INFORMATION TO USERS

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

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

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

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

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

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

A Statistical Model for Fluorescence Image Cytometry

by

James Francis Lymp

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

1997

Approved by _____

(Chairperson of Supervisory Committee)

_____

_____

Program Authorized
to Offer Degree Department of Biostatistics

Date May 30, 1997

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, Michigan 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature James J. Zipp

Date May 30, 1997

University of Washington

Abstract

A Statistical Model for Fluorescence Image Cytometry

by James Francis Lymp

Chairperson of Supervisory Committee: Professor Finbarr O'Sullivan
Department of Biostatistics

Fluorescence image cytometry is a common laboratory method used to analyze tissue and culture specimens at the cellular level. Fluorescence imaging is useful because fluorescent stains are highly specific and imaging allows for direct spatial measurements. A statistical model was developed for analysis of fluorescence images. The model incorporates a nonparametric specification for the characteristic cell shape in the image. A realistic representation of the image data is achieved by flexible positioning, orientation and rescaling of each cell and summation of the contributions from a number of distinct cells. Advantages of this approach relative to current methods based on image segmentation include the improved ability to distinguish clustered structures and the ability to naturally incorporate blurring. The method of regularization is used to estimate the parameters of the model. Simulation studies show that the method is consistent for all parameters and gives reasonable estimates even in the presence of substantial image noise. The method is illustrated with data from a Tangier disease experiment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my dissertation advisor, Finbarr O'Sullivan, for his outstanding ideas, guidance and understanding. It has been a great experience, and I am looking forward to my postdoctoral fellowship under his supervision. Thanks to Brian Leroux and Michael LeBlanc for serving on my reading committee, and for their insightful ideas. I would like to thank Terry Kavanagh for introducing the problem in the first place, for serving on my committee, and for access to the laser cytometry equipment and data. I would like to thank Jack Oram for the opportunity to collect cell image data and to be involved with his research. Thanks also to Mary Richardson for serving on my committee and for being so helpful.

I would like to thank the National Institutes of Health for Grants HL-07183 and CA-57903. This funding made it much easier to complete my dissertation in a timely manner.

I would like to thank my fellow students, especially Robyn, Patrick, Juanjuan, and the members of the PET project: Adam, Ranjan and Kingshuk, for their help and for making my time in graduate school an enjoyable experience.

Thanks also to my family, especially Mom, Dad and Grandma Arlene, for always encouraging me to do my best and to pursue what I enjoy. And an extra special thanks to Robyn for putting up with me, encouraging me, and for reading the dissertation - your turn!

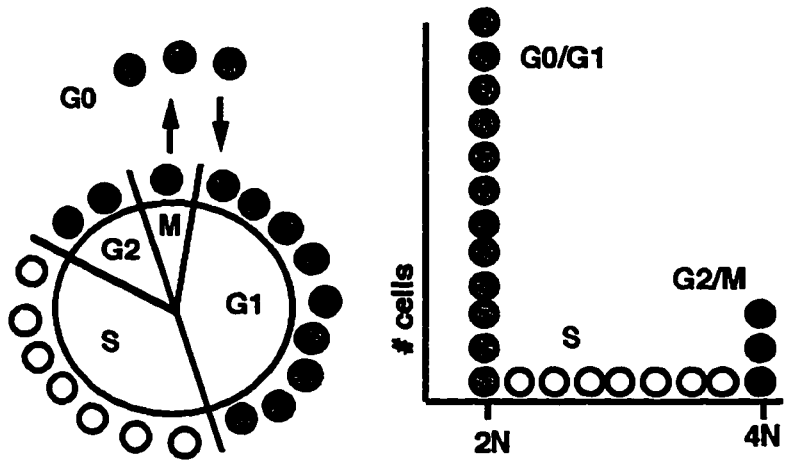
Chapter 1

INTRODUCTION

In this chapter, cytometry and quantitative image analysis are introduced. There are two main categories of techniques used in cytometry: flow cytometry and image analysis. In the description of cytometry, some applications are given and the two main categories are described and compared. Quantitative image analysis is a growing field with many applications and techniques. Some of the techniques used in fluorescence image cytometry are detailed, and a brief overview of image analysis in general is given. Finally, the contents of this dissertation are outlined.

1.1 Cytometry

Cytometry is the measurement and characterization of cells and cellular components. The word *cytometry* comes from the Greek roots *cyto* (cell) and *metry* (measurement). Sometimes cytometry is referred to as analytical cytology. The two main applications of cytometry are DNA analysis and immunophenotyping, especially for evaluating malignancy [37, 46, 80, 87]. Cytometry is also used for many other things including the analysis of cell anatomy and physiology, cell counting, classification and localization. The main cytometry methods are flow cytometry and image cytometry. In the future, the role of cytometry will be to measure cell and tissue features beyond human perception, to make very detailed measurements and classification to assist clinicians in prognosis, diagnosis and screening [87] and to assist cytologists in cell biology research.



(a) Cell cycle. (b) DNA content.

Figure 1.1: Dependence of DNA content on the cell replication and division cycle.

The main application of cytometry is DNA analysis and especially the evaluation of DNA content, the number of replicates of chromosomes in the cell. The normal range for DNA content is from 2N to 4N, where N=23 chromosomes for humans. The amount of DNA in a cell depends on cell cycle stage [26, 37, 80, 87] as illustrated in Figure 1.1. Figure 1.1(a) illustrates the cell replication and division cycle. Cells that are not involved in the replication and division cycle are considered to be in G0 phase and have the standard content of 2N, or 46 chromosomes per cell. Cells that are preparing for replication are in G1 phase and also have DNA content of 2N. As cells undergo replication, they are in S phase and their DNA content ranges from 2N to 4N. When the cells have finished replicating and before starting the division cycle, they are in G2 phase and have content of 4N. When the cells are undergoing mitotic division, they are in M phase with DNA content of 4N until the point when the cells actually separate. Then the cells either go out of the cycle into G0 phase or prepare for replication in G1 phase [26, 37, 80, 87]. Figure 1.1(b) illustrates a typical ideal

histogram of DNA content that would coincide with cells in various stages of the cell cycle. Many of the cells are at 2N and there is also a group at 4N. The rest are spread between 2N to 4N. Deviations from a standard histogram might be an indicator of malignancy [37, 80, 87].

Some other major applications of cytometry are immunophenotyping, anatomy and physiology, cell counting, classification and localization. Immunophenotyping is the identification and classification of cells by analysis of cell surface proteins characteristic of that cell type. This is useful for evaluating the immune system status to monitor HIV/AIDS patients and organ transplant patients. Immunophenotyping is also useful in evaluating tumor malignancy and the effects of drugs on the body [37, 80, 87]. The analysis of cell anatomy and physiology is useful for, among other things, the analysis of cell size and shape [12, 49], and to study apoptosis, a mode of cell death [27, 109]. Cell counting is useful, for example, to study cell culture growth rate. Classification of cells could be based on many things, such as size, shape, or DNA content and is useful, for example, to determine cell type or to study cell abnormalities. Localization is useful for studying cell sociology [14], or how cells interact spatially, and for subcellular localization, to study the spatial aspects of cellular components.

1.1.1 Flow Cytometry

Historically most common, flow cytometry is probably still the most familiar cytometry method. Flow cytometers are widely available and usually come with a cell sorter. The method uses a large number of cells and is fairly automated, so it is very common to see quantitative analyses in conjunction with flow cytometry.

Figure 1.2 is a diagram of a typical flow cytometer [26, 37, 80]. In flow cytometry, many cells are suspended in a fluid. This cell suspension flows in a fine stream past a light source and a group of detectors. The stream is narrow enough (50-150 μm) so that only one cell at a time passes the detectors. The stream flows very fast (about

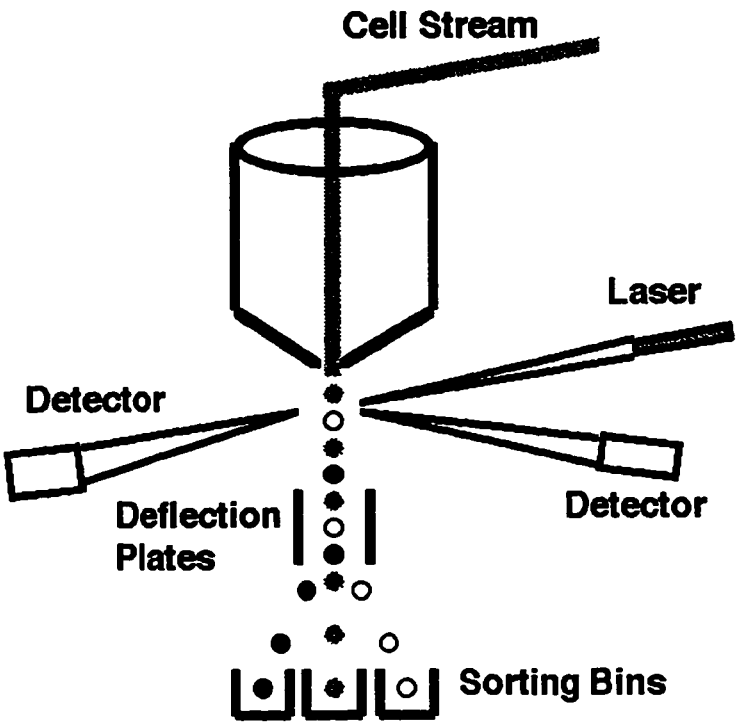


Figure 1.2: Flow cytometer.

10 m/sec) so that 10,000 cells can be analyzed in about 10 seconds. As the light source illuminates the stream, some of the photons pass through, some are deflected, and some are absorbed by particles in the stream. The detectors receive light and then separate colors with a series of mirrors and filters. The position of the detector, the colors detected and the intensity of the light are measured, and analysis is based on these data.

A feature that usually comes with a flow cytometer is a cell sorter. After the cells flow past the light source and detectors, they flow past a pair of electrically charged plates. In the amount of time it takes a cell to flow from the detector to the plates, the information from the detectors is processed, a sorting decision is made, and the cell is either allowed to fall straight into one bin or deflected by the plates into another. The newly sorted cells can then be cultured and analyzed further.

The data from flow cytometry are in the form of intensity histograms, with the horizontal axis of the histogram representing a parameter of interest [37, 80]. The histograms might be analyzed, for example, by comparing the centers of two histograms using standard statistical tests. Frequently, two parameters are analyzed simultaneously. This is done with two-dimensional histograms, or contour plots, called cytograms. Cytograms are plotted with a different parameter on each axis. The parameters could be features like different colors, different detectors or different cell samples. Statistical techniques used include multivariate analyses such as principal components, factor analysis and cluster analysis.

In many cases, the data analyzed in flow cytometry are in the form of DNA content histograms. It might be of interest to compare a test histogram to a control histogram using, for example, the Kolmogorov-Smirnov total probability test. More difficult is estimation of the distribution of cells in S phase. The difficulty arises because the S phase distribution is overshadowed by the cells in other phases of the cell cycle. Some form of mixture distributions might be used here. Another difficult problem is the evaluation of tetraploid abnormality. This is difficult because it is hard to distinguish true tetraploid abnormality from clumping cells or random variations in the cell cycle.

1.1.2 Image Cytometry

The other main cytometry method is image cytometry - cell measurement by analysis of microscopic images [46, 53, 87, 99, 106]. The difference in data between image cytometry and flow cytometry is mainly a matter of dimension. In flow cytometry the data are in the form of histograms. In image cytometry the data are pictures, so in addition to intensity there is also direct spatial data for analysis of characteristics such as size, shape and location. There are several advantages of image cytometry relative to flow. The method is less intrusive. That is, the tissue architecture remains intact and fewer cells are needed; more intensity parameters can be measured;

morphology measurements are possible; repeated measures can be taken on the same tissue section over time; and spatial aspects such as particle location and socialization can be directly observed and measured [45].

Image cytometry is widely available, yet despite all of its advantages and potential, it has not been used as much as flow cytometry. This is mainly because image cytometry has historically been difficult to quantitate. In fact, analysis of image cytometry data is often done by merely observing the resulting picture. Quantitation has been difficult due to the complexity of images, the presence of 3D blur in optically sectioned images, and the slow speed of algorithms used to analyze the images. Image complexity is tackled by the use of fluorescent stains. 3D blur is remedied by improving deconvolution methods [21, 55] and confocal microscopy. Fluorescent stains and confocal microscopy are discussed shortly. The use of quantitation in image cytometry has been increasing because algorithms for image analysis have been improving. Researchers in many fields are working on improved algorithms for image cytometry, and one of these algorithms is the topic of this dissertation.

The use of fluorescent stains helps quantitation in image cytometry. Fluorescent stains are chemical substances [37, 53, 80, 99, 106] that associate with structures or macromolecules of interest in cells, absorb light of a specific color, and emit light of a different color. They are useful because they are highly sensitive and specific, so their light emissions can be analyzed to make inferences about particles of interest. Fluorescent stains are especially helpful in image cytometry by simplifying an image into one where objects of interest have higher fluorescence intensity than the background. This makes the image well-suited for statistical optimization because pixel intensity is correlated with the likelihood of being in an object.

A confocal laser scanning fluorescence microscope is illustrated in Figure 1.3. Laser light of a specific color is focused by a lens onto a dichroic mirror. A dichroic mirror reflects light of certain colors and transmits light of other colors. The laser light is reflected by the dichroic mirror and focused by an objective lens onto a fine

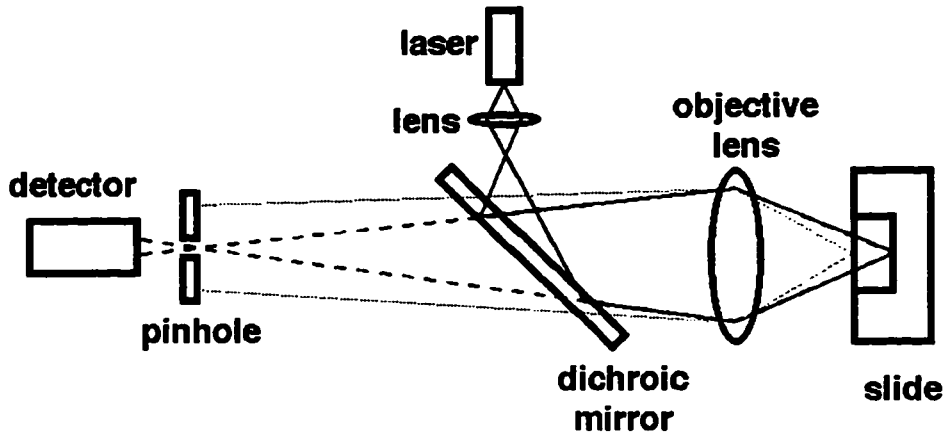


Figure 1.3: Confocal laser scanning fluorescence microscope.

point of a thin focal plane of a slide specimen. The illumination point can be as small as about 0.25 microns in diameter and the focal plane as thin as $0.5 - 1.5$ microns [107]. The fluorescent particles on the slide absorb the laser light then emit a different colored light. This light color passes through the dichroic mirror and is focused by the objective lens through a small pinhole before reaching the detector. Fluorescent emissions that are not in the focal plane are out of focus at the pinhole and only a small portion pass through to the detector. Therefore, photons from out of focus planes have minimal influence on the image formed from the detector. This microscope is called confocal because the laser and the detector are focused on the same point of the slide at the same time. The parts of the microscope move so that the laser scans through the slide specimen to produce an image like the one shown in Figure 1.4. The focal plane can be changed, and the slide scanned again, to produce a series of such images. Thus a 3D image of the slide specimen can be reconstructed and analyzed.

Confocal laser scanning microscopy is useful to reduce 3D blur in fluorescence imaging. 3D blur is the detection of artifacts not in the focal plane and occurs because a microscopic image is a thin 2D optical slice of a relatively thick 3D specimen.

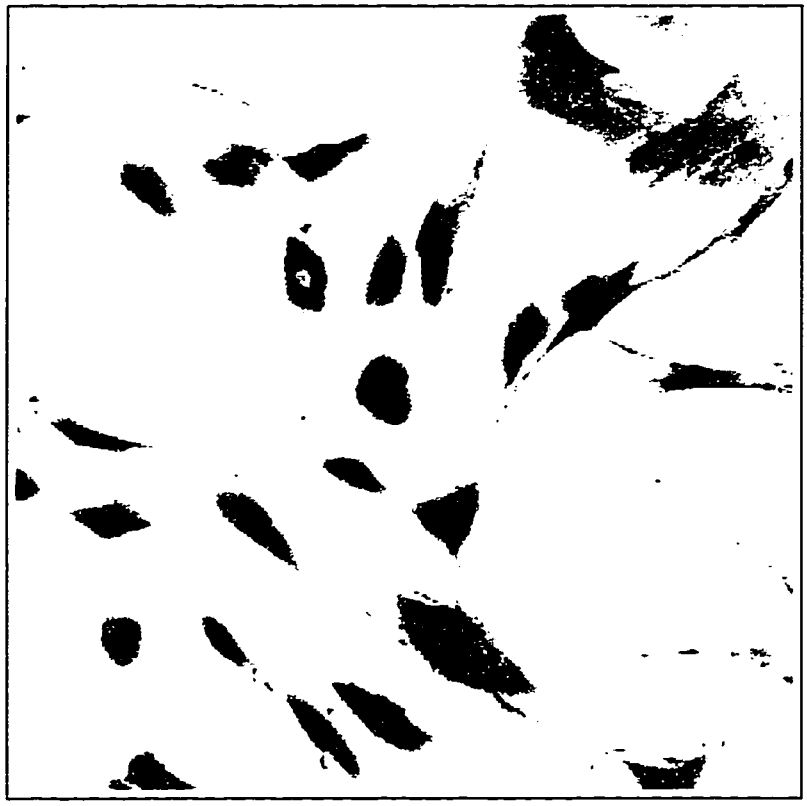


Figure 1.4: Example cell image obtained from confocal laser scanning microscopy. The cells shown are human fibroblast cells from an apolipoprotein study by Dr. John Oram, Department of Medicine, Metabolism and Endocrinology, University of Washington. The confocal microscope is part of a Meridian ACAS Ultima Laser Cytometer in the Environmental Toxicology Labs, University of Washington. The image is 270x270 pixels where each pixel is 2 μm^2.

A confocal laser scanning fluorescence microscope [62, 93] produces images with relatively little 3D blur and is the type of microscope used to obtain the cytological images presented and analyzed in this paper. The imaging system used is a Meridian Instruments (Okemos, MI) laser cytometer located in the Environmental Toxicology Laboratories, University of Washington. Microscopy methods are constantly being improved. One technique that promises to produce even better images is the use of standing waves [8].

