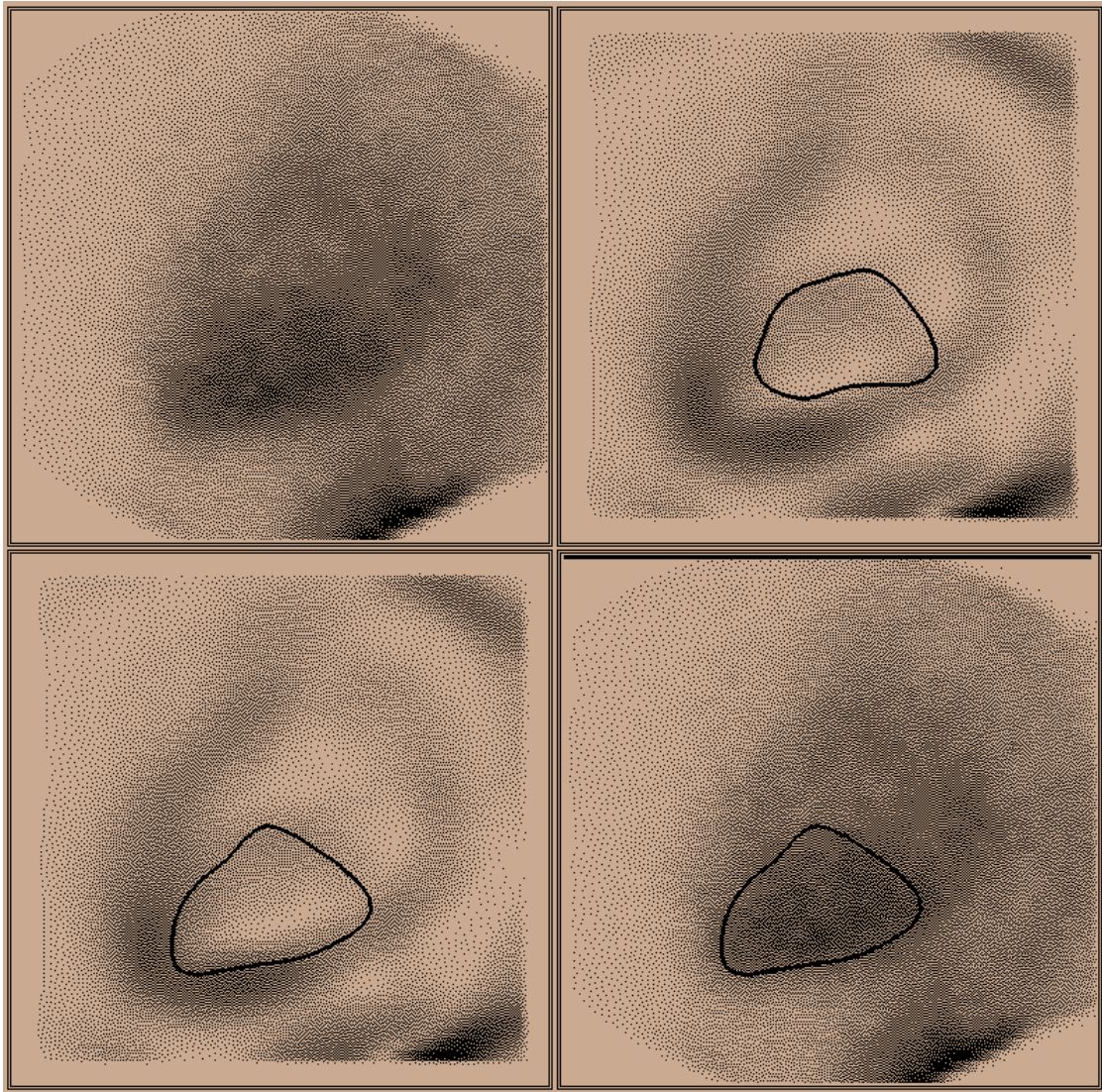


Figure 6.8: Left lateral radionuclide angiogram example. Top left: Radionuclide angiogram ( $48 \times 48$ ). Top right: Gray level gradient magnitude ( $\sigma = 2.0$ ). Bottom left: Initial contour (4 harmonics). Bottom right: Final contour on the endocardium of the left ventricle.

remainder of the contour still adheres to the most plausible portions in the other regions. Note the extension of the apex of the perceived contour off the end of the upper leftmost portion of the image. Apical dropout of this sort often occurs in echocardiograms taken from this view. Thus, this complete boundary model approach may have further merit in estimating the shape of the complete left ventricle even in regions where it is entirely missing.

### 6.3.3 Magnetic Resonance Imaging

The algorithm has also been tested on magnetic resonance images [40]. Gray level in these images represents a combination of the nuclear magnetic resonance properties of the hydrogen nuclei at each pixel position. The results of the boundary finding program applied to the problem of delineating the corpus callosum in the human brain from magnetic resonance images are shown in Figure 6.10. The size and shape of the corpus callosum are important in the study of its relationship to cognitive function [90]. In these images, the corpus callosum is separated from the rest of the brain by a dark line. Thus, a measure that responded to lines was used instead of the gray-level gradient. In this case, we used the positive magnitude of the Laplacian of the Gaussian, which responds to both lines and edges. The final contour succeeds in delineating the structure properly. Because of the narrow, curved shape of the corpus callosum, six harmonics are needed to accurately represent it.

Magnetic resonance is becoming more and more important for cardiac imaging as acquisition rates increase into the range required for imaging the moving heart. In Figure 6.11, an oblique sagittal image of the heart roughly equivalent to the long-axis view is shown. The inner or endocardial wall of the left ventricle is the target structure in the image. The results of the boundary-finding procedure are shown.



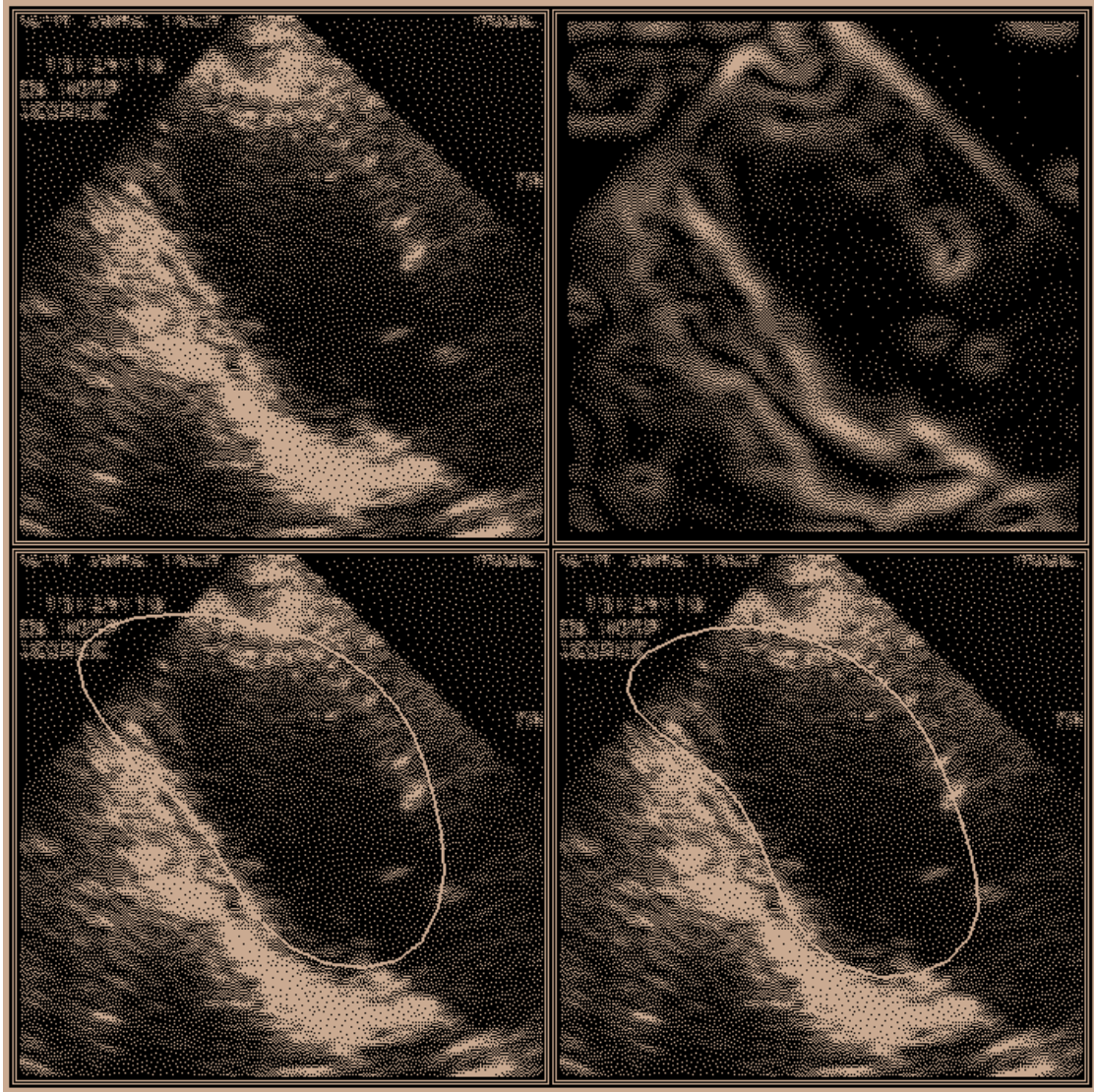


Figure 6.9: Long-axis echocardiogram example. Top left: Echocardiogram ( $170 \times 170$ ). Top right: Gray level gradient magnitude ( $\sigma = 3.0$ ). Bottom left: Initial contour (4 harmonics). Bottom right: Final contour on the endocardium of the left ventricle.