When fluorescent stains are used, this process is often called fluorescence image

cytometry. When fluorescent stains are imaged using confocal laser scanning microscopy, it is often referred to as laser cytometry. The methods developed in this dissertation are mainly intended for images of cells obtained with a confocal laser scanning fluorescence microscope, but might also be useful in other image cytometry settings. The term "fluorescence image cytometry" is thus used in an attempt to most accurately capture the field of application for the methods.

1.2 Quantitative Image Analysis

One of the major challenges for quantitative image cytometry is efficient and accurate image analysis [53, 54, 55]. The main roadblock to achieving this is image segmentation, the decomposition of an image into meaningful subimages or regions. The three basic segmentation methods currently used are thresholding, edge detection and uniform region detection [51, 97, 110]. There are many features in cytometry images that are not accounted for in these methods and cause difficulty. Some methods currently exist for incorporating some of these features. Image analysis is a growing field with a wide range of methods and applications.

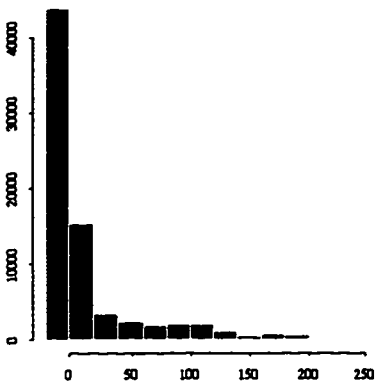
1.2.1 Image Segmentation Methods

Image segmentation is the division of an image into meaningful subimages, or regions. For example, in image cytometry, there might be one region for every cell, plus one for background. There are many ways to accomplish this, most of which use at least one of the three basic methods - thresholding, edge detection, and uniform region detection. None of these methods was designed for fluorescence image cytometry, so it is not surprising that analysis based on these methods will have some shortcomings. Included in the following description of each basic segmentation method is a statement of the difficulty with applying the method to fluorescence image cytometry. All of the basic methods have difficulty distinguishing clustered and overlapping structures.

One segmentation method is thresholding. There are several ways to threshold, but most methods are based on global intensity. Figure 1.5 illustrates one very common method. A histogram was made of the intensities from the image shown in Figure 1.4. There are 2 regions (cell vs. background), so a "cut off" or threshold was chosen. The threshold 75 was chosen because it is the approximate location of a dip in the histogram. More sophisticated methods could also have been used. In this example all of the pixels with intensity less than 75 were set to zero and the remaining pixels set to 1. What remains is a binary image with value 1 for cell and 0 for background. Thresholding is also commonly done without histograms. For example, a person can choose arbitrary levels until finding the threshold level that satisfies the eye. For thresholding to work well, there needs to be a sharp contrast in an image characteristic corresponding to different regions of interest. Threshold methods have difficulties in the presence of variations in particle brightness, size, shape, and orientation, as well as image noise and non-uniformities.

Edge detection involves finding areas in the image with high change in intensity. This can be done with an algorithm where "high intensity change" is clearly defined. Usually, the gradient or a similar operator is used. Figure 1.6(a) illustrates the application of a gradient operator to the image in Figure 1.4. A subsequent algorithm would then be needed to select edge pixels based on this image. Figure 1.6(b) illustrates the result of using the *contour* function in S-PLUS (Statistical Sciences, Inc., Seattle, WA). Edge detection, in some sense, can also be done manually by drawing on a computer screen using a pointer or a mouse. Edge detection algorithms need closed and sharp boundaries between regions. These methods have difficulties when boundaries are degraded by image noise and non-uniformities.

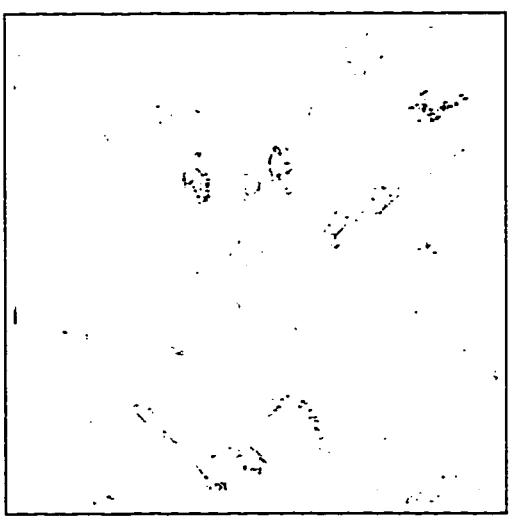
Uniform region detection involves finding areas in an image with homogeneous intensity. This method is fairly flexible in that certain constraints can be easily incorporated, such as number and size of regions. A commonly used uniform region detection method is the split and merge algorithm [17]. The first step is splitting,



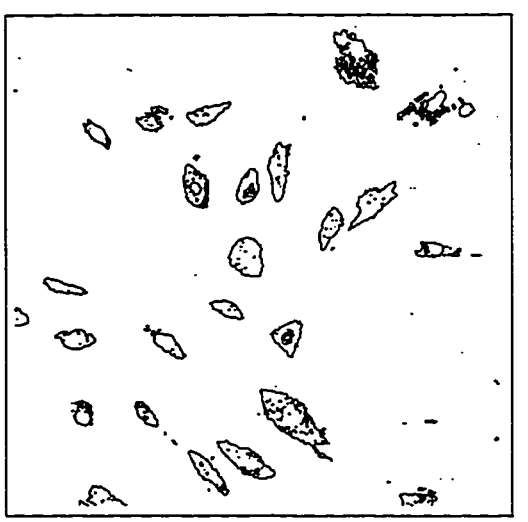
(a) Histogram of pixel intensities from the image in Figure 1.4.

(b) Thresholded image. The dark pixels had intensity greater than 75 in the original image.

Figure 1.5: Thresholding.



(a) Gradient image.



(b) Contour of image.

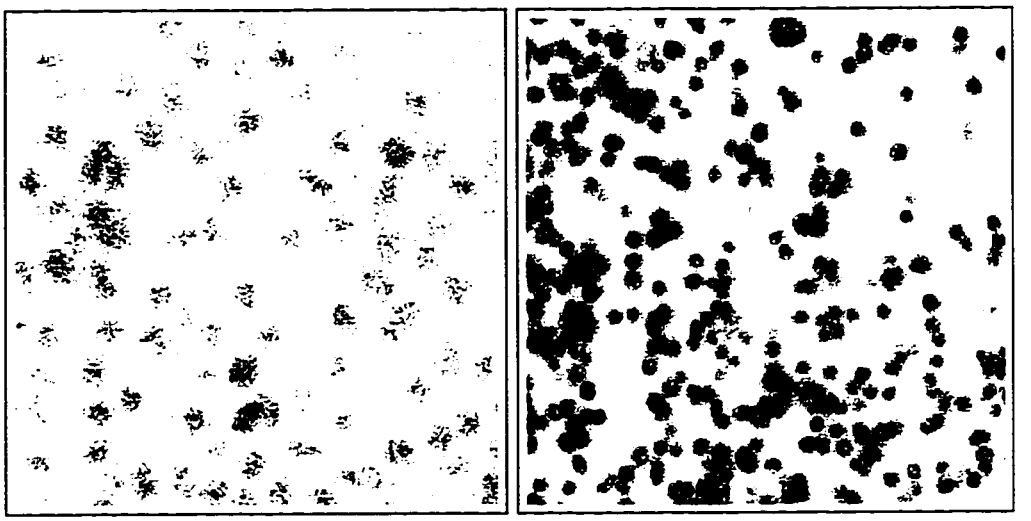
Figure 1.6: Edge detection.

(a) 25 uniform regions. (b) Predicted image.

Figure 1.7: Uniform region detection. An illustration of the split and merge algorithm.

where the image gets split into smaller and smaller homogeneous rectangles. When the image has been divided enough, the merge step begins, where adjacent homogeneous rectangles are merged to make more complex regions. The result of a split and merge algorithm performed on the image in Figure 1.4 is illustrated in Figure 1.7. Uniform region detection methods need homogeneous regions, and have difficulties with image non-uniformities.

The three basic image segmentation methods do fairly well on many types of images [54]. However, these algorithms were not designed for fluorescence image cytometry, where cells can be clustered, and may have trouble correctly identifying the cells. The main problem is that segmentation does not incorporate all of the features in cytometry images. Some of these features are apparent in the images shown in Figures 1.4 and 1.8. The objects in Figure 1.4 have quite complex but fairly similar shapes, with many of them touching each other. In Figure 1.8(a), the image is very noisy, and the objects are very similar in size and shape, with a few touching



(a) Fibroblasts stained with monochlorobimane (MCB) and treated with chlorodinitrobenzene.

(b) Fibroblasts stained with BM7 and untreated.

Figure 1.8: More example images with features to incorporate in modeling. Both images are of human diploid fibroblasts stained for thiol content. Each image was obtained using a Meridian ACAS Ultima Laser Cytometer in the Environmental Toxicology Labs, University of Washington. Each image is 180x180 pixels where each pixel is 1 μm^2.

each other. The objects in Figure 1.8(b) are highly clustered, with similar size and shape, and highly variable fluorescence intensity. An ideal image analysis algorithm would incorporate features such as these. All of the basic methods have difficulty with these features. The next section reviews what has been done so far to improve the performance of these methods.

1.2.2 Extending Segmentation

There has been much work done on variations of the basic segmentation methods to improve their performance in image cytometry. Some of the methods that have been used include human interaction, combining methods, image filtering, cluster analysis, and mathematical morphology.

Human interaction is used often to improve segmentation results in image cytometry. Some common uses of interaction are to initialize the segmentation process [20], to outline the edges of cells [82], to choose threshold level [11, 64], and to exclude [64] or separate objects. The main disadvantages of using human interaction in segmentation are that user bias and error is introduced, the results are not reproducible, and it is very time consuming so that it is not feasible to analyze a large number of images.

Combining methods can help efficiency by using the advantages of more than one method. This can be done either sequentially or simultaneously. For example, thresholding might precede one of the other two methods, or edge detection might be used as a starting point for uniform region detection [110]. An example of simultaneous use would be using the gradient image (a type of edge detector) to guide the placement of the threshold [79].

Image filtering can be thought of as the process of applying a local operator to the image. In image cytometry this is usually done with convolution or deconvolution [1]. Convolution is the process of applying a local averaging kernel or point spread function to every point on the image. The effect of this is to cause the image to appear smoothed, or blurred. Deconvolution is the reverse process and causes object edges in the image to appear sharper and more distinct. Image filtering is normally used before applying other segmentation methods [44, 54, 77].

Cluster analysis, also called clustering, is a statistical technique that is used to group data into homogeneous groups, or clusters [59]. In image cytometry, cluster analysis is used to separate a group of objects, like cells or chromosomes, that has been segmented as one [41, 54, 84]. The main requirements for a clustering algorithm are criteria to detect potential clusters and a method of separating clustered objects. Commonly, the cluster criteria are object size and shape factor (usually the ratio of squared perimeter to area) and division is done by finding a low intensity path. Clustering is also used in 3D image analysis [4].

Mathematical morphology is the application of lattice theory to spatial structures [75]. Image filters and segmentation algorithms are two applications frequently used in image cytometry. Operators are built from set theory operations (union, intersection, difference, and complementation) and two basic transformations called erosion and dilation. Erosion and dilation are best described with set theory notation [75]. Basically, erosion causes objects in the image to become smaller (and dilation causes objects to become larger) in a well-defined manner by using a structuring element, which defines a neighborhood about a point in the image and controls the size and local shape of the transformation. More complex operators are built from erosion and dilation with set theory operations, for example an operator denoted as opening is erosion then dilation and another denoted as closing is dilation then erosion. These operators are also generalizable to gray-scale images, gray-scale structuring elements, and continuous (rather than pixelized) images and/or structuring elements. Thus the operators are examples of useful image filters. Mathematical morphology is used among other things to filter images by separating touching objects, smoothing edges, filling holes in objects, and eliminating small objects. There are many examples of the application of mathematical morphology to cytological images, see for example, Barba et al. [10], Dow et al. [30], Kim et al. [44], Lee et al. [47], and Moragas et al. [64].

1.2.3 Statistical Methods for Image Analysis

Statistical optimization can be used in many aspects of image analysis. The use of these methods in image cytometry has been limited, but is growing. This is largely because of increasing feasibility due to improvements in computer computation speed and numerical optimization techniques. The method described in this dissertation applies to fluorescence image cytometry general optimization methods normally used in other contexts.

Statistical optimization techniques for image analysis have thus far not been

widely taken advantage of in image cytometry. Examples of statistical optimization in use are geometric constraint networks [20], maximum likelihood estimation [54, 84], and template matching [100]. Additionally, the method of regularization has been used in the contexts of edge detection [30] and uniform region detection [83, 84], and probability models have been used to identify intensity peaks [29].

The majority of statistical techniques for image analysis are described under the so-called Bayesian paradigm. Basically, *a priori* probability models are given for various aspects of the image and the feature of interest is estimated by maximizing an appropriate *a posteriori* distribution. The works of Geman and Geman [36] and Besag [13] are often noted as seminal works in statistical image analysis. They did this by introducing to image analysis the ideas of stochastic relaxation, simulated annealing and iterative conditional modes. These methods made it possible to use various statistical models for image restoration. These ideas, along with deformable models for object shape, have been used in many image analysis settings, including object recognition [3, 7, 35, 60] and boundary detection [23, 78].

Another statistical technique that has been used in image analysis is the method of regularization, mainly for reconstruction of images in Positron Emission Tomography (PET) [74]. Work has also been done in the area of shape analysis of landmark data [16, 58, 111] and cluster analysis of curves in images [9]. The method described in this dissertation is a method of regularization procedure incorporating features similar to those used in object recognition and boundary detection.

1.3 Dissertation Outline

In Chapter 1, cytometry was introduced. Most of the attention was given to fluorescence image cytometry, the main field of application for the methods developed in this dissertation. Quantitative image analysis was discussed in the contexts of fluorescence image cytometry and developments in the statistics field. Limitations of

current methods were discussed along with features seen in cytometry images that bring about these limitations.

The proposed statistical model for fluorescence image cytometry is motivated and formulated in Chapter 2. The model incorporates a nonparametric characteristic shape for the cells, various parameters that are allowed to differ among the cells, and a model describing general image features. Results of a fluorescent bead study are given that show the image noise acts as expected. A computer program for generating images under the model is described along with some examples showing the types of features it incorporates.

Chapter 3 focuses on parameter estimation under the cell image model described in Chapter 2. The characteristic shape and the cell parameters are estimated separately. A modified Gauss-Newton procedure is developed for each part of the algorithm. For the characteristic shape function, regularization is also used. The necessary calculations are derived in detail.

The estimation procedure is evaluated in Chapter 4. Simulation results show that estimation of the true image, the characteristic shape function, the cell parameters, and certain functionals of interest appear consistent. That is, the estimation error gets smaller as image noise is reduced. Estimation of the characteristic shape function and average functional values is found to improve as the number of cells increases as well. The rate of estimation appears as might be expected for this type of model. Estimation is quite good even in the presence of substantial noise. Average functional estimates are biased when the image noise is extremely high. Additionally, a few ideas for theoretical analysis are given.

In Chapter 5, the method is illustrated with data from a Tangier disease morphology experiment. A background on Tangier disease is given to motivate the analysis. The results are compared with those obtained using a standard method. The method developed in this dissertation provides more information with less user input than the standard method. The statistical model developed in this dissertation is thus a

useful tool for fluorescence image cytometry.

Finally, Chapter 6 includes a summary of the main results and pathways for future work.

Chapter 2

A CELL IMAGE MODEL

In this chapter, a statistical model for fluorescence image cytometry is described, where fluorescence emission from markers on several cells appear in an image region. The image region is assumed to be two-dimensional and is usually rectangular. Each cell is modeled as having a characteristic shape which falls into the starshaped class of objects. This shape is then adjusted by cell specific parameters including size, aspect ratio, orientation, location, and intensity. The contribution from each cell is summed to form an image. The image is then blurred with a Gaussian kernel and noise is added. A program was written to simulate images under this model. Images are shown revealing features and limitations of the cell model. Error characteristics of a confocal laser scanning fluorescence microscope are discussed. Fluorescent beads were scanned 50 times and the resulting images used to study the statistical aspects of noise in fluorescence microscopy.

2.1 An Overview of the Cell Image Model

The cell image model consists of three components. One component is a characteristic cell shape assumed to be the same for all cells in the image. Another component is cell specific parameters which vary among the cells. The third component is an image representation.

The characteristic shape is assumed to fall in the class of starshaped objects. This means that there is a point in the cell that can "see" the entire edge, i.e. a line connecting that point with any edge point is entirely contained within the cell.

The class of starshaped objects is more general than the convex class, where every point in the object can "see" every other point; however, it is a special case of the simple class, where the boundary is closed but never crosses over itself. The starshape constraint has been used in other image analysis settings [35, 60, 78, 85, 86]. The starshaped class is chosen because it captures a wide variety of objects while allowing the shape to be written as function of one variable on a unit circle. Two examples are shown in Figure 2.1. The shape is assumed to be the same for every cell, while only certain parameters are allowed to vary among the cells. This reduces computation considerably by allowing estimation of a single shape function for each image.

The cell specific parameters are size, aspect ratio, orientation, location and intensity. An illustration is given in Figure 2.2. These parameters are allowed to be different for each cell. The overall size of the cell is represented by one parameter called size. The aspect ratio parameter reflects the difference between round and oblong objects. This parameter is sometimes called shape in other contexts. Here the term shape is restricted to be the edge information remaining after accounting for all the cell specific parameters. Orientation and location reflect how the cell is turned and placed in the image. Intensity is assumed to be constant within each cell, and reflects the amount of emitted fluorescence. Location, size, aspect ratio and orientation define a unique elliptical boundary for each cell. Additionally, the characteristic shape function describes the deviation from these ellipses. With the cell specific parameters combined with the characteristic shape function, quantities of interest in cytometry are readily calculated.