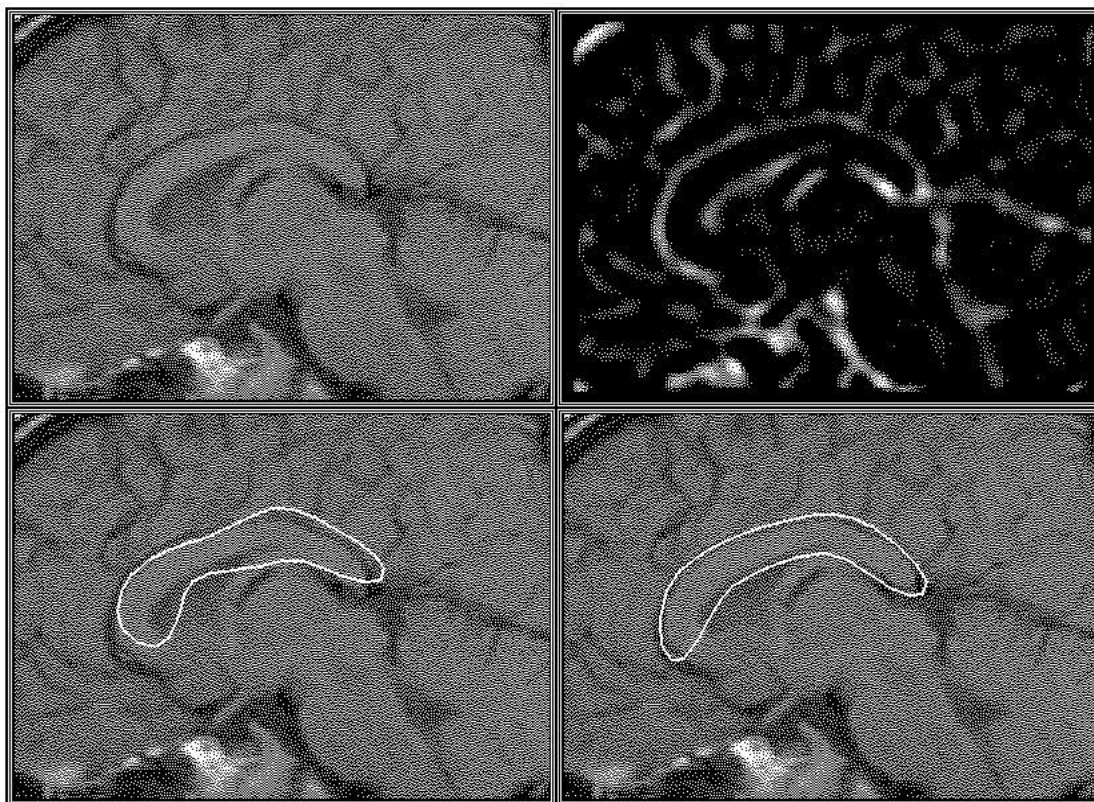


Figure 6.10: Magnetic resonance mid-brain sagittal image example. Top left: Magnetic resonance image ( $146 \times 106$ ). Top right: Positive magnitude of the Laplacian of the Gaussian ( $\sigma = 2.2$ ). Bottom left: Initial contour (6 harmonics). Bottom right: Final contour on the corpus callosum of the brain.

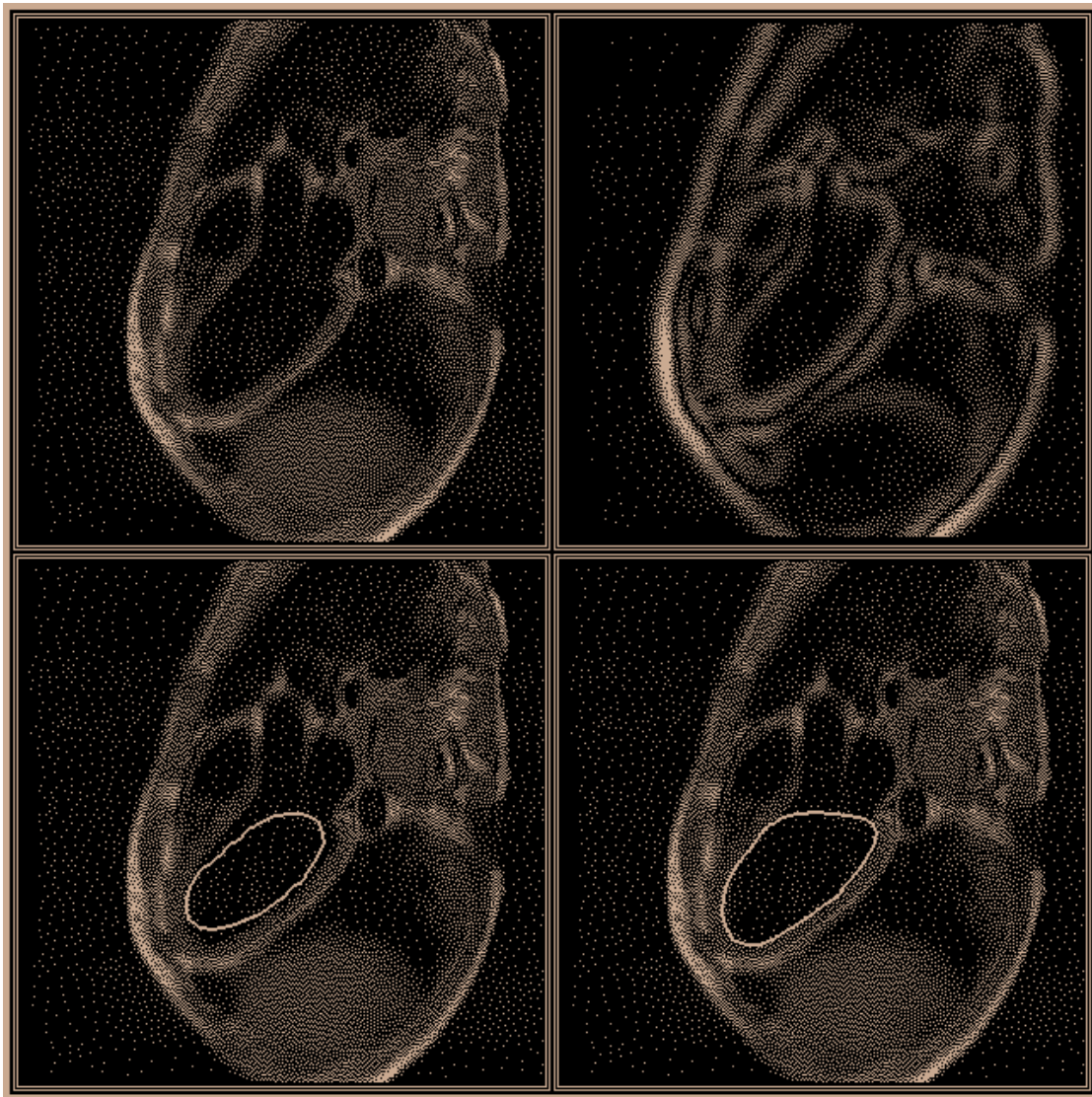


Figure 6.11: Magnetic resonance oblique sagittal cardiac image example. Top left: Magnetic resonance image ( $256 \times 256$ ). Top right: Gray level gradient magnitude ( $\sigma = 4.0$ ). Bottom left: Initial contour (4 harmonics). Bottom right: Final contour on the endocardium of the left ventricle.

In Figure 6.12, a transaxial cardiac image shows a section through the left ventricular wall. Here, the endocardial and epicardial (outer) walls of the left ventricle are objects to be delineated. The results of the two separate optimizations are shown.

In Figure 6.13, another magnetic resonance cardiac example is shown with somewhat poorer image quality compared to the previous example. This is a transverse section through the body showing the endocardium of the left ventricle. As before, the inner wall is well delineated by the final optimized contour.

### 6.3.4 Summary

This method appears to work well delineating structures in real images, and to be relatively insensitive to the problems of broken boundaries and spurious edges from nearby objects. The start position has to be close enough to the true boundary in order to avoid false local minima. This region of success or capture about the true boundary depends on the quality of the image, the degree of smoothing and the particular problem. False minima can be distinguished, however, both visually and by the relative value of the objective function. Experimentation shows good results on a variety of medical images. The flexibility of the model both in terms of its probabilistic nature and the parametric representation make this an attractive method for boundary finding.