These parameters may be different for each cell in the image, but they will be correlated at least somewhat due to the nature of the cells. One way to incorporate these correlations is to use random effects specifications for these parameters. This essentially means that there are average values for these parameters in the cell population, and the parameters for each cell vary around this average in some manner. This might be done using either a nonlinear or a generalized linear model with

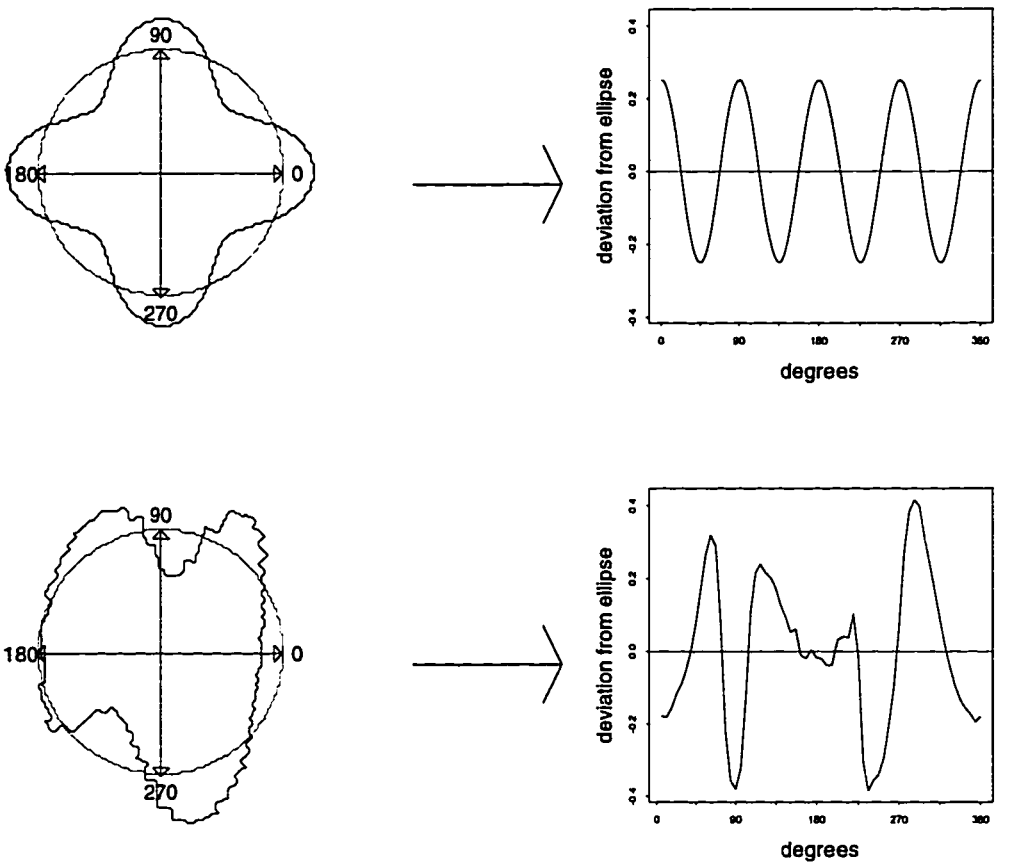
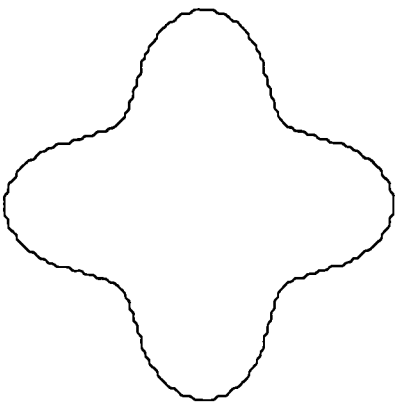
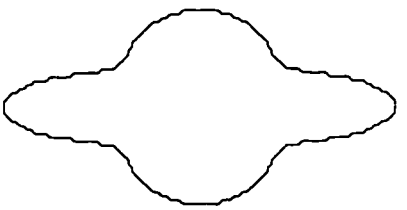


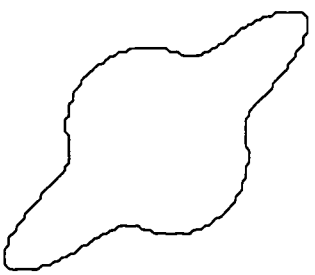
Figure 2.1: The starshape model. A point in the object can "see" the entire edge, allowing the shape to represented as a function of one variable. Two different shapes are shown with the ellipse (constant shape function) added for reference. The second curve is used for the simulation studies in Chapter 4.



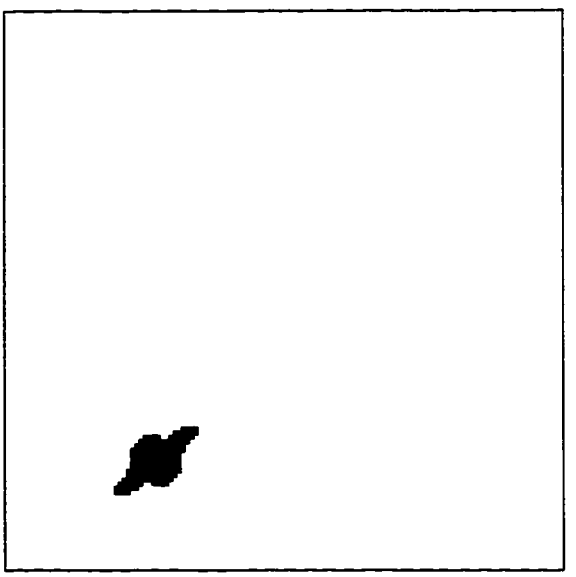
(a) Size



(b) Aspect ratio

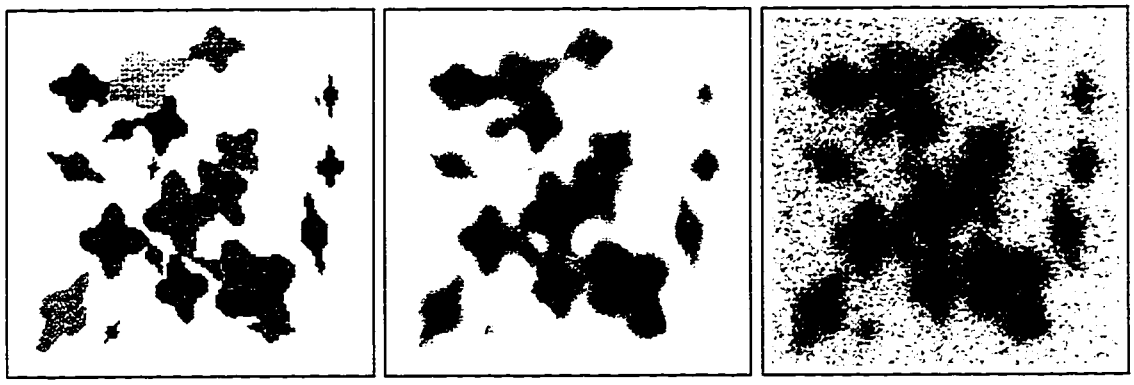


(c) Orientation



(d) Location and intensity

Figure 2.2: Cell specific parameters.



(a) Sum distinct cells (b) Blurring (c) Noise

Figure 2.3: Image representation. Note: for illustrative purposes, cell overlap, blurring and noise shown are all more extreme that what normally occurs in confocal laser scanning fluorescence microscopy of cell monolayers.

random effects specifications [52, 19].

The observed image representation is illustrated in Figure 2.3. The fluorescent intensity contribution from each distinct cell is summed to form an image. A small amount of blurring occurs causing sharp edges to appear fuzzy. This is modeled using a bivariate Gaussian blurring kernel. There is also a small amount of noise involved in any imaging system. A well-known example of this is the picture seen from a weak television signal. Blurring and noise are discussed in detail later in the chapter.

2.2 Model Formulation

To formulate the cell image model, assume the image region is two-dimensional and divided into squares called pixels. Let m be the number of pixels and n be the number of cells. Let $x = x_k, k = 1, \ldots, m$, be the coordinates for the center of the kth pixel in the image region. Then the observed intensity z at x is represented as

$$z(x) = \sum_{i=1}^{n} f_i(x|h, \theta, \psi_i) + \epsilon(x),$$

where $\epsilon(x)$ is independent noise. The function f_i is the intensity contribution from the ith cell to the point x and depends on the characteristic shape function θ, the blurring parameter h, and the cell specific parameters ψ_i. The cell specific parameters include size α_i, aspect ratio β_i, orientation ϕ_i, location μ_i and intensity γ_i, so that $\psi_i = (\alpha_i, \beta_i, \phi_i, \mu_i, \gamma_i)$. The cell function f_i is the cell specific intensity γ_i multiplied by a convolution of an indicator for the cell region $R_i(\theta, \psi_i)$ and a Gaussian kernel g_h with known scale parameter h:

$$f_i(x|h, \theta, \psi_i) = \gamma_i (I_{R_i(\theta,\psi_i)} * g_h)(x),$$

where $I_{R_i(\theta,\psi_i)}(y)$ is 1 if y is in the ith cell region and zero otherwise, and $g_h(x) = \exp(-\|x\|^2/2h^2)/(2\pi h^2)$, the bivariate Gaussian density function with zero means, zero correlation and scale parameters both h. This can be written as an integral over the cell region:

$$f_i(x|h, \theta, \psi_i) = \gamma_i \int I_{R_i(\theta,\psi_i)}(y) g_h(x-y)dy = \gamma_i \int_{R_i(\theta,\psi_i)} g_h(x-y)dy.$$

In order to estimate the characteristic shape and the cell specific parameters, the cell image model needs to be written as an explicit function of the parameters. First, the cell function needs to be reparameterized in terms of coordinates relative to the cell. Let $u_i = A_i^{-1}(y - \mu_i)$, where y is the location in image coordinates and u_i is the location in cell coordinates. The parameter μ_i is the location of the center of the cell and A_i is the transformation matrix for the cell which depends on its size, aspect ratio, and orientation:

$$A_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \alpha_i \begin{bmatrix} 1 & 0 \\ 0 & \beta_i \end{bmatrix}.$$

The size parameter α_i is the length of the longest semiaxis and the aspect ratio β_i is the ratio of the perpendicular and the long axes. Together the parameters μ, α and β define an ellipse with equation

$$\left(\frac{x_1 - \mu_1}{\alpha}\right)^2 + \left(\frac{x_2 - \mu_2}{\alpha\beta}\right)^2 = 1,$$

and the parameter ϕ defines an orthogonal transformation through the matrix

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

The Jacobian for the transformation is $|dy/du_i| = |A_i| = \alpha_i^2\beta_i$. The cell function can now be rewritten in cell coordinates, and as an explicit function of the cell specific parameters, as

$$f_i(x|h,\theta,\psi_i) = \gamma_i\alpha_i^2\beta_i \int_{R(\theta)} g_h(x - \mu_i - A_iu_i)du_i,$$

where $R(\theta)$ is the cell region in cell coordinates, now depending only on θ.

The cell function is transformed into polar coordinates to obtain an explicit function of characteristic shape. Let r be the length of the cell coordinate vector u_i: $r = \|u_i\| = \sqrt{u_{i,1}^2 + u_{i,2}^2}$, and let t be the angle of the cell coordinate vector u_i: $t = \arctan(u_{i,2}/u_{i,1}) + \pi I\{u_{i,1} < 0\} + 2\pi I\{u_{i,1} \geq 0, u_{i,2} < 0\}$, where $I\{x\}$ is an indicator function equal to 1 if x is true and 0 otherwise. Then the cell function can be rewritten in polar coordinates as

$$f_i(x|h,\theta,\psi_i) = \gamma_i\alpha_i^2\beta_i \int_0^{2\pi} \int_0^{\theta(t)} g_h\left(x - \mu_i - B_i(t)r\right) r\,dr\,dt,$$

where $B_i(t) = A_i(\cos t, \sin t)^T$.

The characteristic shape function θ is a function of angle t and maps the interval $[0, 2\pi)$ to the nonnegative real line. It is possible to write the shape function this way because the cell shape is in the starshaped class. The characteristic shape of an ellipse is constant on $[0, 2\pi)$, i.e. $\theta(s) = \theta(t)$ for all $s, t \in [0, 2\pi)$. Note also that the shape function θ represents the object boundary information left over after removing the effects of location, orientation, size, and aspect ratio.

2.3 Quantities of Interest

For inference, various functionals of the shape function and cell specific parameters are studied. Some quantities of interest in cytometry are area, perimeter, shape factor,

average intensity and integrated intensity. These quantities are readily calculated as functions of the shape curve and the cell specific parameters. Shape factor and integrated intensity are functions of area, perimeter and average intensity as follows:

$$\text{shape factor} = \frac{(\text{perimeter})^2}{4\pi \text{area}},$$

$$\text{integrated intensity} = \text{area} \times \text{average intensity}.$$

The shape factor is equal to 1 for a circle and increases with shape complexity. Average intensity is the same as the intensity parameter γ. Area and perimeter are functionals of the shape curve θ, size α and aspect ratio β.

The formula for the area of a cell with shape function θ, size α and aspect ratio β can be approximated by partitioning into sectors and approximating the area of each sector by the area of a portion of an ellipse:

$$\text{area} \approx \sum_{i=1}^{n} \pi \alpha^2 \beta \theta(t_i)^2 \frac{\Delta t_i}{2\pi},$$

where $0 < t_1 < \cdots < t_n < 2\pi \equiv t_{n+1}$ and $\Delta t_i \equiv t_{i+1} - t_i$ is the same value for all i. Taking the limit of this sum, the formula for area is

$$\text{area} = \frac{\alpha^2 \beta}{2} \int \theta(t)^2 dt.$$

To derive the formula for the perimeter of a cell with shape function θ, size α and aspect ratio β, start with the general formula for perimeter in rectangular coordinates from calculus [31]. This is given by

$$\int \sqrt{dx^2 + dy^2}.$$

Then, letting $x = \alpha\theta(t)\cos t$ and $y = \alpha\beta\theta(t)\sin t$, the derivatives are

$$\begin{aligned} \frac{dx}{dt} &= \alpha \left\{ \frac{d\theta}{dt} \cos t - \theta(t) \sin t \right\} \\ \frac{dy}{dt} &= \alpha\beta \left\{ \frac{d\theta}{dt} \cos t - \theta(t) \sin t \right\}, \end{aligned}$$

and the formula for perimeter is

$$\int \sqrt{\alpha^2 \left\{ \frac{d\theta}{dt} \cos t - \theta(t) \sin t \right\}^2 + \alpha^2 \beta^2 \left\{ \frac{d\theta}{dt} \cos t - \theta(t) \sin t \right\}^2} \, dt.$$

With a few steps of algebra, this reduces to

$$\alpha \int \sqrt{\theta(t)^2 + \theta'(t)^2 + (\beta^2 - 1)\{\theta(t) \cos t + \theta'(t) \sin t\}^2} dt.$$

2.4 *Error Characteristics of a Fluorescence Microscope*

This section is concerned with error characteristics due to the nature of the imaging process in confocal laser scanning fluorescence microscopy. The image observed is not exactly the same as what is being imaged. This is due to three sources: the nature of fluorescence, the microscope and the detector. The binding and emission of fluorescent probes is not exact and can cause artifacts to appear. Lenses and mirrors in the microscope cause the image to be slightly blurred, or distorted. The detector introduces random noise.

2.4.1 The Nature of Fluorescence

Fluorescent probes are very useful for cytometry because of some important characteristics. Fluorescent molecules absorb and emit light at very specific wavelengths, even a very small number of fluorescent molecules can be detected, and they can be designed to be extremely sensitive to their immediate environment. Images can thus be made of very small particles with minimal noise from surrounding particles. The background information in this section is summarized from Taylor and Salmon [98].

Fluorescent molecules work by absorbing light of a specific wavelength and subsequently emitting light of a different (usually longer) wavelength. Light is absorbed and emitted in discrete amounts called photons, or quanta. The amount of energy contained in one photon of light is inversely proportional to its wavelength:

$$\text{Energy} = \frac{1}{\lambda} 10^7 hc \text{ ergs/photon},$$

where h is Planck's constant (6.6×10^{-27} erg seconds), c is the speed of light in a vacuum (3.0×10^{10} cm/second), and λ is the wavelength of the light in nanometers.

Intensity is a measurement of energy per unit time per unit area. Intensity is proportional to the number of photons per unit time and area, and inversely proportional to wavelength:

$$\text{Intensity} = \frac{n}{\lambda} 10^7 hc \text{ ergs/sec/cm}^2,$$

where n is the number of photons.

When a molecule absorbs a photon of light, a valence electron is boosted to a higher energy orbit, putting the molecule into an excited state. Every type of molecule has a set of discrete orbital states, and light absorption occurs only at wavelengths with quantum energy equal to the difference in energy between two of the molecule's orbitals. Thus every fluorescent molecule has an absorption spectrum. Emission occurs when the electron returns to ground state, usually from a lower energy state than the one originally boosted to, so that the emission wavelength is typically lower than the absorption wavelength. This process is called fluorescence only when the time delay from absorption to emission is no more than 10^{-8} seconds. When the delay is longer, the process is called delayed fluorescence (delay $< 10^{-6}$ seconds) or phosphorescence (delay $> 10^{-6}$ seconds).

There are sources of error associated with the use of fluorescent molecules. Some natural cell molecules are autofluorescent and will absorb and emit light, creating noise. The fluorescent probe, though highly sensitive and specific, doesn't necessarily attach to every molecule of interest and attaches to some unwanted molecules. Intensity increases proportionally with probe concentration only up to a point, then drops for various reasons, including photobleaching, a chemical reaction that eliminates the fluorescent properties of a molecule. Since there is a spectrum associated with both absorption and emission, their wavelengths are not exact. Only a certain percentage of photons that hit a molecule are absorbed and only some of those produce a

corresponding emission, causing the emission intensity to be much smaller than the illumination intensity.

Most of the errors due to the nature of fluorescence are usually very small and correctable, and can be ignored for the purposes of analysis. Sometimes there are artifacts on the image from fluorescence of molecules unrelated to the structure of the objects being imaged. This source of error is difficult to model or quantify. A possible method might be to model these artifacts as structures much smaller than the objects of interest that appear randomly on the image. Another possibility might be to consider the artifacts as part of the random noise process.

2.4.2 The Microscope

Microscopic lenses and mirrors are useful to focus, magnify and redirect light with the cost of a small amount of blurring. This blurring is characterized by the convolution of the image with a kernel, or point spread function [108]. This means that the intensity at each point on the blurred image is a combination of intensities of several points on the image. The kernel function is the resulting blurred image when the true image is a single point light source.

Most of the blurring in confocal microscopy is due to the objective lens [108]. The theoretical point spread function for the objective lens in confocal microscopy is known to be [5, 61]

$$\left(2\frac{J_1(u_a)}{u_a}\right)^2\left(2\frac{J_1(u_e)}{u_e}\right)^2,$$

where $u_a = 2\pi A/\lambda_a$, $u_e = 2\pi A/\lambda_e$, A is the numerical aperture of the objective lens, λ_a and λ_e are the absorption and emission wavelengths, and $J_1(x)$ is the first order Bessel function of the first kind:

$$J_1(x) = \frac{x}{2}\sum_{k=0}^{\infty}\frac{(-\frac{1}{4}x^2)^k}{k!\Gamma(k+2)}.$$

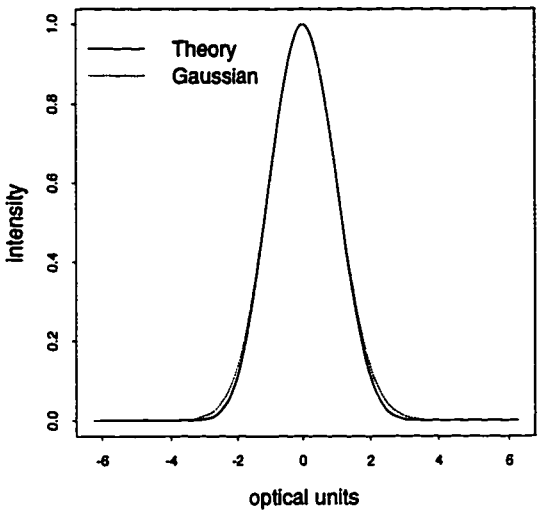


Figure 2.4: Gaussian approximation to the theoretical kernel function. The formula for the theoretical kernel is given in the text.

This can be closely approximated by

$$\left(2\frac{J_1(u_a)}{u_a}\right)^4,$$

if $\lambda_e \approx \lambda_a$. A plot of this function is shown in Figure 2.4 along with the Gaussian density function. The Gaussian function appears to provide a close approximation to the theoretical kernel. Empirical studies [18] show that the observed kernel function is close to the theory. Gaussian functions allow for faster calculations, are a close approximation to the truth, and will thus be the kernel function used in our model.

An important parameter to consider is the width of the kernel function. This is usually expressed in terms of the full width at half maximum (FWHM) of the kernel. For the Gaussian, the maximum is at $x = 0$, so FWHM is the solution to the equation

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(\frac{\text{FWHM}}{2}\right)^2\right\} = \frac{1}{2}\frac{1}{\sqrt{2\pi\sigma^2}}.$$

The FWHM for the Gaussian is thus directly proportional to the standard deviation

(σ):

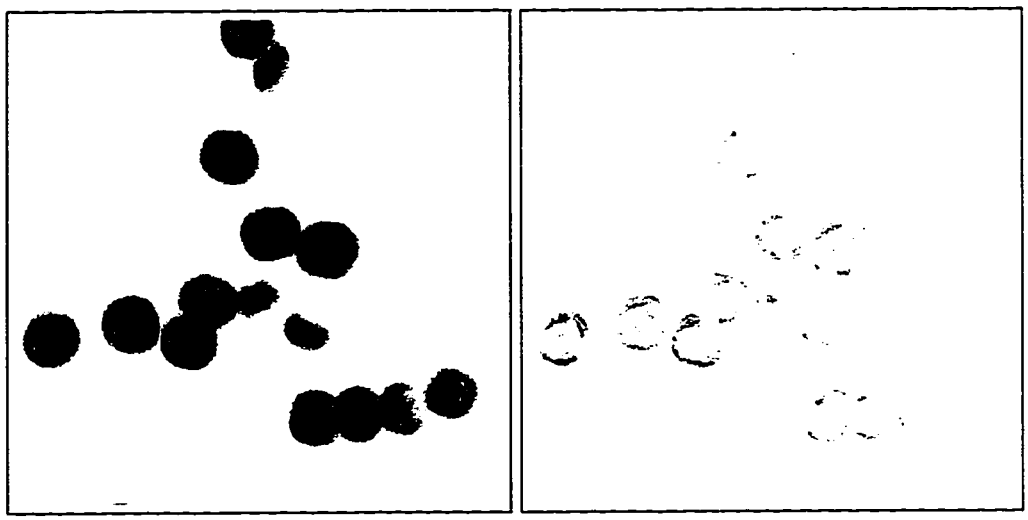
$$\text{FWHM} = 2\sqrt{2\log 2}\,\sigma \approx 2.35482\,\sigma.$$

This parameter can be estimated by imaging a point light source, or by deconvolution methods [21].

2.4.3 The Detector

Most image noise comes from the detector and camera. The two main systems currently used are photomultiplier tubes and CCD's (charge coupled devices) [2, 92, 105]. Both of these systems convert photons of light into electronic signals, which are counted and converted to a 256 gray level image.

There are two sources of noise in these systems: photon count noise and system noise. Photon count noise is a fundamental property of the quantum nature of light [2]. Photons of light detected can be described by a Poisson process with mean depending upon the intensity, which is proportional to the number of photons [2, 5, 92]. This means that the standard deviation of the photon count noise is equal to the number of photons. System noise comes from various sources. The magnitude of this noise depends on the molecules being imaged. The weaker the fluorescent emissions, the higher the detector sensitivity needs to be turned up on the scanner, and thus the higher the system noise. In current systems this noise is negligible for many applications [2, 92]. Therefore, in these situations we expect image noise to be closely approximated by a Poisson distribution with mean depending on the number of photons (μ), implying that the signal to noise ratio (SN) should increase with the square root of the number of samples (n) and with the square root of the number of photons (μ): $\text{SN} = \mu/\sqrt{\mu/n} = \sqrt{n\mu}$. In other applications we might expect to see image noise to appear like white noise, independent of the intensity. Image noise can be estimated by taking repeated samples of the same image.



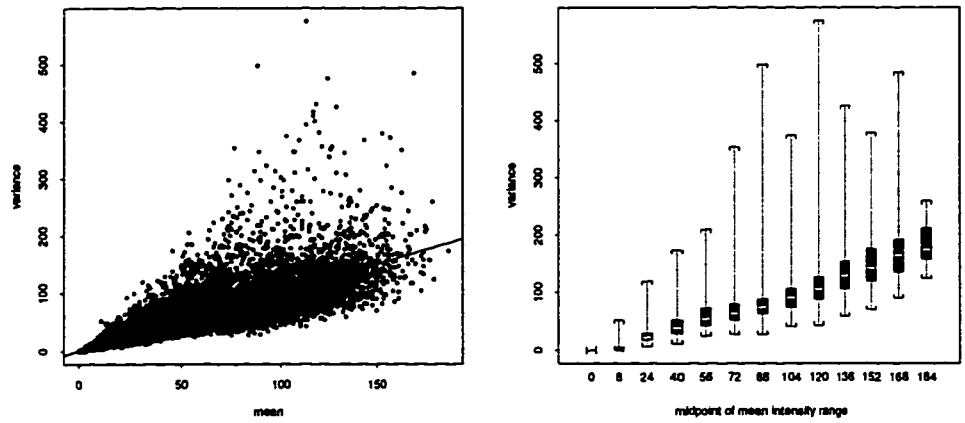
(a) Mean (b) Variance

Figure 2.5: Mean and variance images based on 50 images scanned on the Meridian Laser Cytometer.

2.5 *A Bead Study - What Is the Noise Distribution?*

A fluorescent bead study was conducted to explore the noise distribution in confocal microscopy. Fluorescent beads were mounted on a slide and scanned using the Meridian laser cytometer. The slide was scanned 50 times and each time a 270×270 pixel image was taken of the fluorescence emissions, separated into 256 gray levels and recorded. The noise characteristics of the image data were studied to discover the approximate error distribution of the fluorescence intensity data and the coefficient of variation.

It is first of interest to analyze the noise distribution of the fluorescence image data. Based on theory, it is suspected that the data will approximately follow a Poisson distribution, where the mean is equal to the variance, so that the mean and variance should be closely related. The mean and variance images are shown in Figure 2.5(a-b). Each image shows fluorescence intensity on a 256 level grayscale where black

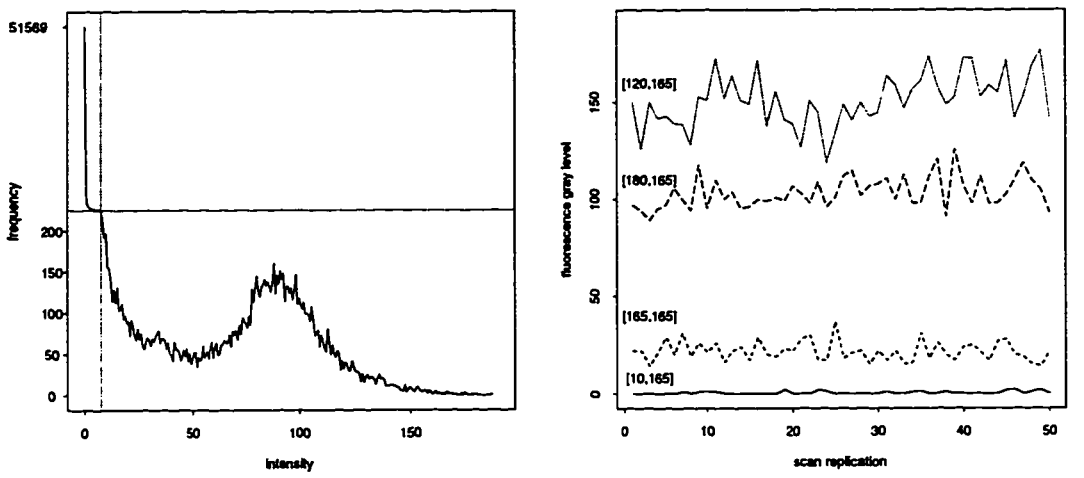


(a) Variance vs. mean intensity (b) Boxplots of mean vs. variance

Figure 2.6: Plots of variance vs. mean. The boxplots in (b) summarize the scatterplot in (a). The boxplots show the median, minimum, maximum, and quartiles of the variance for the given ranges of the mean intensity. The label on the x-axis refers to the midpoint of the mean range. Each range, except 0, is 16 wide, i.e. intervals are {0},(0,16),[16,32),...,[176,190).

is most intense and white is least intense. The images suggest that there is a strong relationship between the mean and variance. The variance image appears to be very similar to the mean image with the exception that the bead area appears lighter due to the presence of spikes (seen as dark spots) in the image.

The plots in Figure 2.6 give further evidence of the strong relationship between the mean and the variance. Figure 2.6(a) illustrates a scatterplot of the variance vs. the mean intensity (correlation = 0.92). Points that fall on the line have variance exactly equal to the mean. There are 72,900 (270 × 270) points on this plot, with the vast majority of them scattered about the line. Figure 2.6b shows a boxplot of the variance for the given ranges of mean intensity. These boxplots give the same information in a more summarized form. It is quite clear that for the majority of the points, the variance is approximately equal to the mean.



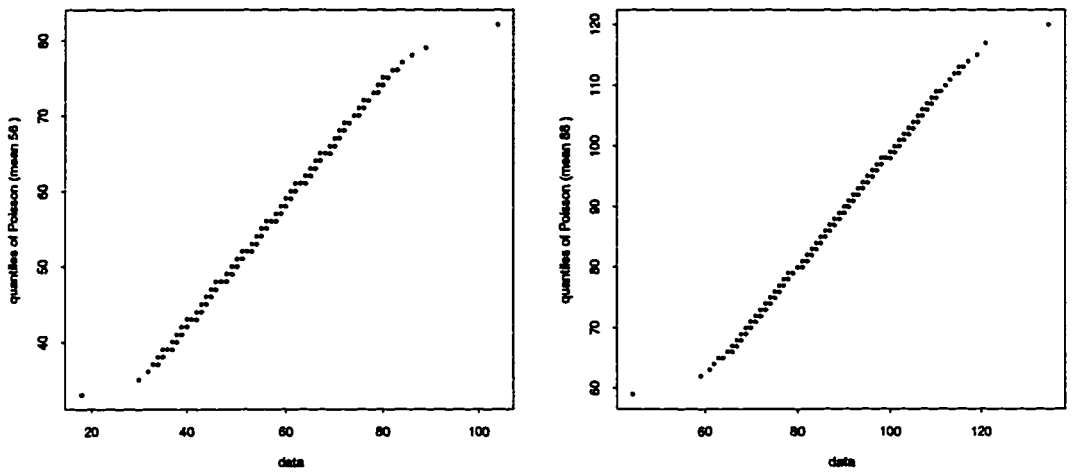
(a) Intensity histogram

(b) Time course plots

Figure 2.7: A histogram of mean intensity and time course plots for selected points on the image. The scale of the histogram is changed for intensity ≤ 10 because of extremely high frequency. The time course plots show all the intensity recordings for 50 scans for the 4 selected points. The numbers in brackets indicate the position of the point on the 270×270 image.

A histogram of mean intensity and time course plots at 4 selected points are shown in Figure 2.7. The histogram shows that most of the pixels have very small intensity; in fact, about 70% have mean intensity equal to 0. There is a local peak in frequency at about 90. The time course plots show no noticeable trend, supporting the hypothesis that the data come from a random sample. The plots also show variability increasing with mean intensity.

The information presented so far provides strong evidence that the variance increases with the mean. To see if the data are approximately Poisson, qq-plots were made for each of 12 mean intensity ranges. The quantiles of the data (all 50 replicates for points with mean in the given range) were plotted against quantiles of the Poisson distribution with mean equal to the midpoint of the mean range. The results were



(a) Midpoint = 56 (b) Midpoint = 88

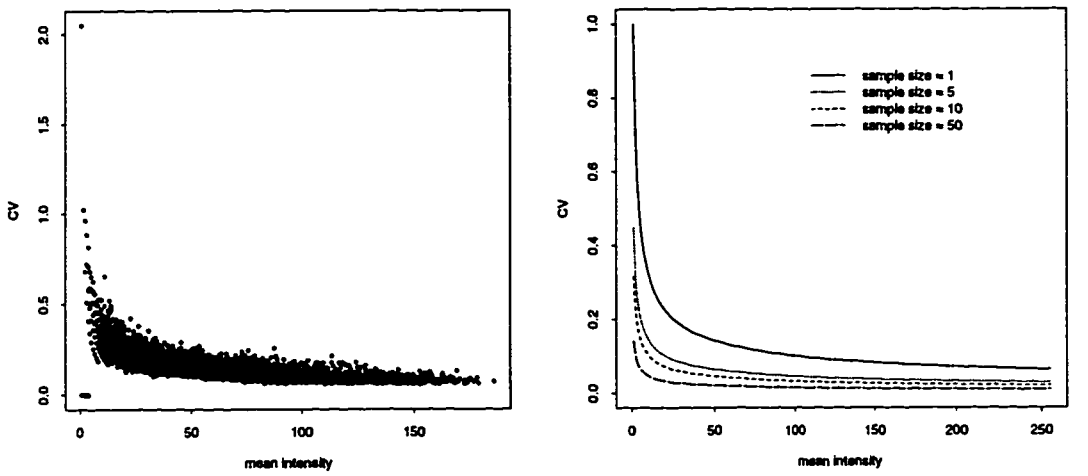
Figure 2.8: QQ-plots of all pixels on image in 2 selected mean ranges vs. quantiles of the Poisson distribution with mean equal to the mean of the data. Points closely following a straight line indicate close approximation to a Poisson distribution with the same mean as the data.

supportive of the hypothesis of Poisson noise. QQ-plots for 2 of the mean ranges are given in Figure 2.8. Most of the qq-plots formed almost perfectly straight lines. Only in the extremes of the data range did this not quite hold, and these were fairly straight as well. Therefore, a Poisson model closely approximates the noise distribution of the data.

The Poisson model can be used to estimate the coefficient of variation (CV) of the scanner. The coefficient of variation is the standard error of the mean divided by the mean:

$$\mathrm{CV} = \frac{\sigma/\sqrt{n}}{\mu} = \frac{\sqrt{\mu/n}}{\mu} = \frac{1}{\sqrt{n\mu}},$$

where μ is the mean, σ is the standard deviation, n is the number of replicates and $\sigma^2 = \mu$ for the Poisson model. Note that CV = 1/SN, where SN is signal to



(a) Coefficient of variation for one scan vs. mean

(b) Coefficient of variation vs. mean for the Poisson distribution with various sample sizes

Figure 2.9: Plots of coefficient of variation vs. mean and vs. number of scans.

noise ratio, given in the previous section. A plot of coefficient of variation vs. mean intensity for the bead data is given in Figure 2.9(a). Plots of theoretical coefficient of variation vs. mean for various sample sizes assuming the Poisson distribution are given in Figure 2.9(b). A sample size of 5 is thus sufficient to keep the coefficient of variation below 10% for the majority of the image pixels.

2.6 A Cell Image Simulator

A cell image simulation program was developed as a first step toward achieving the goal of fitting a statistical model to segment laser cytometry images. In order to be useful, the program needs to generate images under the cell image model. This program generates simulated images of cells marked with a fluorescent probe. The main uses are to validate estimation procedures developed for cell image data and to

quantify the variability associated with such procedures. The program is written in C and uses a Fortran subroutine provided by Finbarr O'Sullivan. Special features of the program are that the image resolution, number of cells, level of overlap, and amount of blur and noise can be set arbitrarily. Additionally, the cell specific parameters (fluorescent intensity, orientation angle, location, size and aspect ratio) can either be set arbitrarily or generated randomly. It is also possible to set the characteristic shape of the cells by specifying a radial function on a unit ellipse.

The function does the following. The image region is made to be a rectangular set of a fixed number of pixels. The number of cells to be generated, the level of blurring, and the amount of noise are determined and then fixed. Several parameters are either fixed or randomly generated. These parameters are the cell mean location (two coordinates in pixel units), the length of the longest semiaxis (size), the ratio of the perpendicular to the long axes (aspect ratio), the rotation angles and the fluorescence intensities. If random, the parameters for cell mean location, aspect ratio and orientation are generated on a unform interval and the parameters for size and intensity are generated from a Gaussian distribution. A shape function for the cells is also specified.