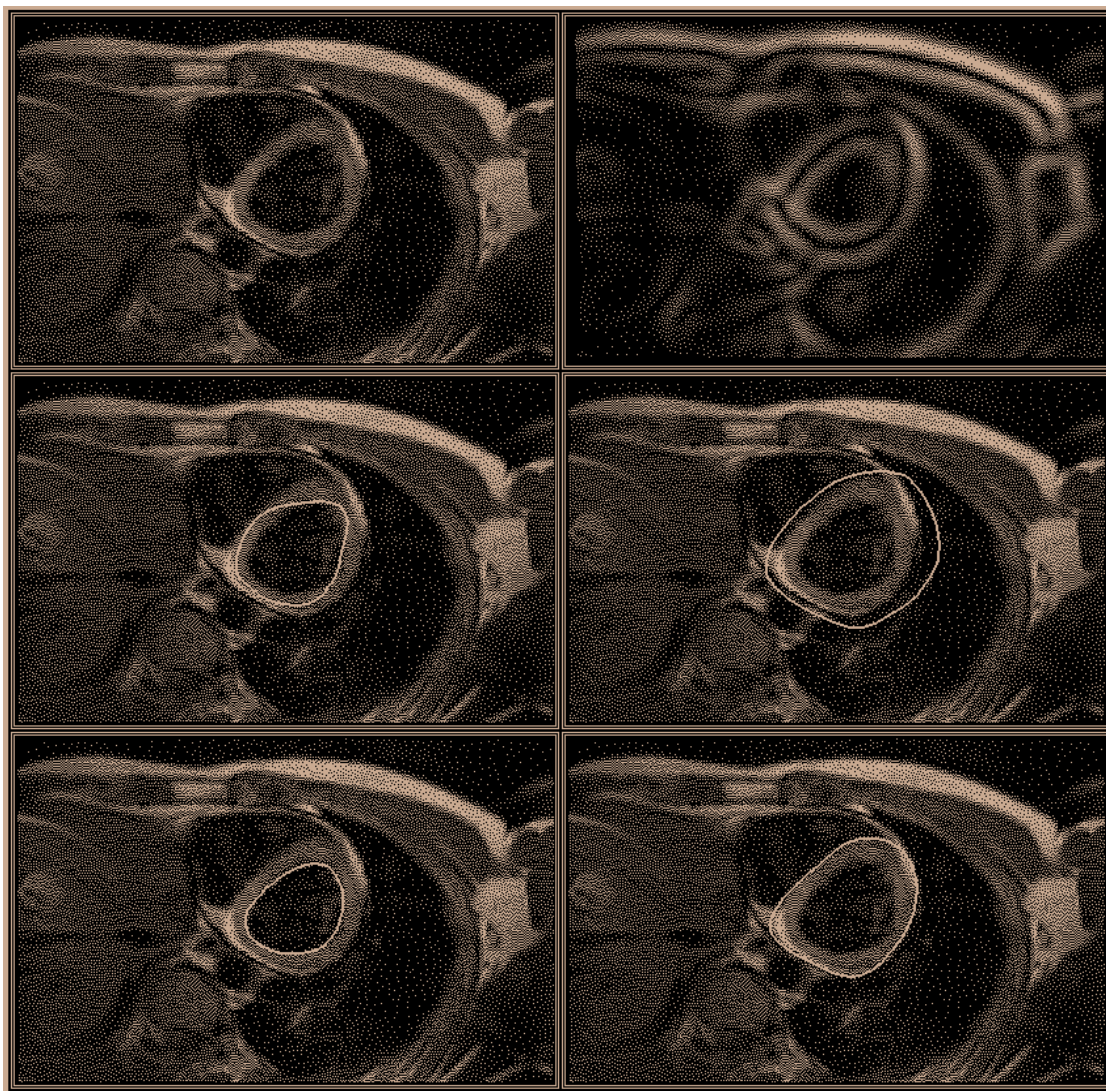


Figure 6.12: Magnetic resonance transaxial cardiac image example. Top left: Magnetic resonance image ( $256 \times 156$ ). Top right: Gray level gradient magnitude ( $\sigma = 4.0$ ). Middle left: Initial contour on endocardium (4 harmonics). Bottom left: Final contour on endocardium of the left ventricle. Middle right: Initial contour on the epicardium (4 harmonics). Bottom right: Final contour on the epicardium of the left ventricle.

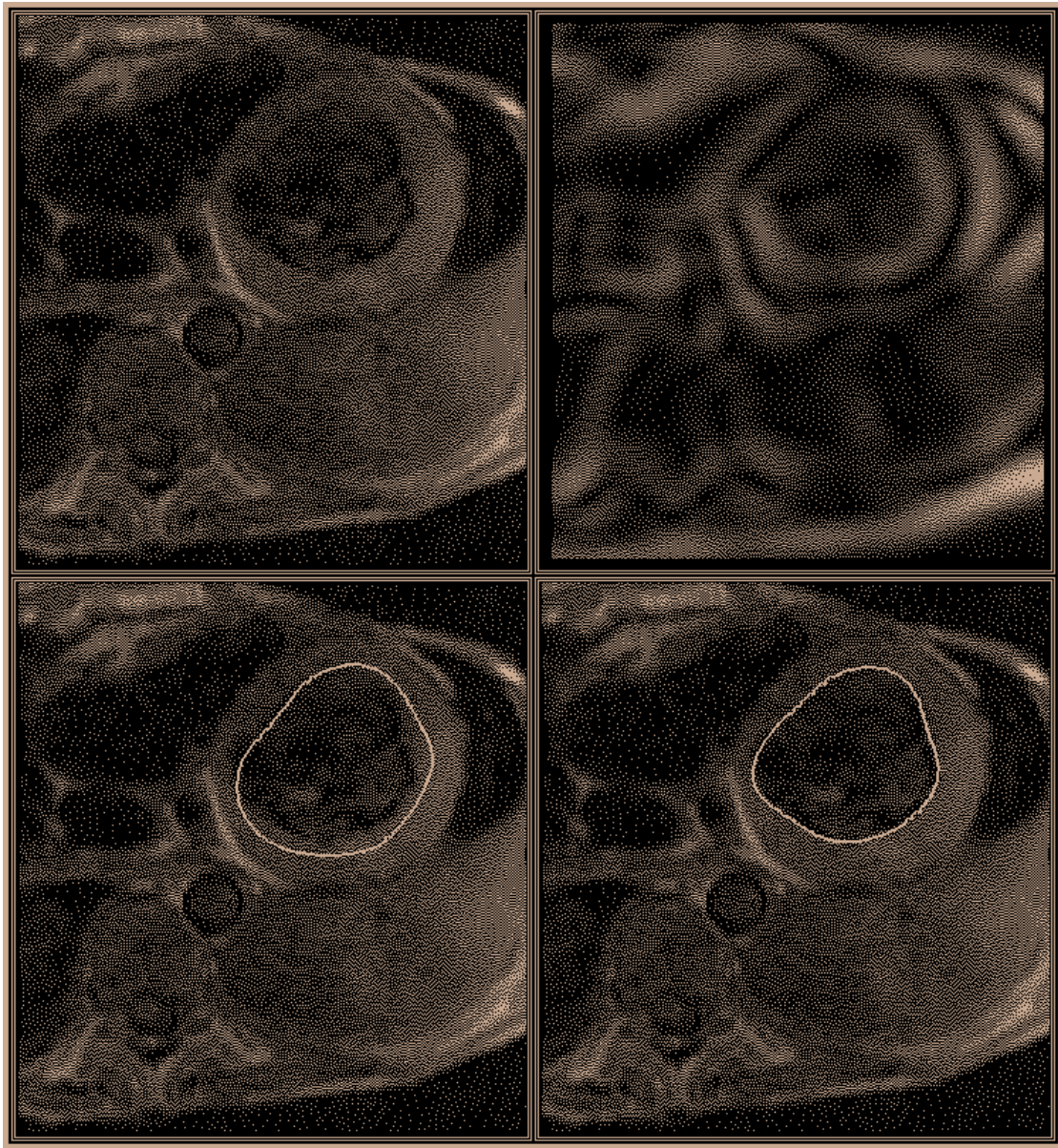


Figure 6.13: Magnetic resonance oblique sagittal cardiac image example. Top left: Magnetic resonance image ( $128 \times 140$ ). Top right: Gray level gradient magnitude ( $\sigma = 4.0$ ). Bottom left: Initial contour (5 harmonics). Bottom right: Final contour on the endocardium of the left ventricle.

## Chapter 7

# Extensions

The general framework presented here can be extended in many ways. This chapter will consider the initial steps for some of those extensions. These include incorporating constraints, extending the parametrization to curves and surfaces in three-dimensions and applying the technique to temporal image sequences and multimodality boundary finding.