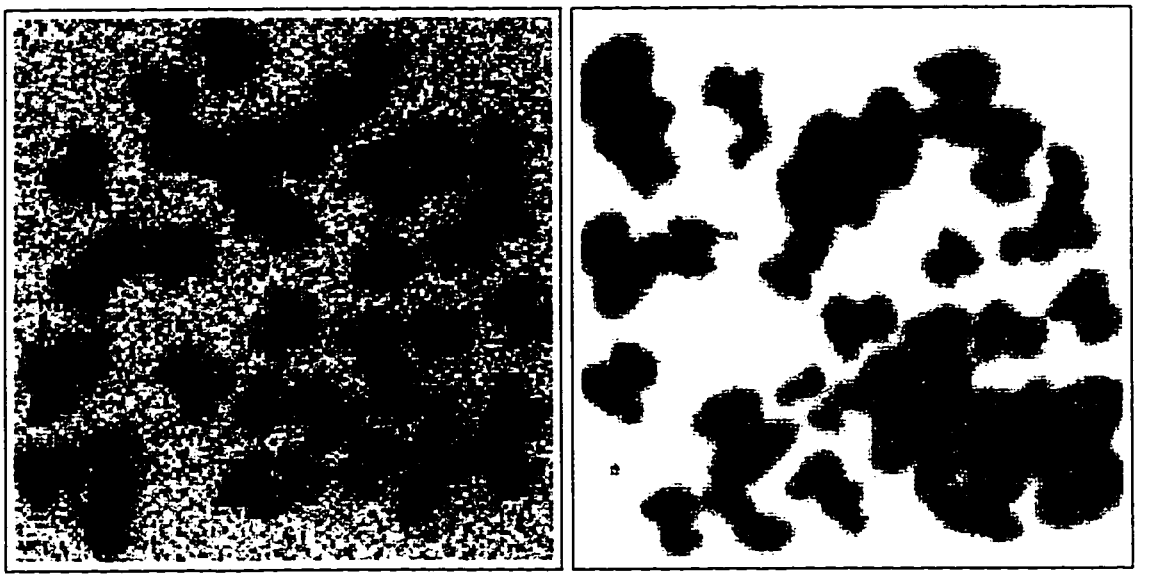
The cell image is then created as follows. The edge of each cell is first parameterized in elliptical coordinates and then a radial shape function is added to create a figure that falls within the starshaped class of shapes. The region enclosed by this figure is considered to be the cell and is given the appropriate fluorescent intensity (held constant for each cell). Additional cells are generated in the same way and the intensity for each cell is added to create an image. Overlapping is controlled by varying how close cells are allowed to be to each other. All cell specific parameters, but not the shape function, are accounted for here. Artifacts can be added as a fixed number of square objects of any size. The next stage is the addition of Gaussian blurring. A two-dimensional Gaussian smoother is applied to the image using a given smoothing parameter. This causes the fluorescent intensity to decrease slowly at the edges of

the cells rather than being cut immediately to zero. The final step is the addition of noise. Noise can be added independently to each pixel that is either proportional to intensity, independent of intensity, or a sum of the two types. The resulting value is taken to be the pixel intensity. The image intensities are then converted to 256 gray levels and stored in PGM format. This can be converted other image types such as GIF, JPEG, BMP or TIFF. It is also fairly straightforward to import PGM files into statistical packages such as S.

There are some generalizations that could be applied to the program. The number of cells included in the image simulation is set by the user and could be randomly generated. The intensity of the fluorescence in the cell is currently held constant and could be allowed to vary with distance from the center; however, an example of this is modeled by the blurring procedure. Cell overlap and clustering is currently modeled with a repulsion factor and uniform distribution on the image. Some sort of clustered Poisson process could be used instead.

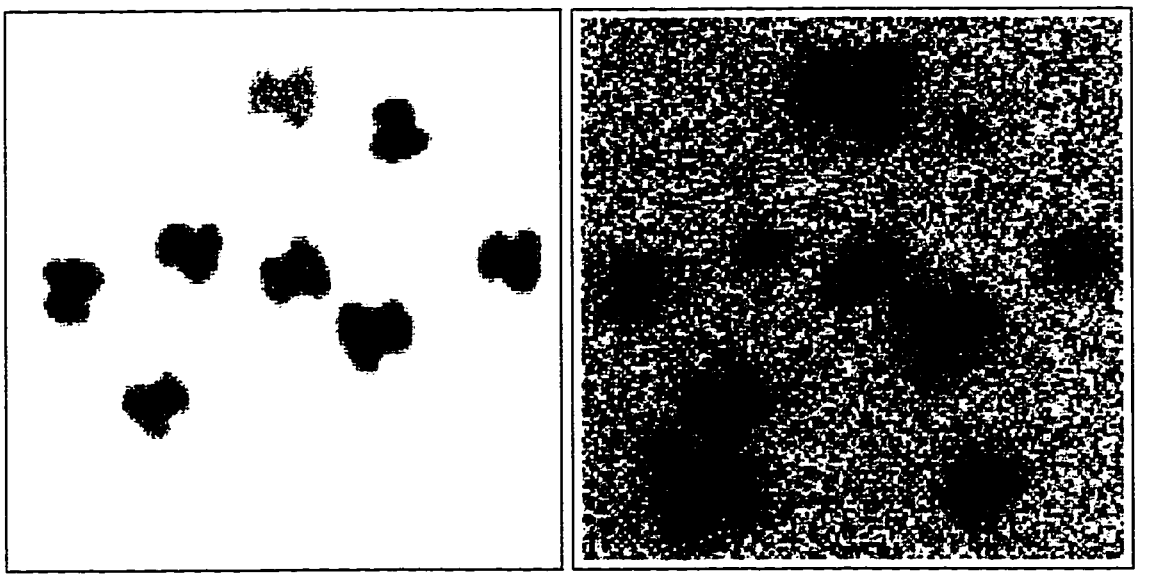
Figure 2.10 illustrates some of the types of images that can be simulated with this program. In Figure 2.10(a), there are a large number of clustered cells with similar sizes and aspect ratios, and a lot of image noise. In Figure 2.10(b), the cells are more variable in size and aspect ratio and the image noise is much smaller. The cells in Figure 2.10(c) are fewer in number, not clustered, and have variable intensity. In Figure 2.10(d), the cells have a lot of variation in size and aspect ratio, and a lot of blurring and noise.

The cell image model incorporates many of the features in cell images. There are some exceptions, for example non-starshaped cells and cells with non-uniform intensity. This model covers a wide range of images while providing a reasonable approximation to many of the rest. The next step then is to develop an algorithm to estimate the parameters of the model. This is the subject of the next chapter.



(a) Many clustered cells, noisy image.

(b) Many clustered cells, variable size and aspect ratio.



(c) A few cells, variable intensity.

(d) Variable cells, noisy and blurred image.

Figure 2.10: Simulated images.

Chapter 3

PARAMETER ESTIMATION

In this chapter, a procedure for parameter estimation is developed for the cell image model described in Chapter 2. The model is a nonlinear convolution model and estimation falls under the class of nonlinear ill-posed inverse problems. A method of regularization procedure is described that involves separating the estimation into two parts and using a modified Gauss-Newton method to do each part. Calculation of functions and derivatives is discussed in some detail.

3.1 Overview

Recall the cell image model for intensity z at a point x:

$$z(x) = \sum_{i=1}^{n} \gamma_i \alpha_i^2 \beta_i \int_0^{2\pi} \int_0^{\theta(t)} g_h(x - \mu_i - B_i(t)r) r dr dt + \epsilon(x),$$

where n is the number of cells, $g_h(x) = \exp(-\|x\|^2/2h^2)/(2\pi h^2)$, $\epsilon(x)$ is independent noise, $B_i(t) = A_i(\cos t, \sin t)^T$, and

$$A_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \alpha_i \begin{bmatrix} 1 & 0 \\ 0 & \beta_i \end{bmatrix}.$$

The shape function is $\theta : [0, 2\pi) \to R^+$ and $\psi = \{\mu, \alpha, \beta, \phi, \gamma\}$ are the cell specific parameters.

This model is an example of a nonlinear convolution. It is nonlinear in the parameters θ, μ, α, β and ϕ but linear in γ. This means that $z(x|f_1\gamma_1 + f_2\gamma_2) = f_1 z(x|\gamma_1) + f_2 z(x|\gamma_2)$, but this property does not, in general, hold for the other parameters.

Parameter estimation under this model falls into the class of nonlinear ill-posed inverse problems. It is called an inverse problem because the parameters θ and ψ are estimated based on noisy observations on functionals $z(x)$. Inverse problems are a general class of problems that include linear regression. Ill-posed problems are characterized by lack of a unique solution or by estimates largely affected by slight perturbations in the data.

To deal with the ill-posedness of the problem, a method of regularization procedure [101, 102] is used. The method of regularization involves optimizing an objective functional of the form:

$$L(z, N(\theta, \psi)) + \lambda J(\theta).$$

The functional L measures the lack of fit between the data z and $N(\theta, \psi)$, where N is a nonlinear operator acting on θ and ψ; J is a penalty functional and measures closeness to prior beliefs about the function θ; and the parameter λ controls the tradeoff between L and J. For the cell image model,

$$N(\theta, \psi) = \sum_{i=1}^{n} \gamma_i \alpha_i^2 \beta_i \int_0^{2\pi} \int_0^{\theta(t)} g_h(x - \mu_i - B_i(t)r) r dr dt.$$

The method of regularization is used in many contexts. If the lack of fit functional is a log likelihood function, the procedure is equivalent to maximum penalized likelihood estimation [38]. If the penalty functional is the inverse of the covariance matrix of a prior distribution, the method is equivalent to maximum *a posteriori* estimation [48]. If the lack of fit functional is the least squares criterion and the penalty functional is the integral of the squared second derivative, the method is equivalent to spline smoothing [104]. The method of regularization procedure has also proven useful in problems arising in Positron Emission Tomography [56, 68, 74] and stereology [66].

In this dissertation, the estimates of θ and ψ are chosen to be the minimizer of the objective functional:

$$l_{n,m,\lambda,h}(\theta, \psi) = \frac{1}{m} \sum_{k=1}^{m} \{z(x_k) - \sum_{i=1}^{n} f_i(x_k | h, \theta, \psi_i)\}^2 + \lambda J(\theta),$$

where $J(\theta) = \int_0^{2\pi}\{\theta''(t)\}^2 dt$, the parameter λ is the smoothing parameter, and the rest of the parameters are as before. This criterion is known as penalized least squares (PLS). The first term, the lack of fit functional, is the least squares criterion. The second term consists of the smoothing parameter and the penalty functional. It is called a penalty because it penalizes functions that are not smooth in terms of the integrated squared second derivative. The smoothing parameter is so called because it controls the tradeoff between closeness to the data and smoothness of the function. Setting $\lambda = 0$ allows θ to interpolate the data and letting $\lambda \to \infty$ forces θ to be a straight line. If the functions f_i were linear in θ, the solution for θ would be a natural polynomial spline [104]. The penalized least squares estimator can also be interpreted as a Bayes estimator with a Gaussian process prior.

3.2 Main Algorithm

There are many ways to obtain estimates of the parameters that minimize the objective functional. Because of the nature of the model, the model is broken into two parts and each part solved separately using a modified Gauss-Newton type of algorithm. The characteristic shape function θ is estimated conditional on a current estimate of the cell specific parameters ψ and vice versa. This could be done in a two step manner by estimating the cell parameters first and then the shape function. Optimally, it would be done iteratively, alternating shape function and cell parameter estimation and updating the current estimates each time.

3.2.1 Characteristic Shape Function Estimation

This part of the algorithm conditions on the cell specific parameters ψ to obtain an estimate of the characteristic shape function θ. The estimate $\hat{\theta}$ is the minimizer of the objective functional

$$l_{n,m,\lambda,h}(\theta) = \frac{1}{m}\sum_{k=1}^{m}\{z(x_k) - \sum_{i=1}^{n} f_i(x_k|h,\theta)\}^2 + \lambda\int_0^{2\pi}\{\theta''(t)\}^2 dt.$$

The functionals f_i are nonlinear in θ so a Gauss-Newton approach [28, 89] is used to obtain the estimate $\hat{\theta}$. This involves starting with an initial estimate θ^0, approximating f_i with a functional linear in θ, and obtaining successive updates until convergence is reached. The initial estimate can be any prior guess or simply a constant function corresponding to an ellipse. The linear approximation used is the first order Taylor expansion about θ^0:

$$f_i(\theta) \approx f_i(\theta^0) + D_\theta f_i(\theta^0)(\theta - \theta^0),$$

where $D_\theta f_i(\theta^0)\zeta$ is the functional derivative of $f_i(\theta)$ with respect to θ in the direction ζ evaluated at θ^0.

Since it is infinite-dimensional, θ is approximated for computational purposes with a finite sum of coefficients and basis functions:

$$\theta(t) \approx \sum_{j=1}^{J} \eta_j \xi_j(t),$$

where

$$\xi_j(t) = I\Big\{t \in \Big[\frac{j-1}{J}, \frac{j}{J}\Big)2\pi\Big\}, \quad j = 1, \ldots, J.$$

The integer J needs to be chosen large enough to characterize the shape function but small enough to keep computation reasonable. The vector $\eta = \{\eta_1, \ldots, \eta_J\}$ is the coefficients and $\xi = \{\xi_1, \ldots, \xi_J\}$ is the basis functions. Now the parameter to be estimated is the J-dimensional vector of coefficients η. The penalty functional can now be written as a function of η as

$$J(\eta) = \sum_{j=1}^{J} \{\eta_{j+1} - 2\eta_j + \eta_{j-1}\}^2,$$

where $\eta_0 = \eta_J$ and $\eta_{J+1} = \eta_1$.

At each iteration the objective functional to minimize is

$$l_{n,m,\lambda,h}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\Big(z(x_k) - \sum_{i=1}^{n}\{f_i(x_k|h,\eta^0) + D_\eta f_i(x_k|h,\eta^0)(\eta - \eta^0)\}\Big)^2 + \lambda J(\eta),$$

where $D_\eta f_i(x_k|h,\eta^0)$ is the vector of ordinary partial derivatives with respect to each coefficient:

$$D_\eta f_i(x_k|h,\eta^0) = \{D_{\eta_1} f_i(x_k|h,\eta^0), \ldots, D_{\eta_J} f_i(x_k|h,\eta^0)\}.$$

Let $z^* = \{z^*(x_1), \ldots, z^*(x_m)\}^T$ be the pseudovalues and $X = \{X(x_1), \ldots, X(x_m)\}^T$ the corresponding design matrix, where

$$\begin{aligned} z^*(x_k) &= z(x_k) - \sum_{i=1}^{n} f_i(x_k|h,\eta^0) + \sum_{i=1}^{n} D_\eta f_i(x_k|h,\eta^0)\eta^0 \\ X(x_k) &= \sum_{i=1}^{n} D_\eta f_i(x_k|h,\eta^0). \end{aligned}$$

Then the objective functional can be written as

$$\begin{aligned} l_{n,m,\lambda,h}(\eta) &= \frac{1}{m}\sum_{k=1}^{m}\{z^*(x_k) - X(x_k)\eta\}^2 + \lambda J(\eta) \\ &= \frac{1}{m}(z^* - X\eta)^T(z^* - X\eta) + \lambda\eta^T\Omega\eta, \end{aligned}$$

where $\Omega = \Gamma^T\Gamma$ and

$$\Gamma = \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 1 & -2 \end{bmatrix}.$$

The new estimate $\hat{\eta}$ given a current estimate η^0 is then given by

$$\hat{\eta} = (X^TX + \lambda\Omega)^{-1}X^Tz^*.$$

The process is repeated by replacing η^0 with $\hat{\eta}$, updating z^* and X, and obtaining a new estimate $\hat{\eta}$.

This continues until

$$\frac{|l(\hat{\eta}) - l(\eta^0)|}{1 + l(\hat{\eta})} < \epsilon$$

or

$$\|\hat{\eta} - \eta^0\|^2 < \epsilon,$$

where ϵ is chosen to match machine precision.

3.2.2 Cell Specific Parameter Estimation

This part of the algorithm conditions on the characteristic shape function θ to obtain an estimate of the cell specific parameters ψ. In this case, $\lambda J(\theta)$ is a constant and the estimate $\hat{\psi}$ is the minimizer of the functional

$$l_{n,m,h}(\psi) = \frac{1}{m}\sum_{k=1}^{m}\{z(x_k) - \sum_{i=1}^{n} f_i(x_k|h,\psi)\}^2.$$

This is a nonlinear least squares problem. As with the characteristic shape function, a modified Gauss-Newton procedure is used. The initial estimates can be the estimates from a previous iteration or any other prior estimate.

To obtain the updating equation, define z^* and X as before but make the derivatives with respect to ψ instead of η. The updated estimate $\hat{\psi}$ given ψ^0 is then

$$\hat{\psi} = (X^T X)^{-1} X^T z^*.$$

3.3 Modifications

Due to the nonlinear nature of the model and the high dimensionality of the parameter space, estimation is potentially quite unstable. Estimating the shape curve and cell specific parameters separately has already been discussed. Other things to deal with are local minima, a large parameter space, modifying the Gauss-Newton step, and placing constraints on the parameters.

The algorithm is quite sensitive to the initial estimates. In fact, if the initial guesses for location μ are too far from the truth, say outside the cell, the algorithm is likely to fail. This is not surprising because we expect the first order Taylor approximation to be good only near the initial value. To deal with this, a function

was developed in the S language to read a PGM formatted image and allow a user to use a computer mouse to identify cell locations and four landmarks describing the longest and perpendicular axes of a typical cell. This gives initial values for location μ, size α and aspect ratio β. The orientation ϕ is initialized at zero and the intensity γ at the highest value on the image (255 for a PGM image). For the shape function θ, it is sufficient to use the constant function corresponding to an ellipse.

The objective function can have more than one minimum with respect to the orientation parameter ϕ even in the simplest cases. Other parameters can also be led astray if the orientation is too far from the true value. For this reason, each iteration is started with a search for the optimal orientation parameter before proceeding with the Gauss-Newton step. At first the search is over many possible values, and the number of values is reduced with each successive iteration. This is done because the orientation parameter has seemed to be more stable near convergence and this search procedure is computationally burdensome. Another way to deal with this problem might be to use another optimization method such as simulated annealing. The paper by Bohachevsky et al. [15] gives a good description of the basic method. We have avoided this mainly for computational reasons; however, in the future it might become a more feasible alternative.