### 7.1 Boundary Constraints

Sections of the boundary that are ill-defined can be a problem when both the image and the prior information fail to resolve them. Manual intervention may be necessary in these cases for practical reasons.

It is therefore desirable to be able to indicate sections of the boundary to which the optimization process must hold. These sections could be indicated manually by the user who has specific knowledge of the image domain. The indicated sections are constraints on the boundary that the optimization must satisfy. This could be accomplished with a

penalty function approach [57]. A simple approach is to adjust the boundary measure as follows:

$$b'(x, y) = b(x, y) + \sum_{i=1}^{N_{\text{points}}} l(x_i, y_i) \quad (7.1)$$

$$l(x_i, y_i) = ce^{-\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}} \quad (7.2)$$

where the points  $(x_i, y_i)$  are the constraint points.

In order to make it a hard constraint, solve the optimization problem with this boundary measure and let  $c \rightarrow \infty$  and  $\sigma \rightarrow 0$ . This procedure makes convergence difficult, so instead choose a large value for  $c$  and a  $\sigma$  large enough to draw the boundary to the constraint points. In Figure 7.1, an example is shown using this method. The top section of the target object (the valve plane of the left ventricle) is not indicated in the image. The final boundary can be adjusted by including a boundary constraint in this region. The boundary constraint has the effect of pulling up the top edge of the boundary.

## 7.2 Three dimensions

Three-dimensional image analysis is becoming more important due to the availability of practical three-dimensional imaging technologies and laser and acoustic range data capabilities [66]. In diagnostic radiology, magnetic resonance imaging (MRI), single photon emission computed tomography (SPECT) and positron emission tomography (PET) [28] have necessitated the development of techniques for handling three-dimensional images. Confocal microscopy is also a growing source of three-dimensional images [89].

As mentioned in Chapter 3, the main approaches to three-dimensional parametric modeling have been superquadrics, spherical harmonics and generalized cylinders. All of



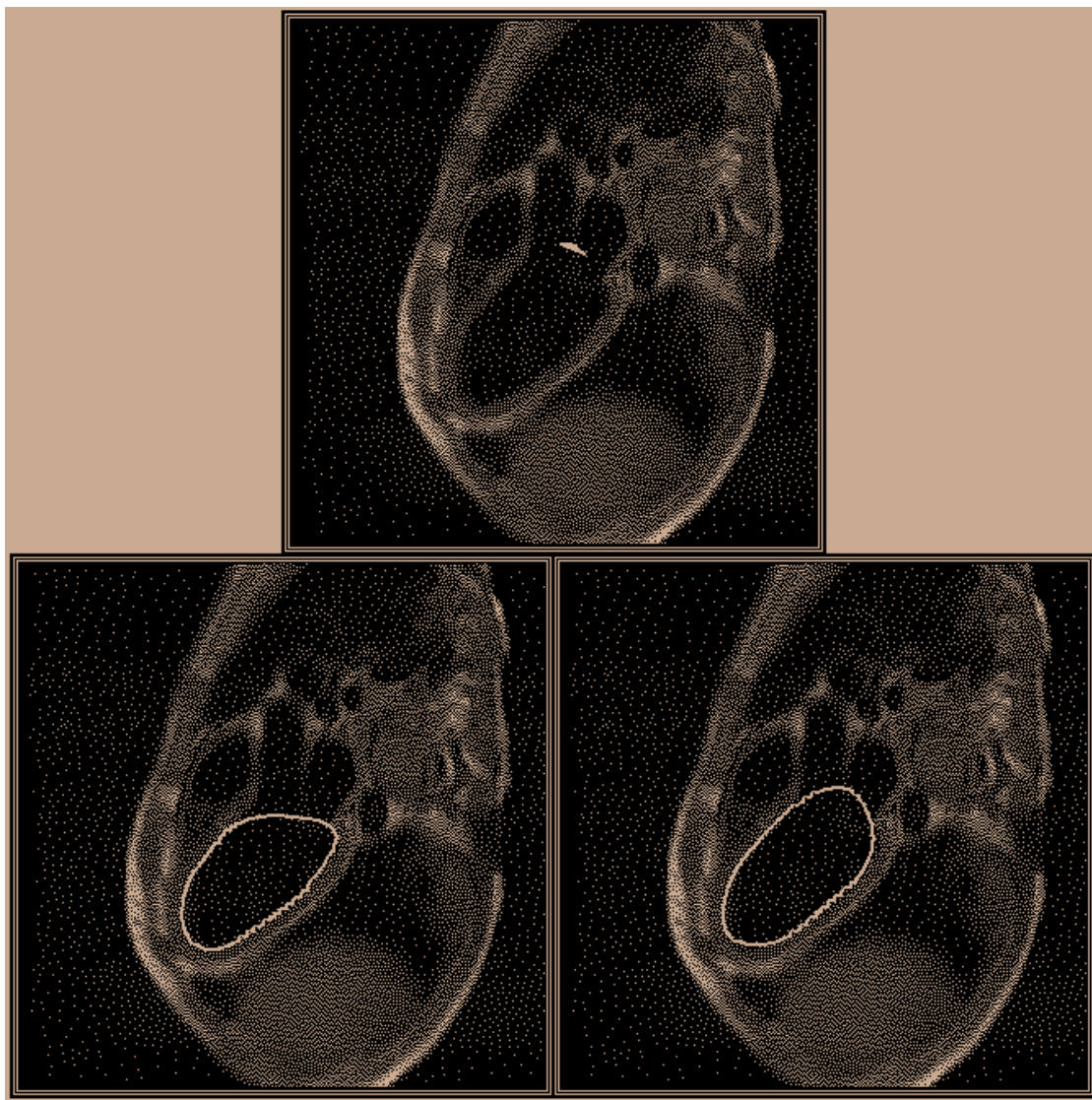


Figure 7.1: Boundary constraint example. Top: Magnetic resonance oblique sagittal cardiac image from Figure 6.11 with boundary constraint points, indicated by the short bright line in the center of the image. Bottom left: Final contour without the boundary constraint. Bottom right: Final contour with boundary constraint.

these parametrizations are restricted to a limited class of objects. The parametrizations described below are an attempt to develop a more general method of describing objects in three dimensions.

### 7.2.1 Curves in three dimensions

The two-dimensional curve parametrization described in Chapter 3 can be easily extended to three-dimensional curves [54]. The curve is now represented by three functions of  $t$  using the same basis functions as before.

$$v(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \\ e_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \\ e_k & f_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \quad (7.3)$$

Each term in the summation is now the parametric form for an ellipse in three-space. The ellipse properties can be derived from these raw coefficients as before in order to develop an invariant representation. Linear structures in three-dimensional images could be represented in this way.

### 7.2.2 Surfaces in three dimensions

To use a Fourier representation for a function of two variables, the following basis can be used [87]:

$$\begin{aligned} \phi = \{ & 1, \cos m\alpha, \sin m\alpha, \cos l\beta, \sin l\beta, \dots, \\ & \cos m\alpha \cos l\beta, \sin m\alpha \cos l\beta, \\ & \cos m\alpha \sin l\beta, \sin m\alpha \sin l\beta, \dots \quad (m = 1, 2, \dots; l = 1, 2, \dots) \} \end{aligned} \quad (7.4)$$

This can be used as the basis for parametrizations of surfaces in three dimensions. Such surfaces can be described explicitly by three functions of two surface parameters,  $x(\alpha, \beta)$ ,  $y(\alpha, \beta)$  and  $z(\alpha, \beta)$  where  $\alpha$  and  $\beta$  vary over the surface and  $x$ ,  $y$ , and  $z$  are the associated Cartesian coordinates. While the choice of  $\alpha$  and  $\beta$  is obvious for simple surfaces such as spheres (use latitude and longitude) or cylinders (use longitude and height), very complicated surfaces will require some further analysis to determine the appropriate tessellation. Axis transforms [32] may provide a way of determining the overall structure on which to base the surface parametrization.

There are three classes of simple surfaces in three dimensions that will be useful to describe: open surfaces (with one edge), tubes (open surfaces with two edges) and closed surfaces (no edges). The simplest type of surface to describe is actually the torus (a closed tube) but it is of limited use in medical images. It is formed using the entire basis shown in Equation 7.5. The result is a torus because both surface parameters are forced to be periodic. Thus, it would be represented as follows:

$$\begin{bmatrix} x(\alpha, \beta) \\ y(\alpha, \beta) \\ z(\alpha, \beta) \end{bmatrix} = \begin{bmatrix} a_{x,0,0} \\ a_{y,0,0} \\ a_{z,0,0} \end{bmatrix} + \sum_{m=1}^{\infty} \begin{bmatrix} a_{x,m,0} & b_{x,m,0} \\ a_{y,m,0} & b_{y,m,0} \\ a_{z,m,0} & b_{z,m,0} \end{bmatrix} \begin{bmatrix} \cos m\alpha \\ \sin m\alpha \end{bmatrix} + \sum_{l=1}^{\infty} \begin{bmatrix} a_{x,0,l} & b_{x,0,l} \\ a_{y,0,l} & b_{y,0,l} \\ a_{z,0,l} & b_{z,0,l} \end{bmatrix} \begin{bmatrix} \cos l\beta \\ \sin l\beta \end{bmatrix} +$$

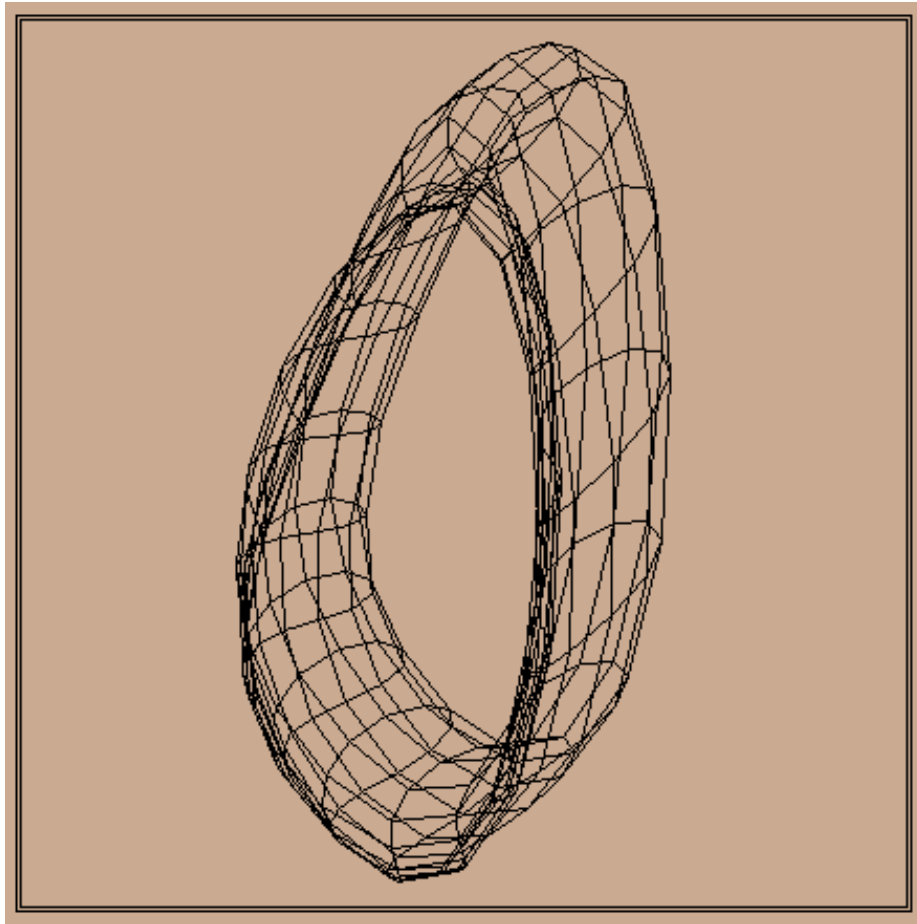


Figure 7.2: Torus surface example using first and second harmonics.

$$\sum_{m=1}^{\infty} \sum_{l=1}^{\infty} \begin{bmatrix} a_{x,m,l} & b_{x,m,l} & c_{x,m,l} & d_{x,m,l} \\ a_{y,m,l} & b_{y,m,l} & c_{y,m,l} & d_{y,m,l} \\ a_{z,m,l} & b_{z,m,l} & c_{z,m,l} & d_{z,m,l} \end{bmatrix} \begin{bmatrix} \cos m\alpha \cos l\beta \\ \sin m\alpha \cos l\beta \\ \cos m\alpha \sin l\beta \\ \sin m\alpha \sin l\beta \end{bmatrix} \quad (7.5)$$

An example torus surface using this parametrization, with terms up to  $l = 2$  and  $m = 2$ , is shown in Figure 7.2. The other three types of surfaces can be described using subsets

of the above basis.

Representing open surfaces with the above basis is complicated by the periodicity property. Since the surface is open, a straightforward representation of the surface would result in discontinuities at the boundary. This problem can be addressed by analogy to the approach used for open curves, as described in Section 3.3.2. Thus, these discontinuities can be avoided by having the two surface parameters start at one side of the surface, trace along the surface to the other end, and then retrace the surface in the opposite direction to create a closed path.

This results in a function  $x(\alpha, \beta)$  that is even and thus only the purely even terms,  $a_{x,0,0}$ ,  $a_{x,m,0}$ ,  $a_{x,0,l}$  and  $a_{x,m,l}$  are nonzero. This also holds for  $y(\alpha, \beta)$  and  $z(\alpha, \beta)$ . The converse is also true; that is, any expansion with only those terms nonzero for all  $l$  and  $m$  results in an even function and thus describes an open surface. We are therefore effectively restricting the basis to include only even functions of both  $l$  and  $m$ .

$$\begin{aligned} \phi_{\text{open}} = \{ & 1, \cos m\alpha, \cos l\beta, \dots, \\ & \cos m\alpha \cos l\beta, \dots \quad (m = 1, 2, \dots; l = 1, 2, \dots) \} \end{aligned} \quad (7.6)$$

Open surfaces are useful for a wide variety of structures including organs with one opening and flat structures. An example open surface using this parametrization, with terms up to  $l = 2$  and  $m = 2$ , is shown in Figure 7.3.

Tubes require the open representation along one of the surface parameters and the closed representation along the other. This results in the following basis which is even

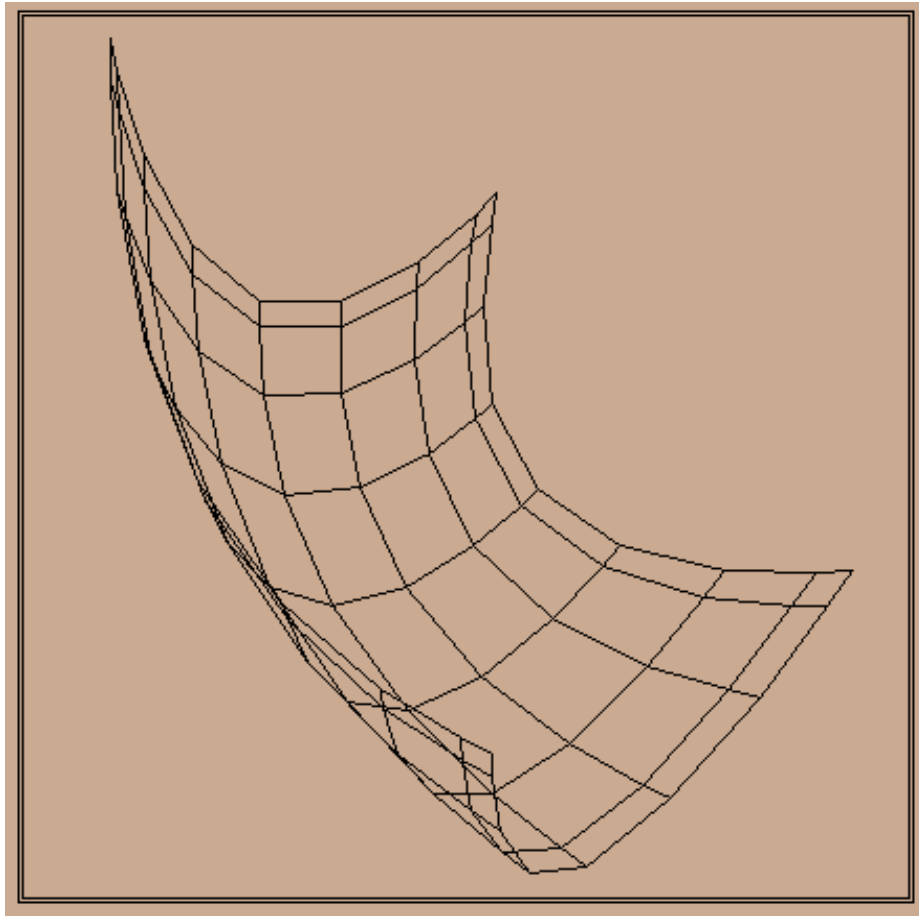


Figure 7.3: Open surface example using first and second harmonics.

in  $l$  and unrestricted in  $m$ :

$$\begin{aligned} \phi_{\text{tube}} = & \{1, \cos l\beta, \sin m\alpha, \cos m\alpha, \dots, \\ & \cos m\alpha \cos l\beta, \sin m\alpha \cos l\beta, \dots \quad (m = 1, 2, \dots; l = 1, 2, \dots)\} \end{aligned} \quad (7.7)$$

Thus the only nonzero terms are  $a_{x,0,0}$ ,  $a_{x,0,l}$ ,  $a_{x,m,0}$ ,  $b_{x,m,0}$ ,  $a_{x,m,l}$  and  $b_{x,m,l}$  and the corresponding  $y$  and  $z$  terms. Tubes are an extension of traditional generalized

cylinders [2] used to represent elongated objects but usually limited to simple cross-sections such as circles. The cross-section of a tube can be an arbitrary closed curve. Tubes are useful for vessels and other elongated objects. They are also useful for temporal sequences of planar images, where the third dimension is time (see Section 7.3), and multimodal images, where the third dimension is modality (see Section 7.4). An example tube surface using this parametrization, with terms up to  $l = 2$  and  $m = 2$ , is shown in Figure 7.4.