The dimension of the parameter space in the cell image model is equal to

$$\dim(\theta, \psi) = \dim(\theta) + \dim(\psi) = J + 6n,$$

where J is the dimension of the shape curve θ, n is the number of cells, and 6 is the number of cell specific parameters per cell. This means that while the dimension of the shape curve is constant with respect to the number of cells, the dimension of the cell specific parameters increases linearly with the number of cells. This means that, for the iterative procedure for the cell specific parameters, inversion of the design matrix is numerically unstable when the number of cells is large. This is dealt with by inverting the matrix in an approximate method. The approximation used has the

same effect as updating only a subset of cells at a time or assuming that certain cells are independent. For cells that are far apart, the corresponding columns of the design matrix are independent in the sense that $x^T y = 0$ for any column x corresponding to a parameter of one cell and column y corresponding to a parameter of the other cell. In contrast, this is almost never true if x and y are columns corresponding to two different parameters from the same cell. Thus $X^T X$ has a block diagonal structure if all cells are sufficiently far apart and the design matrix is arranged in the following order:

$$\begin{aligned} X(x) &= \sum_{i=1}^{n} D_{\psi} f_i(x|\psi^0) \\ &= \{D_{\psi_1} f_1(x|\psi^0), \ldots, D\psi_n f_n(x|\psi^0)\}, \end{aligned}$$

where

$$D_{\psi_i} f_i(x|\psi^0) =$$
$$\{D_{\mu_i} f_i(x|\psi^0), D_{\alpha_i} f_i(x|\psi^0), D_{\beta_i} f_i(x|\psi^0), D_{\phi_i} f_i(x|\psi^0), D_{\gamma_i} f_i(x|\psi^0)\}.$$

It is unclear whether updating the cells this way will result in loss or gain in numerical efficiency. There is a trade-off between stability in the sense of matrix inversion and stability in the sense of avoiding local minima. The approach used is to update a few cells at a time in hope of providing some balance between the extremes. A method that might be even better is region-based updating, where all cells in each region are updated simultaneously.

The Gauss-Newton algorithm is an iterative algorithm involving a parameter update and a stopping rule. The so-called Gauss-Newton update, $\hat{\delta} = \hat{\eta} - \eta^0$, is not necessarily the optimal choice in terms of computation speed. The algorithm will usually converge in fewer steps if a descent method [28, 89] is used. One way to do this is to define

$$\eta^{\rho} = \eta^0 + \rho(\hat{\eta} - \eta^0),$$

so that $\eta^0 = \eta^0$ and $\eta^1 = \hat{\eta}$, and choose $\rho \in [0, \infty)$ to minimize

$$l(\eta^\rho) = \|z - \sum_i f_i(\eta^\rho)\|^2 + \lambda \eta^{\rho T} \Omega \eta^\rho.$$

For computational purposes, the minimum is actually chosen from about 10 possible values: $\rho = 2^i, i = -7, \ldots, 4$. The effect of this step length procedure, or descent method, is to reduce the number of iterations required for the algorithm to converge, with the cost of adding a relatively small amount of computation to each iteration.

Constraints are incorporated into the algorithm in a post-hoc manner at each iteration. That is, the updated estimates, after being computed, are adjusted if they fall out of a specified range. For the shape curve, the constraint is simply that the curve be positive. This constraint forces the cell "center" to actually be in the cell. The locations are constrained to be on the image. The size parameter is forced to be greater than 1 pixel and less than half the largest image dimension. Aspect ratio is constrained to be between 0 and 1. Orientation is kept between 0° and 360°. Intensity is forced to be positive and no more than a fixed percent higher than the highest intensity seen on the image.

3.4 Calculation of Cell Function and Derivatives

In order to implement the estimation algorithm, a method is needed for calculating the cell function and its derivatives with respect to the various parameters. In this section, two methods for calculating the cell function, the calculations for derivatives and the divided difference approximations for the derivatives are all derived in detail.

3.4.1 Cell Function

The cell function is a convolution that can be calculated as a double integral directly by converting the inner integral to a sum of easily calculable functions and using the

trapezoidal rule on the outer integral. Recall the cell function in polar cell coordinates:

$$f_i(x|h,\theta,\psi_i) = \gamma_i \alpha_i^2 \beta_i \int_0^{2\pi} \int_0^{\theta(t)} g_h\left(x - \mu_i - B_i(t)r\right) r\,dr\,dt.$$

To calculate the inner integral, rewrite it as

$$\int_0^{\theta(t)} \frac{1}{2\pi h^2} \exp\left\{-\frac{1}{2h^2}\left\|x - \mu_i - B_i(t)r\right\|^2\right\} r\,dr.$$

Suppressing the i subscripts, the norm squared portion is

$$\begin{aligned}
&\|x - \mu - B(t)r\|^2 \\
&= \left(x_1 - \mu_1 - \{\alpha\cos\phi\ \cos t - \alpha\beta\sin\phi\ \sin t\}r\right)^2 + \Big(x_2 - \mu_2 - \{\alpha\sin\phi\ \cos t \\
&\quad + \alpha\beta\cos\phi\ \sin t\}\Big)^2 \\
&= (x_1-\mu_1)^2 + (x_2-\mu_2)^2 - 2\Big\{(x_1-\mu_1)(\alpha\cos\phi\ \cos t - \alpha\beta\sin\phi\ \sin t) \\
&\quad + (x_2-\mu_2)(\alpha\sin\phi\ \cos t + \alpha\beta\cos\phi\ \sin t)\Big\}r + \Big\{(\alpha\cos\phi\ \cos t - \alpha\beta\sin\phi\ \sin t)^2 \\
&\quad + (\alpha\sin\phi\ \cos t + \alpha\beta\cos\phi\ \sin t)^2\Big\}r^2 \\
&\equiv a_1(t) - 2a_2(t)r + a_3(t)^2 r^2 \\
&= \left\{a_3(t)r - \frac{a_2(t)}{a_3(t)}\right\}^2 + a_1(t) - \frac{a_2(t)^2}{a_3(t)^2}
\end{aligned}$$

Suppressing the dependence of θ, a_1, a_2, and a_3 on t, the inner integral can be written as

$$\begin{aligned}
&\int_0^{\theta} \frac{1}{2\pi h^2} \exp\left\{-\frac{(a_3 r - a_2/a_3)^2 + a_1 - {a_2}^2/{a_3}^2}{2h^2}\right\} r\,dr \\
&\quad = \frac{1}{2\pi h^2} \exp\left\{-\frac{a_1 - {a_2}^2/{a_3}^2}{2h^2}\right\} \int_0^{\theta} \exp\left\{-\left(\frac{a_3 r - a_2/a_3}{h\sqrt{2}}\right)^2\right\} r\,dr.
\end{aligned}$$

Let $y = (a_3 r - a_2/a_3)/(h\sqrt{2})$. Then $dy = (a_3/(h\sqrt{2}))dr$, $r = (yh\sqrt{2} + a_2/a_3)/a_3$, and the integral part can be rewritten as

$$\int_L^U e^{-y^2} \left\{\frac{yh\sqrt{2} + a_2/a_3}{a_3}\right\} \frac{h\sqrt{2}}{a_3} dy,$$

where $L = -a_2/(a_3 h\sqrt{2})$ and $U = (a_3\theta - a_2/a_3)/(h\sqrt{2})$. This integral can be evaluated as follows:

$$\begin{aligned}
&\int_L^U e^{-y^2}\left\{\frac{yh\sqrt{2}+a_2/a_3}{a_3}\right\}\frac{h\sqrt{2}}{a_3}dy \\
&= \left(\frac{h\sqrt{2}}{a_3}\right)^2\int_L^U ye^{-y^2}dy + \frac{h\sqrt{2}}{a_3}\frac{a_2/a_3}{a_3}\int_L^U e^{-y^2}dy \\
&= \frac{h^2}{{a_3}^2}\left\{e^{-L^2}-e^{-U^2}\right\} + \frac{a_2 h\sqrt{\pi}}{{a_3}^3\sqrt{2}}\left\{\mathrm{erf}(U)-\mathrm{erf}(L)\right\},
\end{aligned}$$

where erf is the error function,

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}dy,$$

which can easily be evaluated using available Fortran or C functions.

To evaluate the full integral, the inner integral is evaluated at N points and the trapezoidal rule is used for the outer integral. Let $b(t)$ be the inner integral evaluated at t. Then the cell function can be written as

$$\begin{aligned}
f_i(x|h,\theta,\psi_i) &= \gamma_i\alpha_i^2\beta_i\int_0^{2\pi} b(t)dt \\
&\approx \gamma_i\alpha_i^2\beta_i\frac{2\pi}{N}\left\{\frac{1}{2}b(t_0)+b(t_1)+\cdots+b(t_{N-1})+\frac{1}{2}b(t_N)\right\},
\end{aligned}$$

where $0 \equiv t_0 < t_1 < \cdots < t_N \equiv 2\pi$, and the t's are equally spaced at $2\pi/N$ apart, i.e. $t_i = 2\pi i/N$. Choose N just large enough to adequately characterize the function. Alternatively, the integral can be evaluated on finer and finer partitions until convergence.

3.4.2 Faster Cell Function Calculation

Cell function calculation by direct integration turns out to be computationally excessive in practice. This occurs not only because calculating convolutions by direct integration is slow, but because it can only be done for one cell at a time. An alternative approach is to use Fourier transforms. This not only turns out to be faster for one cell but is only necessary once for the whole image.

For a function g, its Fourier transform is

$$\check{g}(\nu) = \int g(t)e^{-i2\pi\nu t}dt.$$

This converts the function from the *time* domain to the *frequency* domain. The function g can be recovered using the inverse Fourier transform:

$$g(t) = \int \check{g}(\nu)e^{i2\pi t\nu}d\nu.$$

A useful result from Fourier theory is that the Fourier transform of the convolution between two functions is the product of their Fourier transforms. That is, if

$$h(x) = \int k(y)g(x-y)dy,$$

then

$$\check{h}(x) = \check{k}(x)\check{g}(x),$$

and the convolution is obtained by taking the inverse Fourier transform of $\check{h}(x)$.

In order for Fourier transformations to be useful, methods are needed to calculate each function and the Fourier transforms. Note that the cell image model is a sum of convolutions, and can be rewritten as a convolution of sums:

$$\begin{aligned}
\sum_{i=1}^{n} f_i(x|h,\theta,\psi_i) &= \sum_i \gamma_i (I_{R_i(\theta,\psi_i)} * g_h)(x) \\
&= \sum_{i=1}^{n} \gamma_i \int I_{R_i(\theta,\psi_i)}(y) g_h(x-y)dy \\
&= \int \sum_{i=1}^{n} \gamma_i I_{R_i(\theta,\psi_i)}(y) g_h(x-y)dy \\
&= (\sum_{i=1}^{n} \gamma_i I_{R_i(\theta,\psi_i)} * g_h)(x).
\end{aligned}$$

To evaluate this convolution, $I_{R_i(\theta,\psi_i)}(x)$ needs to be calculated for all x and i. This can be done by transforming to cell coordinates then polar coordinates. Suppressing

the i subscripts, let $u = A^{-1}(x - \mu)$, where

$$A = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \alpha \begin{bmatrix} 1 & 0 \\ 0 & \beta \end{bmatrix}.$$

Then let $r(t) = \|u\| = \sqrt{u_1^2 + u_2^2}$, and $t = H(u)$, where $H(u) \equiv \arctan(u_2/u_1) + \pi I\{u_1 < 0\} + 2\pi I\{u_1 \geq 0, u_2 < 0\}$, and $0/0 \equiv 0$. Then

$$I_{R_i(\theta,\psi_i)}(x) = I\{r_i(t) \leq \theta(t)\} = I\left\{\|A_i^{-1}(x - \mu_i)\| \leq \theta\left(H[A_i^{-1}(x - \mu_i)]\right)\right\},$$

which is simple to calculate. The indicator functions can then be added together and one convolution performed.

The convolution is performed using the Fast Fourier Transform (FFT), which not surprisingly is a discrete method of computing Fourier transforms quickly. This means the region indicators $\sum_{i=1}^{n} \gamma_i I_{R_i(\theta,\psi_i)}$ and the Gaussian kernels g_h are calculated only at the center of each pixel in the image domain. The FFT is able to take advantage of the discreteness to compute convolutions quickly. The fact that the kernel is Gaussian helps as well. These functions are performed using a Fortran subroutine provided by Finbarr O'Sullivan that uses functions from cmlib.

The two methods are nearly indistinguishable. Any differences between them are due to the discreteness of the second method. Since the Gaussian kernel is an approximation (although a good one) to begin with, the minor differences are easily made up for by the huge improvement in computation speed.

3.4.3 Derivative for Shape Function

The derivative for the shape function is a functional derivative of the cell function. This can be written down analytically as an integral and computed using the trapezoidal rule. The derivative of the cell function with respect to the characteristic shape

function is given by

$$\begin{aligned} D_\theta f_i(x|\theta)\zeta &= \lim_{s\to 0}\frac{d}{ds} f_i(x|\theta+s\zeta) \\ &= \lim_{s\to 0}\frac{d}{ds}\gamma_i\alpha_i^2\beta_i\int_0^{2\pi}\int_0^{\theta(t)+s\zeta(t)} g_h\left(x-\mu_i-a_i(t)r\right) r\,dr\,dt \\ &= \lim_{s\to 0}\gamma_i\alpha_i^2\beta_i\int_0^{2\pi} g_h\Big(x-\mu_i-a_i(t)\{\theta(t)+s\zeta(t)\}\Big)\{\theta(t)+s\zeta(t)\}\zeta(t)dt \\ &= \gamma_i\alpha_i^2\beta_i\int_0^{2\pi} g_h\Big(x-\mu_i-a_i(t)\theta(t)\Big)\theta(t)\zeta(t)dt. \end{aligned}$$

This form of functional derivative is known as the Gâteaux derivative. The more general Hadamard and Fréchet forms of the derivative, if they exist, are equivalent to the Gâteaux in the direction ζ. The derivative can be calculated using the trapezoidal rule by rewriting the derivative as

$$\begin{aligned} D_{\theta,\zeta} f_i(x|\theta) &\equiv \int_L^U b(t)dt \\ &\approx \frac{U-L}{N}\Big\{\frac{1}{2}b(t_0)+b(t_1)+\cdots+b(t_{N-1})+\frac{1}{2}b(t_N)\Big\}, \end{aligned}$$

where $L \equiv t_0 < t_1 < \cdots < t_N \equiv U$, and the t's are equally spaced at $(U-L)/N$ apart, i.e. $t_i = L + (U-L)i/N$. As before, choose N large enough but not too large or refine partitions until convergence.

3.4.4 Derivatives for Cell Specific Parameters

To implement the algorithm for estimation of size, aspect ratio, orientation, location and intensity, formulas are needed to calculate the derivative of the cell function with respect to each of these parameters. In each of the following derivative calculations, the dependence on cell (subscript i) is ignored.

The derivative of the cell function with respect to size is given by

$$\begin{aligned}
&\frac{d}{d\alpha} f(x|h,\theta,\psi) \\
&= \frac{d}{d\alpha}\gamma\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= 2\gamma\alpha\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&\quad + \gamma\alpha^2\beta\int_{R(\theta)} \left\{\frac{-1}{2h^2}\right\}\left\{\frac{d}{d\alpha}\|x-\mu-Au\|^2\right\} g_h(x-\mu-Au)du \\
&= \frac{2}{\alpha} f(x|h,\theta,\psi) + \frac{\gamma\alpha^2\beta}{h^2}\int_{R(\theta)} (x-\mu-Au)^T Au \frac{1}{\alpha} g_h(x-\mu-Au)du.
\end{aligned}$$

The derivative of the cell function with respect to aspect ratio is given by

$$\begin{aligned}
&\frac{d}{d\beta}\gamma\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= \frac{1}{\beta} f(x|h,\theta,\psi) + \frac{\gamma\alpha^2\beta}{-2h^2}\int_{R(\theta)} \frac{d}{d\beta}\left\{(-2x+2\mu+Au)^T Au\right\} g_h(x-\mu-Au)du \\
&= \frac{1}{\beta} f(x|h,\theta,\psi) + \frac{\gamma\alpha^2\beta}{h^2}\int_{R(\theta)} \{\alpha u_2 (x-\mu)^T b(\phi) - \alpha^2\beta u_2^2\} g_h(x-\mu-Au)du,
\end{aligned}$$

where $b(\phi) = [-\sin\phi, \cos\phi]^T$. The derivative with respect to orientation is given by

$$\begin{aligned}
&\frac{d}{d\phi}\gamma\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= \frac{\gamma\alpha^2\beta}{h^2}\int_{R(\theta)} (x-\mu-Au)^T (\frac{d}{d\phi}Au) g_h(x-\mu-Au)du \\
&= \frac{\gamma\alpha^2\beta}{h^2}\int_{R(\theta)} (x-\mu-Au)^T (d_\phi A)u \; g_h(x-\mu-Au)du,
\end{aligned}$$

where

$$d_\phi A \equiv \frac{d}{d\phi}A = \begin{bmatrix} -\sin\phi & -\cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix} \alpha \begin{bmatrix} 1 & 0 \\ 0 & \beta \end{bmatrix} \frac{\pi}{180},$$

so that

$$(d_\phi A)A^{-1} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

The derivative with respect to location is given by

$$\begin{aligned}
&\frac{d}{d\mu}\gamma\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= \frac{\gamma\alpha^2\beta}{h^2}\int_{R(\theta)}(x-\mu-Au)g_h(x-\mu-Au)du.
\end{aligned}$$

Finally, the derivative with respect to intensity is given by

$$\begin{aligned}
&\frac{d}{d\gamma}\gamma\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= \frac{d}{d\gamma}\alpha^2\beta\int_{R(\theta)} g_h(x-\mu-Au)du \\
&= \frac{1}{\gamma}f(x|h,\theta,\psi).
\end{aligned}$$

To evaluate these derivatives, the trapezoidal rule could be used, but that would be way too slow. Instead, the Fast Fourier Transform method is used. This is described next.

3.4.5 Faster Derivative Calculation

The derivatives for the cell specific parameters can all be calculated using the Fast Fourier Transform method using the same Gaussian blurring function used to calculate the cell function. The derivative for the shape function cannot be written in terms of the image dimensions, so a divided difference approximation to the derivative is used.