A closed surface can be represented by considering tubes whose ends close up to a point at both ends instead of being open. This is done by expressing  $x$  and  $y$  using the following basis:

$$\begin{aligned} \phi_{\text{closed-xy}} = \{ & 1, \sin l\beta, \dots, & (7.8) \\ & \cos m\alpha \sin l\beta, \sin m\alpha \sin l\beta, \dots & (m = 1, 2, \dots; l = 1, 2, \dots)\} \end{aligned}$$

thus forcing both functions to constants at the extremes of  $l$  and expressing  $z$  using the same basis as for tubes:

$$\begin{aligned} \phi_{\text{closed-z}} = \{ & 1, \cos l\beta, \sin m\alpha, \cos m\alpha, \dots, & (7.9) \\ & \cos m\alpha \cos l\beta, \sin m\alpha \cos l\beta, \dots & (m = 1, 2, \dots; l = 1, 2, \dots)\} \end{aligned}$$

Closed surfaces are useful for organs and other structures with no prominent openings. An example tube surface using this parametrization, with terms up to  $l = 2$  and  $m = 2$ , is shown in Figure 7.5. A geometric interpretation may exist for each harmonic based on these shape parameters which would allow for invariance to view transformations.

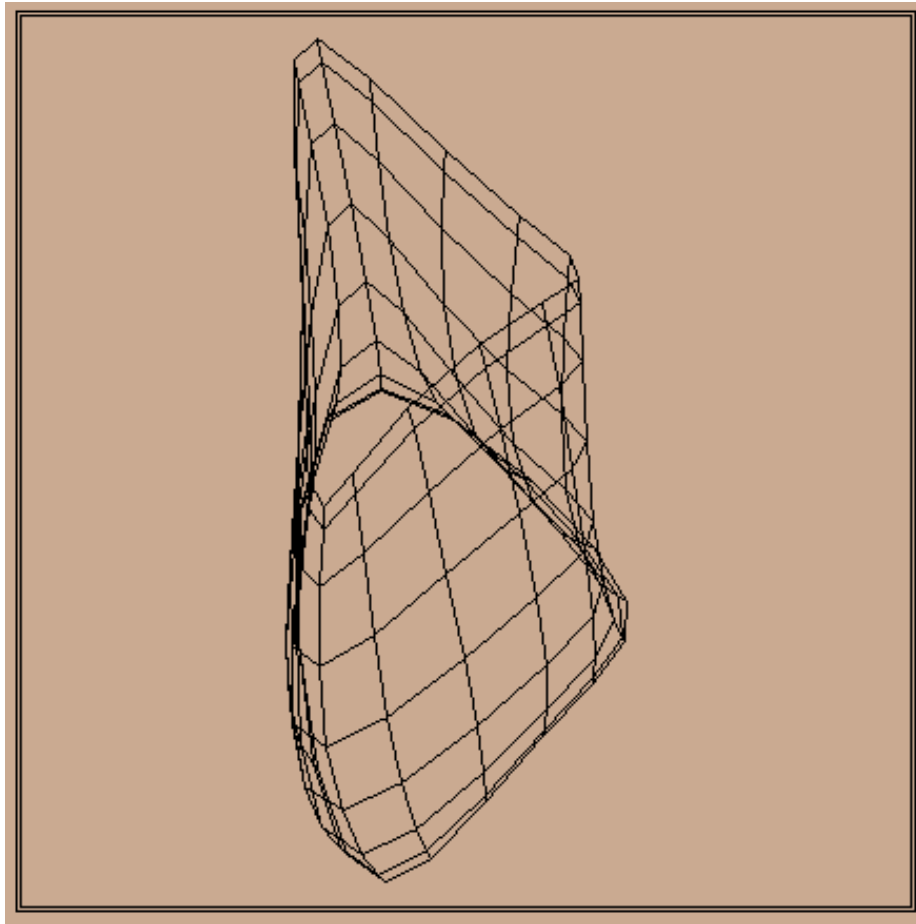


Figure 7.4: Tube surface example using first and second harmonics.

### 7.3 Temporal sequences

Temporal image sequence analysis is important in general because it allows for the detailed study of dynamic phenomena from images. The current system can be adapted for the boundary-based study of object motion in a number of different ways. By delineating an object in successive temporal image frames, the motion of its boundary can be inferred. Note that if points on successive boundaries can be identified, this represents an approach



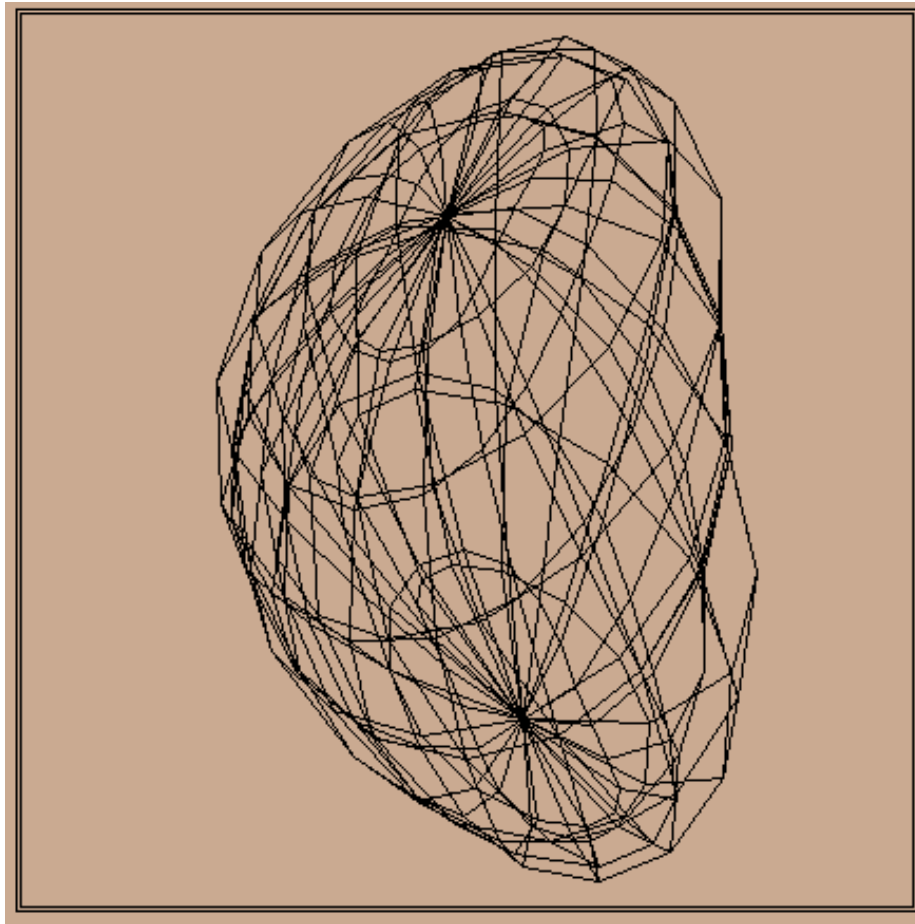


Figure 7.5: Closed surface example using first and second harmonics.

to general, non-rigid object motion, a relatively unexplored and new area of research [36, 68].

A relatively simple approach to temporal sequences is first to solve the delineation problem on the initial frame of the sequence. Then, that solution can be passed to the next frame as the initial guess. The rationale for this approach is that the object would have only moved a small distance between frames, given some limited velocity, and thus the boundary in each frame should be close to each preceding frame. In Figure 7.6 we

show this approach applied to a cardiac motion sequence from magnetic resonance. This technique could benefit from the inclusion of a model of expected motion. This could be simply an interframe smoothness constraint or a more detailed motion model.

As mentioned in Section 7.2.2, the tube model could be effectively applied to temporal sequences. This would allow for unified representation for both shape and motion prior information, where shape is represented in the two spatial dimensions, and motion is represented by the temporal dimension along the axis of the tube. A boundary representation for a temporal sequence of three-dimensional images is even more complex. Such a boundary would be modeled by a three-dimensional object in four dimensions.

## 7.4 Multimodality

Finding boundaries in images of the same object taken from different modalities is a problem of practical importance because of the additional, and often complementary, information available by using more than one imaging method. Because of their different views and imaging properties, images from different modalities must be registered in order to be comparable. This will, in general, involve translation, rotation, scaling and, in general, warping. This transformation can be determined by finding corresponding points in the images using either natural landmarks or fiducial marks placed in the image to establish a coordinate system. More general techniques involve affine [70] or elastic matching [6]. Once this transformation is determined, the boundary finding problem can be treated in the same way as the temporal sequence problem and could be solved using either an interframe smoothness constraint or the tube model.