The derivative for size can be rewritten as

$$\begin{aligned}
&\frac{2}{\alpha}f(x|h,\theta,\psi)+\frac{\gamma}{h^2}\int_{R(\theta,\psi)}(x-y)^T(y-\mu)\frac{1}{\alpha}g_h(x-y)dy \\
&= \frac{2}{\alpha}f(x|h,\theta,\psi)+\frac{\gamma}{\alpha h^2}\int_{R(\theta,\psi)}(x-\mu)^T(y-\mu)g_h(x-y)dy \\
&\quad -\frac{\gamma}{\alpha h^2}\int_{R(\theta,\psi)}\|y-\mu\|^2 g_h(x-y)dy \\
&= \frac{2}{\alpha}f(x|h,\theta,\psi)+\frac{\gamma}{\alpha h^2}\Big(x-\mu\Big)^T\Big(\int_{R(\theta,\psi)}(y-\mu)g_h(x-y)dy\Big) \\
&\quad +\frac{\gamma}{\alpha h^2}\int_{R(\theta,\psi)}\|y-\mu\|^2 g_h(x-y)dy.
\end{aligned}$$

The derivative for aspect ratio can be rewritten as

$$\begin{aligned}
&\frac{1}{\beta}f(x|h,\theta,\psi) \\
&\quad + \frac{\gamma}{h^2}\int_{R(\theta,\psi)}\left\{\frac{\alpha}{\alpha\beta}(y-\mu)^T b(\phi)(x-\mu)^T b(\phi) - \alpha^2\beta\{\frac{1}{\alpha\beta}(y-\mu)^T b(\phi)\}^2\right\} g_h(x-y)dy \\
&= \frac{1}{\beta}f(x|h,\theta,\psi) \\
&\quad + \frac{\gamma}{\beta h^2}(x-\mu)^T b(\phi)\int_{R(\theta,\psi)}(y-\mu)^T b(\phi) g_h(x-y)dy \\
&\quad - \frac{\gamma}{\beta h^2}\int_{R(\theta,\psi)}\{(y-\mu)^T b(\phi)\}^2 g_h(x-y)dy.
\end{aligned}$$

The derivative for orientation can be rewritten as

$$\begin{aligned}
&\frac{\gamma}{h^2}\int_{R(\theta,\psi)}(x-y)^T(d_\phi A)A^{-1}(y-\mu)g_h(x-y)dy \\
&\quad = \frac{\gamma}{h^2}\int_{R(\theta,\psi)}(x-\mu)^T(d_\phi A)A^{-1}(y-\mu)g_h(x-y)dy \\
&\qquad - \frac{\gamma}{h^2}\int_{R(\theta,\psi)}(y-\mu)^T(d_\phi A)A^{-1}(y-\mu)g_h(x-y)dy \\
&\quad = \frac{\gamma}{h^2}(x-\mu)^T(d_\phi A)A^{-1}\int_{R(\theta,\psi)}(y-\mu)g_h(x-y)dy - 0 \\
&\quad = \frac{\gamma}{h^2}(x_2-\mu_2)\int_{R(\theta,\psi)}(y_1-\mu_1)g_h(x-y)dy \\
&\qquad - \frac{\gamma}{h^2}(x_1-\mu_1)\int_{R(\theta,\psi)}(y_2-\mu_2)g_h(x-y)dy.
\end{aligned}$$

The derivative for location can be rewritten as

$$\begin{aligned}
&\frac{\gamma}{h^2}\int_{R(\theta,\psi)}(x-y)g_h(x-y)dy \\
&\quad = \frac{x-\mu}{h^2}f(x|h,\theta,\psi) - \frac{\gamma}{h^2}\int_{R(\theta,\psi)}(y-\mu)g_h(x-y)dy.
\end{aligned}$$

Once again, the derivative for intensity is given by

$$\frac{1}{\gamma}f(x|h,\theta,\psi).$$

Therefore, for calculation of the derivatives with respect to the cell specific parameters, five convolutions need to be computed for each cell. These equations can

be written as

$$\int_{R(\theta,\psi)} f(y) g_h(x-y) dy,$$

where $f(y)$ takes the following five forms: 1, $y_1 - \mu_1$, $y_2 - \mu_2$, $\|y-\mu\|^2$, and $\{(y-\mu)^T b(\phi)\}^2$. The cell image function $f(x|h,\theta,\psi)$ corresponds with $f(y) = \gamma$:

$$f(x|h,\theta,\psi) = \gamma \int_{R(\theta,\psi)} g_h(x-y) dy.$$

Note that a separate convolution for $(y-\mu)^T b(\phi)$ does not need to be computed because it is a linear combination of $y_1 - \mu_1$ and $y_2 - \mu_2$:

$$(y-\mu)^T b(\phi) = -(y_1 - \mu_1)\sin\phi + (y_2 - \mu_2)\cos\phi.$$

The derivative for the shape function, unlike those for the cell parameters, cannot be written as a combination of functions of the form $\int_{R(\theta,\psi)} f(y) g_h(x-y) dy$. Instead, a divided differences approximation to the derivative is used:

$$D_{\theta,\zeta} f_i(x|\theta) \approx \frac{f_i(x|\theta + d\zeta) - f_i(x|\theta)}{d},$$

where d is the minimum of $\zeta/10$ and ϵ, and ϵ is chosen to match machine precision.

3.5 *Additional Topics*

There are a couple of additional things that are not currently incorporated, but are still worth discussion. Accounting for nonconstant variance and selecting the smoothing parameter are both important topics in ill-posed inverse problems. Discussion of these matters is included as a precursor to future work.

3.5.1 Accounting for Nonconstant Variance

Recall from Chapter 2 that noise comes from two sources: photon counts and the detector. Photon count noise acts like Poisson noise with the variance proportional to the mean, say $\mathrm{Var}(z) = \sigma_p^2 \mathrm{E}(z)$, while detector noise acts like Gaussian noise with

the variance independent of the mean, say $\mathrm{Var}(z) = \sigma_d^2$. So the noise can be written as

$$\mathrm{Var}(z) = \sigma_p^2 \mathrm{E}(z) + \sigma_d^2.$$

The estimation method outlined in this chapter is probably optimal if the noise is dominated by the detector and thus acts like Gaussian noise. However, we know that in many cases the noise is dominated by the photon counts and acts like Poisson noise. Several methods are used in statistics to account for this nonconstant noise. Some of these methods are briefly outlined and discussed in this section.

The cell image model for intensity z at pixel k can be written as

$$z_k = f_k(\theta) + \epsilon_k,$$

where θ represents all of the parameters and f_k is the nonlinear functional describing the cell image model. For simplicity, in this discussion the infinite-dimensionality of the parameter space and regularization are ignored.

If the ϵ_k's are independent and identically distributed with mean zero and variance σ^2, then it is optimal to minimize the functional

$$\frac{1}{m} \sum_{k=1}^{m} \{z_k - f_k(\theta)\}^2.$$

The updating equation for the Gauss-Newton algorithm turned out to be

$$\hat{\theta} = (X^T X)^{-1} X^T z^*,$$

with $X = D_\theta f(\theta^0)$ and $z^* = z - f(\theta^0) + D_\theta f(\theta^0)\theta^0$. This is the scheme developed in this chapter.

If the ϵ_k's are independent with mean zero and variance $\sigma^2 f_k(\theta)$, then it might be good to try one of three methods: weighted least squares, transforming both sides, or transforming one side.

For the weighted model, the functional to minimize is

$$\frac{1}{m} \sum_{k=1}^{m} \{z_k - f_k(\theta)\}^2 w_k,$$

where $w_k^{-1} = f_k(\theta)$. Approximating w_k^{-1} with $f(\theta^0)$, the Gauss-Newton update is

$$\hat{\theta} = (X^T W X)^{-1} X^T W z^*,$$

where z^* and X are as before and $W = \text{diag}\{w_k\}$. This method is a version of iterated reweighted least squares (IRLS). The additional calculation for the weighted model then is just calculating $X^T W$ at each step.

The variance stabilizing transform could also be used. When the variance is proportional to the mean, this transform is first order equivalent to the square root. To see this, note that

$$z_k \sim (f_k(\theta), \sigma^2 f_k(\theta)).$$

Using a first order Taylor expansion, $\sqrt{z_k}$ can be approximated by

$$\sqrt{z_k} \approx \sqrt{f_k(\theta)} + \frac{1}{2\sqrt{f_k(\theta)}}(z_k - f_k(\theta)),$$

so that the mean and variance of the square root can be approximated to first order by

$$\begin{aligned} \text{E}(\sqrt{z_k}) &\approx \sqrt{f_k(\theta)} \\ \text{Var}(\sqrt{z_k}) &\approx \frac{1}{4f_k(\theta)}\text{Var}(z_k) = \frac{\sigma^2}{4}. \end{aligned}$$

The functional to minimize here is

$$\frac{1}{m}\sum_{k=1}^{m}\left\{\sqrt{z_k} - \sqrt{f_k(\theta)}\right\}^2.$$

The update is then

$$\hat{\theta} = (X^T X)^{-1} X^T z^*,$$

but now with

$$z^* = \sqrt{z_k} - \sqrt{f_k(\theta^0)} + \frac{1}{2\sqrt{f_k(\theta^0)}} D_\theta f_k(\theta^0)\theta^0$$

and

$$X = \frac{1}{2\sqrt{f_k(\theta^0)}} D_\theta f_k(\theta^0).$$

This should be quicker to compute at each iteration than the weighted model, but may or may not take more iterations to converge. These two methods are first order equivalent:

$$\begin{aligned}
&\sum_{k=1}^{m}\left\{\sqrt{z_k}-\sqrt{f_k(\theta)}\right\}^2 \\
&\quad\approx \sum_{k=1}^{m}\left\{\sqrt{f_k(\theta)}+\frac{1}{2\sqrt{f_k(\theta)}}(z_k-f_k(\theta))-\sqrt{f_k(\theta)}\right\}^2 \\
&\quad= \sum_{k=1}^{m}\frac{1}{4f_k(\theta)}\{z_k-f_k(\theta)\}^2 \\
&\quad\propto \sum_{k=1}^{m}\{z_k-f_k(\theta)\}^2 w_k.
\end{aligned}$$

Another method would be to just transform the data and not the model. Intuitively, it might seem like a way to use the original algorithm (perhaps with a different blurring parameter) with data that has constant variance. This method does not preserve the model, and so it is probably better to use one of the previously discussed methods.

Studies in other settings [74] suggest that improvements obtained by accounting for the nonconstant variance need to be weighed against cost in computation time. The work in this dissertation is a first attempt at fitting this type of model to cell image data, so throughout, the unweighted model is used and accounting for possible nonconstant variance is not pursued any further.

3.5.2 Selection of Smoothing Parameter

In the method of regularization setting, the value of the smoothing parameter can have a dramatic effect on the estimate. Therefore, it is important to have a method for selecting an appropriate smoothing parameter. One method that is often used to adaptively choose the smoothing parameter is called cross validation. The basic idea is to see which value of the smoothing parameter does the best job at predicting. The problem with this method in this nonlinear setting is the extent of computation

required for implementation. Therefore, an arbitrary selection method is used when fitting the model to real images. This is not the optimal thing to do, but if the guess is reasonable, the results should still be quite good. In the future, it would prove beneficial to develop a computationally reasonable method of selecting the smoothing parameter.

Chapter 4

EVALUATION OF THE CELL IMAGE MODEL

In this chapter, the properties of the cell image model are evaluated. Simulation studies were conducted to evaluate estimation error of the characteristic shape function, the cell specific parameters, the image, and functionals of interest. Shape function estimation is evaluated both when the cell parameters are known and estimated. Cell parameter estimation is evaluated when the shape function is known, held constant and estimated. Image recovery and functional estimation are evaluated when the cell parameters are known, when the shape function is known, when the shape function is held constant, and when both the cell parameters and shape function are estimated. Theoretical analysis is difficult for this model. Some ideas for a way to approach this are given.

4.1 Simulation Studies

Simulation studies are useful to characterize the estimation procedure when the true image is actually generated under the model. In this study, images were generated exactly like a "true" image but with varying amounts of noise. On each image, various parts of the estimation procedure were applied. This design enables evaluation of several aspects of the algorithm when the model holds.

The true image was simulated under the cell image model using the cell image simulator and is shown in Figure 4.1. The image area is a 128×128 grid of square pixels. There are 40 cells with characteristic shape function as shown in the bottom part of Figure 2.1 and their cell specific parameters generated as follows. For each

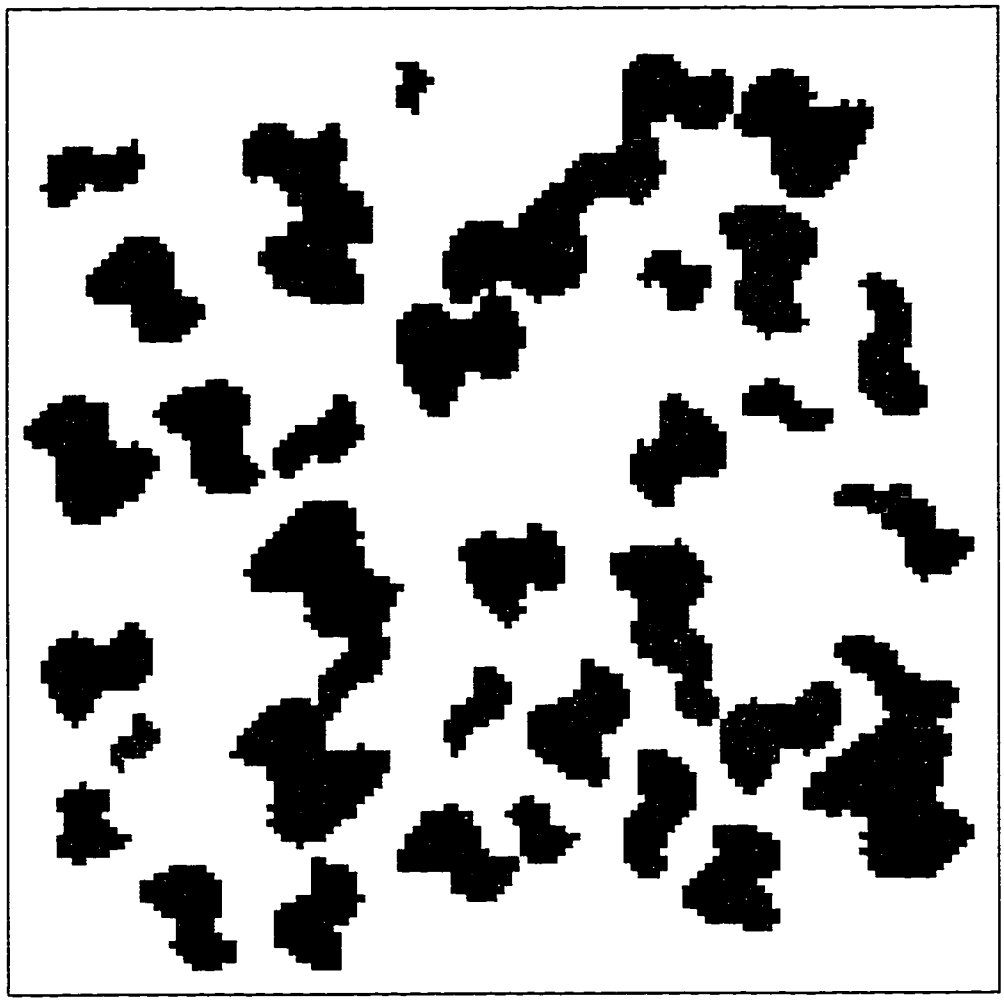


Figure 4.1: True image used for simulation study. See text for full description of its characteristics.

cell, location is in pixel units and generated from a bivariate uniform density on the image region and constrained so that the cell area is entirely on the image. Cells are allowed to touch and even to overlap slightly. Size is generated from the normal density with mean 8 and standard deviation 2 in pixel units. Aspect ratio is generated from the uniform density ranging from 0.1 to 0.4. Orientation is uniform between 0° and 360°. Intensity is normal with mean 255 and standard deviation 10.

The estimation procedure is evaluated with respect to two types of image noise: proportional noise with variance increasing proportionally with mean intensity of the

blurred image, and independent noise with variance constant over the image. In all cases, the full width at half maximum (FWHM) of the bivariate Gaussian blurring kernel is set at 3 pixels. To evaluate estimation in the presence of proportional noise, independent noise was held at zero and images were generated with proportional noise standard deviation varying from 1 up to 100. More explicitly, the noisy image z at pixel k is

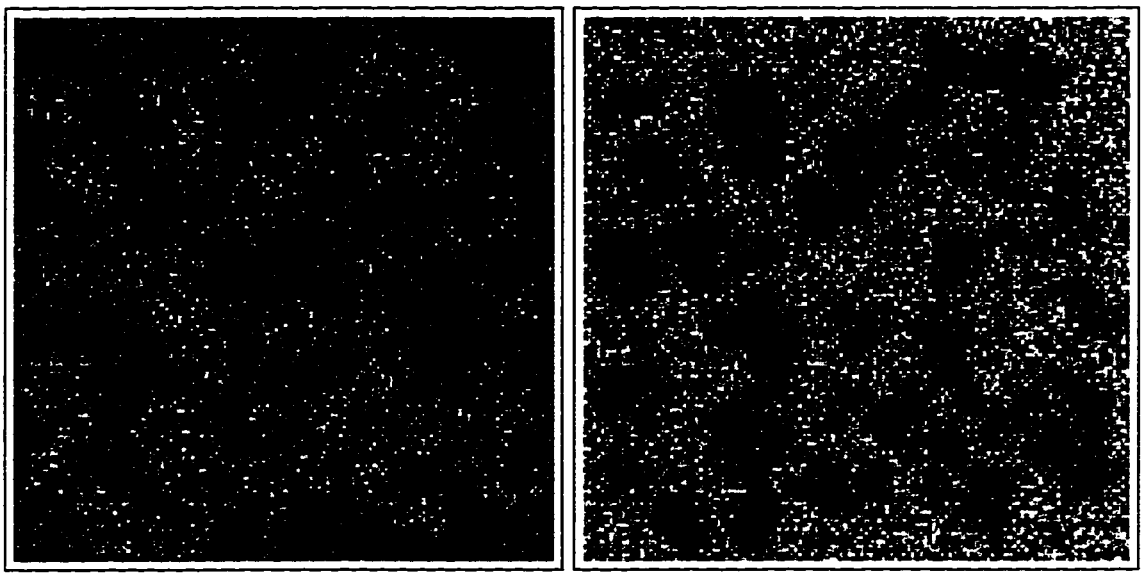
$$z_k = (f * g_h)_k + \sqrt{(f * g_h)_k} \times \epsilon_k,$$

where f_k is the value of the kth pixel of the true image, g_h is the Gaussian blurring kernel, $*$ is the convolution operator, and $\epsilon_k \sim N(0, \sigma^2)$ are independent and σ ranges from 1 to 100. To evaluate estimation in the presence of independent noise, proportional noise was held at zero and images were generated with independent noise standard deviation varying from 4 to 400, so that

$$z_k = (f * g_h)_k + \epsilon_k.$$

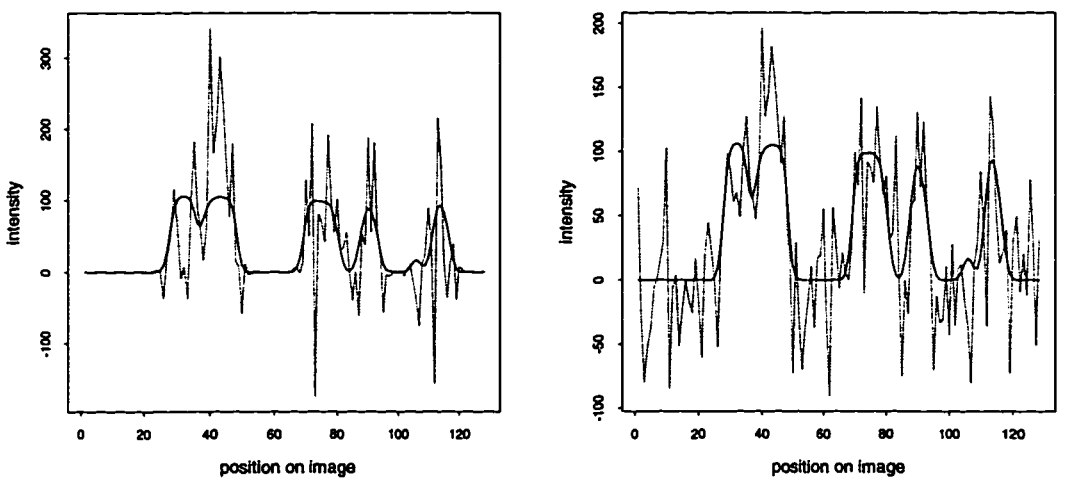
Where appropriate, the estimation procedure was also evaluated with respect to the number of cells on the image. To do this, proportional noise and independent noise were held at 1 and 25 respectively, and images were generated with anywhere from 1 to all 40 of the cells in the true image. The effect of number of cells was evaluated on estimation of the characteristic shape function and average values of the quantities of interest. For each type of noise, 21 different noise levels were chosen and one image was generated at each noise level. For the number of cells, one image was generated for each of 9 different number of cells.