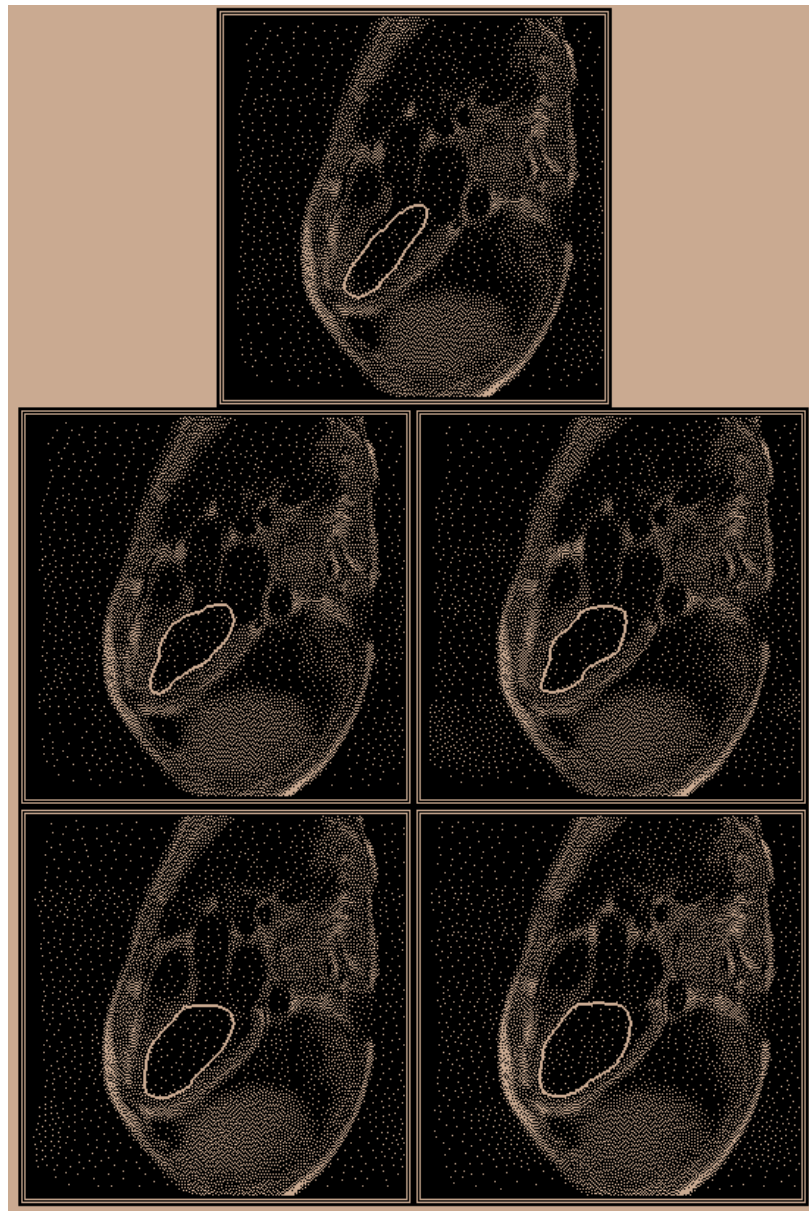


Figure 7.6: Magnetic resonance oblique sagittal cardiac image motion example. Top: Magnetic resonance image ( $256 \times 256$ ) with initial contour (4 harmonics). Middle left: Final contour on the endocardium of the left ventricle (frame 1). Middle right: Final contour (frame 2). Bottom left: Final contour (frame 3). Bottom right: Final contour (frame 4).

## Chapter 8

### Summary

This work presents a general boundary finding system for images of simple natural objects. The goal of this work was to incorporate model-based information into boundary finding for continuously deformable objects. It was found to work well from testing on real and synthetic images. The idea of augmenting a shape parametrization with probabilistic information may have applications to other areas of computer vision for matching or feature detection.

There are, of course, areas of potential improvement for this work. Further analysis of the correlation between the values of the calculated boundary measure would help to better understand the correlation's effect on the match measure. Allowing for correlation in the prior probability for the parameters could also be included and studied. In order to speed up the optimization for relative parameters, the calculation of the gradient of the objective function with respect to the relative parameters could be attempted, in spite of the complexity. Since one of the main benefits of the relative parameters is the isolation of the view parameters (translation, rotation and scale), perhaps a simpler transformation that isolates the view parameters could be devised. This work would also benefit from

the incorporation of a method to determine the initial view, that is, the translation and rotation. Instead of assuming that the prior mean values for these parameters will be close enough to the correct values, some kind of preprocessing or cooperative processing could be included, aimed at determining these parameters. One approach would be to exhaustively search through the possible translations and rotations for the best initial match. If this were done at a low resolution, the computation might not be excessive. Additional information, such as constraints between objects, might also help to guide the initial placement.

Future directions include development and implementation of the ideas presented in Chapter 7. In particular, the surface finding and motion analysis discussed represent important and relatively unexplored new areas. Another extension, not mentioned in Chapter 7, is to other parametrizations. The framework presented here could be used with other shape parametrizations with computational properties similar to the elliptic Fourier representation. In particular, a representation better suited to objects with straight sides and corners would be desirable for man-made objects. This could involve a piecewise parametrization. Other work has arisen using some of the ideas presented here. A multilevel optimization object finder incorporating inter-object constraints has been developed based on a similar framework [16]. Work is also being done using the boundary finder and some of the ideas from this work for deformable object motion [68].

# Appendix A

## The Gaussian

The Gaussian is used throughout this work as both a smoothing filter and as a probability density. The Gaussian was first discussed by De Moivre in 1733 as the limit of the binomial distribution [24]. As a filter, the Gaussian has many advantages. The two-dimensional Gaussian is the only separable rotationally symmetric two-dimensional operator. The Gaussian has the smallest space-frequency width product. It is the optimally localized function that is smooth in both space and frequency. It has the same form as its Fourier transform.

It is also the natural form for a probability density. The Central Limit Theorem shows that additive random variables converge to a Gaussian distribution under fairly general conditions [69]. Also, among probability density functions having a given variance, the Gaussian is the one with the maximum entropy [20, 79].

# Bibliography

- [1] I. E. Abdou and W. K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proc. IEEE*, 67(5):753–763, May 1979.
- [2] G. J. Agin and T. O. Binford. Computer description of curved objects. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 629–640, 1973.
- [3] A. A. Amini, S. Tehrani, and T. E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Proc. Second Int. Conf. Comp. Vision*, pages 95–99, December 1988.
- [4] N. C. Andreasen. Brain imaging: Applications in psychiatry. *Science*, 239:1381–1388, March 1988.
- [5] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.
- [6] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Comp. Vision Graphics Image Proc.*, 46:1–21, 1989.
- [7] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proc. First Int. Conf. Comp. Vision*, pages 231–239, June 1987.

- [8] D. H. Ballard. *Hierarchical Recognition of Tumors in Chest Radiographs with Computer*. Birkhäuser Verlag, Basel, 1976.
- [9] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [10] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, 1982.
- [11] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.
- [12] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, July 1984.
- [13] F. Bergholm. Edge focusing. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(6):726–741, 1987.
- [14] P. J. Besl. Geometric modeling and computer vision. *Proc. IEEE*, 76(8):936–958, 1988.
- [15] J. Blokland, A. Vossepoel, A. Bakker, and E. Pauwels. Delineating elliptical objects with an application to cardiac scintigrams. *IEEE Trans. Medical Imaging*, 6(1):57–66, March 1987.
- [16] I. Bozma and J. Duncan. Integration of vision modules: A game-theoretic approach. In *Proc. Comp. Vision Pattern Recog. IEEE*, June 1991.
- [17] C. R. Brice and C. L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1:205–226, 1970.



- [18] D. J. Burr. A dynamic model for image registration. *Comp. Graphics Image Proc.*, 15:102–112, 1981.
- [19] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(6):679–698, November 1986.
- [20] R. Christensen. *Entropy Minimax Sourcebook*. Entropy Limited, Lincoln, Massachusetts, 1981.
- [21] S. M. Collins and D. J. Skorton. *Cardiac Imaging and Image Processing*. McGraw-Hill, New York, 1986.
- [22] D. B. Cooper. Maximum likelihood estimation of Markov-process blob boundaries in noisy images. *IEEE Trans. Pattern Anal. Machine Intell.*, 1(4):372–384, October 1979.
- [23] D. B. Cooper, H. Elliot, F. Cohen, L. Reiss, and P. Symosek. Stochastic boundary estimation and object recognition. In Azriel Rosenfeld, editor, *Image Modeling*, pages 63–94. Academic Press, New York, 1981.
- [24] A. de Moivre. On the law of normal probability. In D. E. Smith, editor, *A Source Book in Mathematics*. Dover Publications, New York, 1929.
- [25] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15(1):11–15, January 1972.
- [26] J. S. Duncan and L. H. Staib. Shape determination from incomplete and noisy multisensor imagery. In A. Kak and S. Chen, editors, *Spatial Reasoning and Multi-Sensor Fusion: Proceedings of the 1987 Workshop*, pages 334–344, Los Altos, CA, 1987. Morgan Kaufmann.

- [27] James S. Duncan and Thomas Birkhölzer. Edge reinforcement using parametrized relaxation labeling. In *Proc. Comp. Vision Pattern Recog.*, pages 19–27, June 1989.
- [28] P. J. Ell and B. L. Holman. *Computed Emission Tomography*. Oxford University Press, Oxford, 1982.
- [29] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comp.*, 22:67–92, January 1973.
- [30] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, Reading, Massachusetts, 1982.
- [31] H. Freeman. Computer processing of line drawing images. *ACM Comp. Surveys*, 6(1):57–98, March 1974.
- [32] J. M. Gauch and S. M. Pizer. Image description via the multiresolution intensity axis of symmetry. In *Proc. Second Int. Conf. Comp. Vision*, pages 269–274, 1988.
- [33] D. Geman and S. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.
- [34] J. J. Gerbrands. *Segmentation of Noisy Images*. PhD thesis, Delft University of Technology, June 1988.
- [35] C. R. Giardina and F. P. Kuhl. Accuracy of curve approximation by harmonically related vectors with elliptical loci. *Comp. Graphics Image Proc.*, 6:277–285, 1977.
- [36] D. B. Goldgof, H. Lee, and T. S. Huang. Motion analysis of nonrigid surfaces. In *Proc. Comp. Vision Pattern Recog.*, pages 375–380, June 1988.

- [37] G. H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Trans. Comp.*, 21:195–201, February 1972.
- [38] W. Eric L. Grimson. Recognition of object families using parameterized models. In *Proc. First Int. Conf. Comp. Vision*, pages 93–101, June 1987.
- [39] C. W. K. Gritton and E. A. Parrish, Jr. Boundary location from an initial plan: The bead chain algorithm. *IEEE Trans. Pattern Anal. Machine Intell.*, 5(1):8–13, January 1983.
- [40] Waldo S. Hinshaw and Arnold H. Lent. An introduction to NMR imaging: From the Bloch equation to the imaging equation. *Proc. IEEE*, 71(3):338–350, March 1983.
- [41] B. K. P. Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.
- [42] B. K. P. Horn and E. J. Weldon. Filtering closed curves. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(5):665–668, August 1986.
- [43] Berthold K. P. Horn. Extended Gaussian images. *Proc. IEEE*, 72(12):1671–1686, December 1984.
- [44] S. L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proc. Int. Joint Conf. Pattern Recognition*, pages 424–433, August 1974.
- [45] Robert Hummel and David Lowe. Computing large-kernel convolutions of images. Technical Report 254, Courant Institute of Mathematical Sciences, New York University, March 1987.
- [46] Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Anal. Machine Intell.*, 5(3):267–287, May 1983.

- [47] J. Illingworth and J. Kittler. A survey of the Hough transform. *Comp. Vision Graphics Image Proc.*, 44:87–116, 1988.
- [48] Gaetano Kanizsa. Subjective contours. *Scientific American*, 234:48–52, April 1976.
- [49] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4):321–331, 1988.
- [50] Kurt Konolige. A computer-based consultant for mineral exploration, Final report, Appendix D. Technical report, SRI International, Menlo Park, California, September 1979.
- [51] F. P. Kuhl and C. R. Giardina. Elliptic Fourier features of a closed contour. *Comp. Graphics Image Proc.*, 18:236–258, 1982.
- [52] Michael Leyton. A process-grammar for representing shape. In A. Kak and S. Chen, editors, *Spatial Reasoning and Multi-Sensor Fusion: Proceedings of the 1987 Workshop*, pages 148–157, Los Altos, CA, 1987. Morgan Kaufmann.
- [53] C. S. Lin and C. L. Hwang. New forms of shape invariants from elliptic Fourier descriptors. *Pattern Recognition*, 20(5):535–545, 1987.
- [54] Chun-Shin Lin. Invariants of three-dimensional contours. In *Proc. Comp. Vision Pattern Recog.*, pages 286–290, 1988.
- [55] David Lowe. Three dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–396, 1987.
- [56] David G. Lowe. Organization of smooth image curves at multiple scales. In *Proc. Second Int. Conf. Comp. Vision*, pages 558–567, December 1988.

- [57] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, 1984.
- [58] D. Marr. *Vision*. W. H. Freeman and Company, New York, 1982.
- [59] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London B*, 207:187–217, 1980.
- [60] A. Martelli. An application of heuristic search methods to edge and contour detection. *Comm. ACM*, 19(2):73–83, February 1976.
- [61] E. E. Milios. Shape matching using curvature processes. *Comp. Vision Graphics Image Proc.*, 47:203–226, 1989.
- [62] O. R. Mitchell and T. A. Grogan. Global and partial shape discrimination for computer vision. *Optical Engineering*, 23(5):484–491, 1984.
- [63] U. Montanari. On the optimal detection of curves in noisy pictures. *Comm. ACM*, 14(5):335–345, May 1971.
- [64] R. Nevatia and K. R. Babu. Linear feature extraction and description. *Comp. Graphics Image Proc.*, 13:257–269, 1980.
- [65] R. Nevatia and T. O. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98, 1977.
- [66] D. Nitzan, A. Brain, and R. Duda. The measurement and use of registered reflectance and range data in scene analysis. *Proc. IEEE*, 65(2), February 1977.
- [67] R. B. Ohlander, K. Price, and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Comp. Graphics Image Proc.*, 8:313–333, 1978.

- [68] R. L. Owen, L. H. Staib, P. Anandan, and J. S. Duncan. Measurement of left ventricular wall motion from contour shape deformation. In J. Llacer and D. Ortendahl, editors, *Information Proc. Med. Imaging*. Wiley-Liss, New York, 1989.
- [69] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1984.
- [70] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C. T. Chen. Accurate three-dimensional registration of CT, PET and MR images of the brain. *J. Comp. Assisted Tomogr.*, 13(1):20–26, 1989.
- [71] E. Persoon and K. S. Fu. Shape discrimination using Fourier descriptors. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(3):388–397, May 1986.
- [72] D. A. Pierre. *Optimization Theory with Applications*. Dover, New York, 1986.
- [73] Tomaso Poggio and Vincent Torre. Ill-posed problems and regularization analysis in early vision. Memo 773, MIT AI Laboratory, April 1984.
- [74] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7:155–162, July 1964.
- [75] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes*. Cambridge U. Press, Cambridge, 1986.
- [76] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 2. Academic Press, Orlando, Florida, second edition, 1982.

- [77] Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker. Scene labelling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):420–433, June 1976.
- [78] R. B. Schudy. Harmonic surfaces and parametric image operators: Their use in locating the moving endocardial surface from three-dimensional cardiac ultrasound data. Computer Science Tech. Rpt. 112, University of Rochester, Rochester, New York, March 1981.
- [79] C. E. Shannon. A mathematical theory of communication (concluded). *Bell System Technical Journal*, 27:623–656, October 1948.
- [80] J. Sklansky. On the Hough technique for curve detection. *IEEE Trans. Comp.*, 27(10):923–926, October 1978.
- [81] D. R. Smith. *Variational Methods in Optimization*. Prentice-Hall, Englewood Cliffs, 1974.
- [82] L. H. Staib and J. S. Duncan. Left ventricular analysis from cardiac images using deformable models. In *Proc. Computers in Cardiology*, pages 427–430. IEEE Comp. Soc., Los Alamitos, CA, 1988.
- [83] L. H. Staib and J. S. Duncan. Left ventricular analysis using parametrically deformable models. In *AAAI Spring Symposium Series: Artificial Intelligence in Medicine*, Stanford University, March 1988. (abstract).
- [84] L. H. Staib and J. S. Duncan. Parametrically deformable contour models. In *Proc. Comp. Vision Pattern Recog.*, pages 98–103, June 1989.

- [85] S. Tehrani, T. E. Weymouth, and G. B. J. Mancini. Knowledge guided left ventricular boundary detection. In *Proc. Comp. Vision Pattern Recog.*, pages 342–347, 1989.
- [86] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3D object reconstruction. *Int. J. Computer Vision*, 1:211–221, 1987.
- [87] G. P. Tolstov. *Fourier Series*. Prentice-Hall, Englewood Cliffs, 1962.
- [88] B. Widrow. The “rubber mask” technique - I and II. *Pattern Recognition*, 5:175–211, 1973.
- [89] T. Wilson and C. J. R. Sheppard. *Theory and Practice of Scanning Optical Microscopy*. Academic Press, London, 1984.
- [90] S. F. Witelson and D. L. Kigar. Anatomical development of the corpus callosum in humans: A review with reference to sex and cognition. In D. L. Molfese and S. J. Segalowitz, editors, *Developmental Implications of Brain Lateralization*. Guilford Press, New York, 1988.
- [91] Andrew P. Witkin and Jay M. Tenenbaum. What is perceptual organization for? In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 1023–1026, 1983.
- [92] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. In *Proc. Comp. Vision Pattern Recog.*, pages 104–109, 1989.