Figure 4.2 shows two example images and transect profiles. Each of these images has more noise than what would normally be observed in fluorescence image cytometry; however, they serve well to contrast the two types of image noise. The transect profiles give a perspective on what the types of noise look like in a one dimensional problem. The effects of the two different types of noise on the image are quite different visually both in the images and transects.



(a) Proportional noise = 10

(b) Independent noise = 40



(c) Transect profile from (a)

(d) Transect profile from (b)

Figure 4.2: Example noisy images and transect profiles. Each image has noise level somewhat greater than what would usually be seen in fluorescence image cytometry. The transects are vertical running through the center of the image. The corresponding transect for the blurred image with no noise is included for reference.

Four different estimation procedures were applied to each image. The shape function was estimated with the cell parameters known exactly. The cell parameters were estimated both with the shape function known exactly and with it set as a constant. The cell parameters and the shape function were also estimated together in a two step manner, where the cell parameters were estimated and then the shape function was estimated.

For each of the methods involving estimation of the shape function, the initial value was the constant function corresponding to an ellipse. The cell parameters location, size and aspect ratio were initialized using the S function discussed in Chapter 3. This was done on the true image and these values were used for all images. Orientation and intensity were initialized at 0 and 255 respectively. Figure 4.3 shows the estimated image with all parameters at the initial value.

The smoothing parameter was optimized by minimizing the squared difference between the estimated shape function and the true shape function. That is, the smoothing parameter λ was chosen to minimize root integrated squared error (RISE):

$$\sqrt{\frac{1}{2\pi}\int_0^{2\pi}\left\{\hat{\theta}_\lambda(t)-\theta(t)\right\}^2},$$

where $\hat{\theta}_\lambda$ is the estimated shape function when the smoothing parameter is λ and θ is the true shape function. In the case when the cell parameters were estimated, RISE between the estimate rotated 180° and the true curve was also computed, and the minimum of these two values was used. This was done because 180° rotations are indistinguishable when the cell parameters are estimated with a constant shape function. Figure 4.4 illustrates the difference between the true shape function and the same function rotated 180°. It is clear that 180° rotations have a dramatic effect.

The design of this simulation study enables evaluation of several different aspects of the estimation algorithm in terms of consistency, rate of estimation and bias. The characteristic shape function estimation can be evaluated both when the cell specific parameters are known and when they are estimated. Estimation of the cell

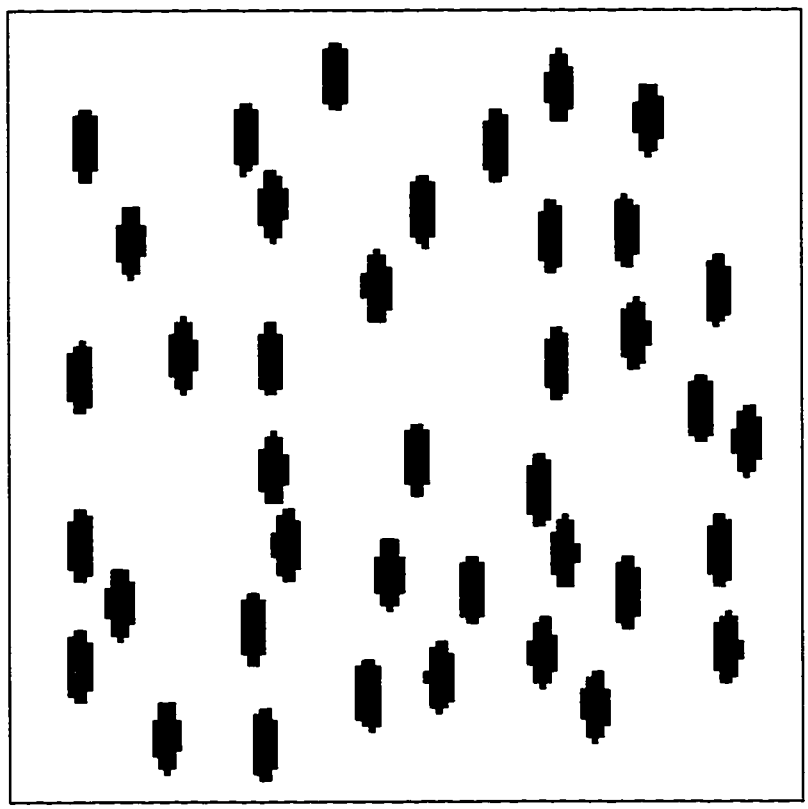


Figure 4.3: Estimated image with all parameters set at the initial values.

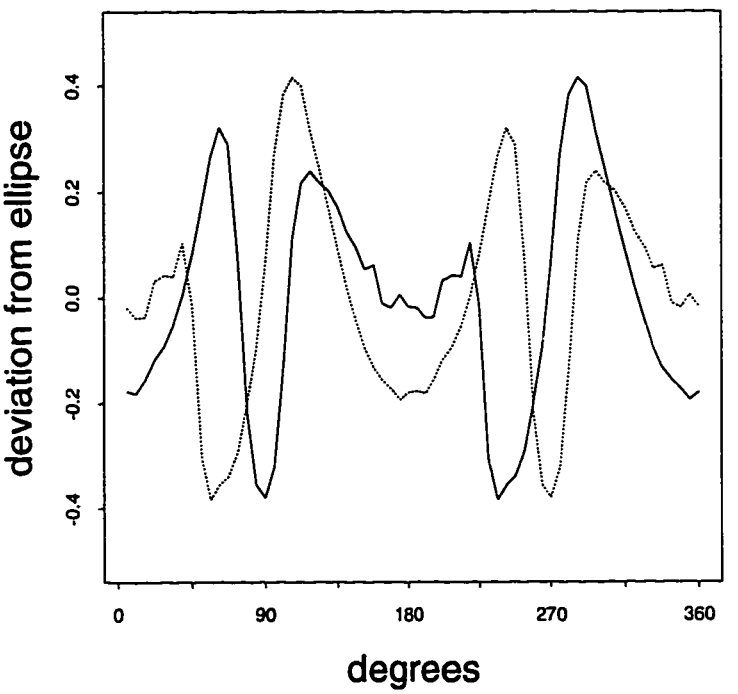


Figure 4.4: Effect of 180° rotation on the true shape function. The solid line is the true shape function and the dotted line is the same function rotated 180°.

parameters can be evaluated when the shape function is correct and misspecified as a constant. Also, estimation of the functionals of interest and image recovery can be evaluated for each of the estimation procedures. This shows which parameters in the model are more difficult to estimate, and perhaps sheds some light on the degree of interdependence between the two parts of the estimation procedure.

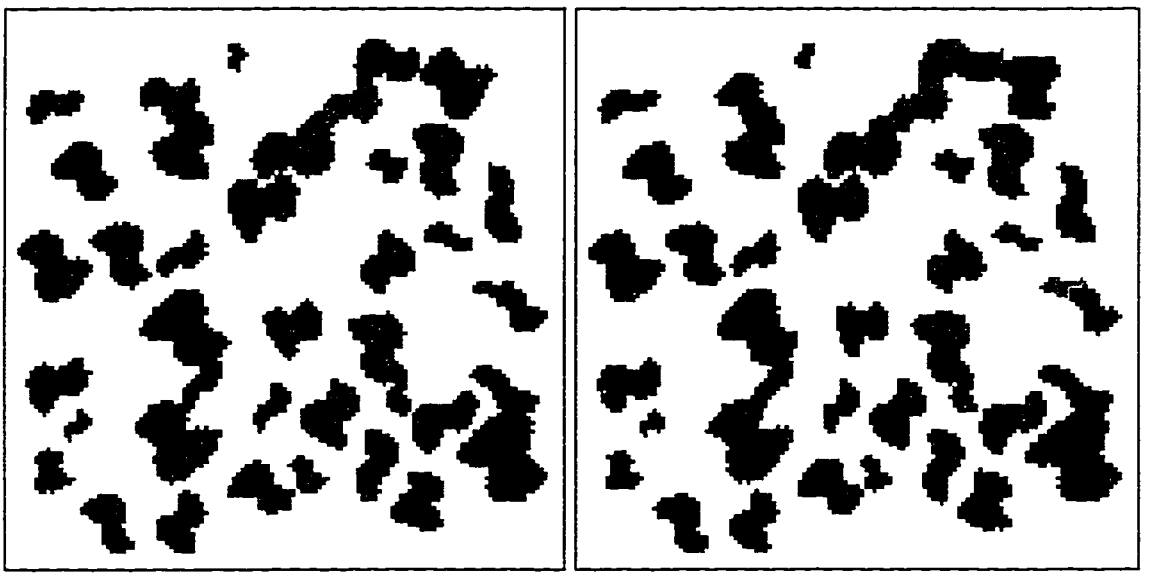
4.2 Recovering the Image

In image analysis it is interesting to see how well the estimation procedure recovers the true image. Estimated images for the four estimation procedures when the independent image noise is 40 are illustrated in Figure 4.5. The estimated image appears best when either the shape function or cell parameters are known, with a slight advantage when the cell parameters are known. The two step procedure is a refinement of the estimate with the shape function misspecified and visually improves the estimate. Residual images were made (the absolute value of the difference between the estimated image and the true image) for each case and are illustrated in Figure 4.6. In the residual images, the estimate appears best in the case when the cell parameters are known. The case when the shape function is known appears better than the two step case, which looks a little better than the case when the shape function is assumed to be constant.

For each estimation procedure applied to the simulated image, Figure 4.7 illustrates the root mean squared error (RMSE) of the image estimates plotted against each type of noise. RMSE is defined to be

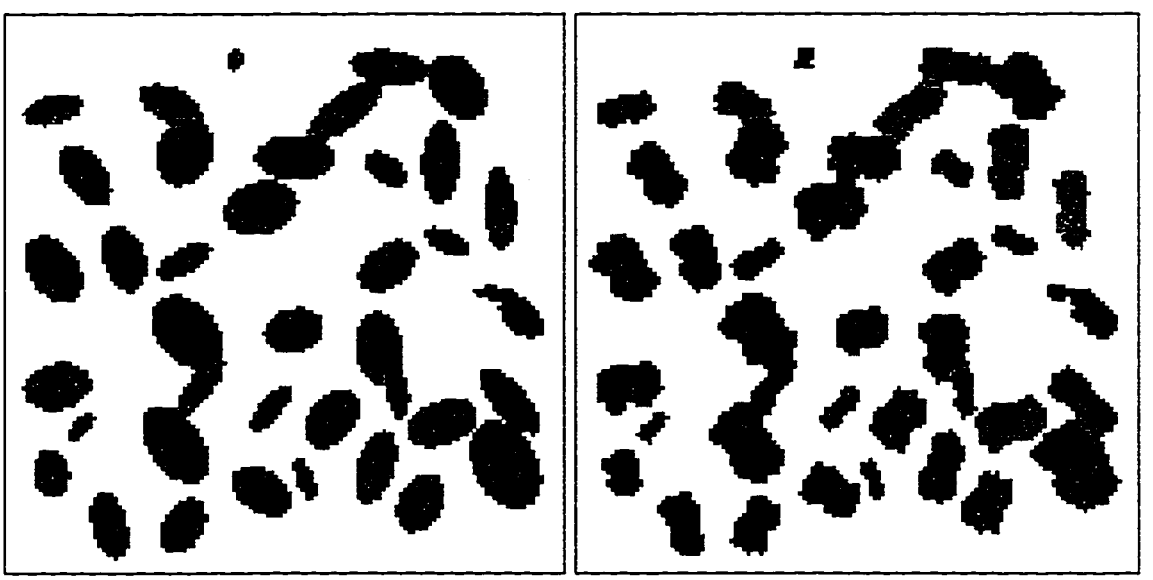
$$\sqrt{\frac{1}{m}\sum_{k=1}^{m}(\hat{f}_k - f_k)^2}\,,$$

where $\hat{f}_k$ and f_k are respectively the estimated and true image at pixel k, and m is the total number of pixels. In the plots these are divided by the RMSE of the true



(a) Cell parameters known

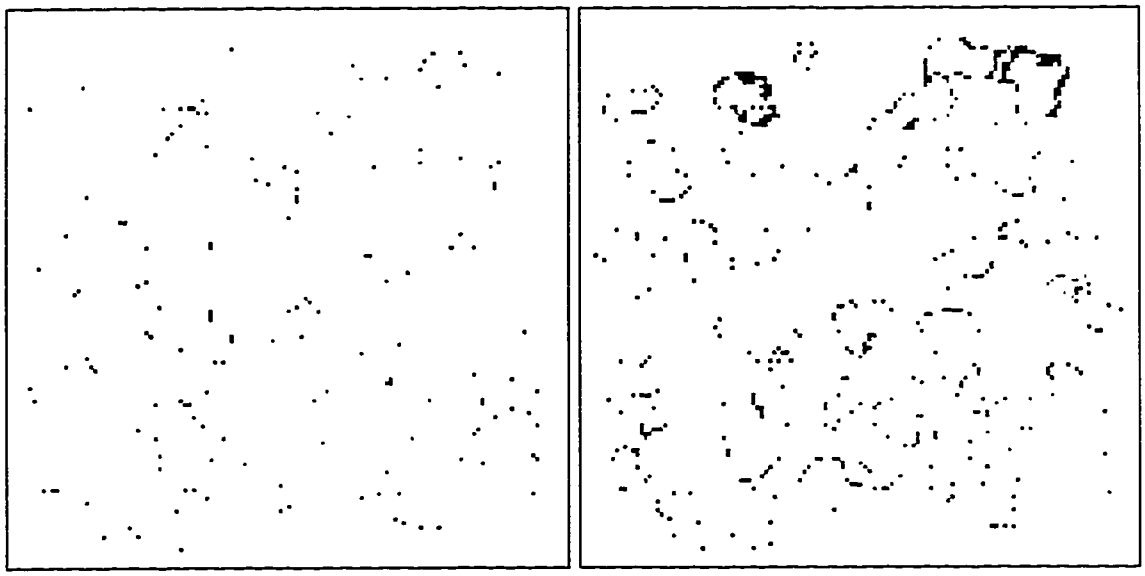
(b) Shape function known



(c) Shape function misspecified

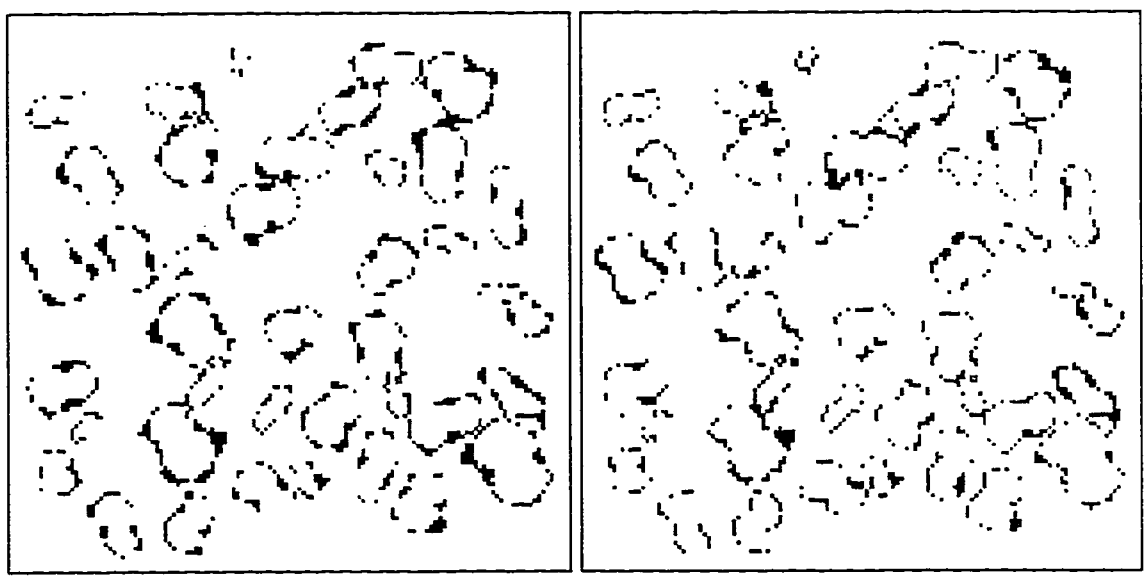
(d) Shape and parameters estimated

Figure 4.5: Recovered images for the four different estimation methods studied when the data is the image in Figure 4.2(b).



(a) Cell parameters known

(b) Shape function known



(c) Shape function misspecified

(d) Shape and parameters estimated

Figure 4.6: Recovered residual images for the four different estimation methods studied when the data is the image in Figure 4.2(b).

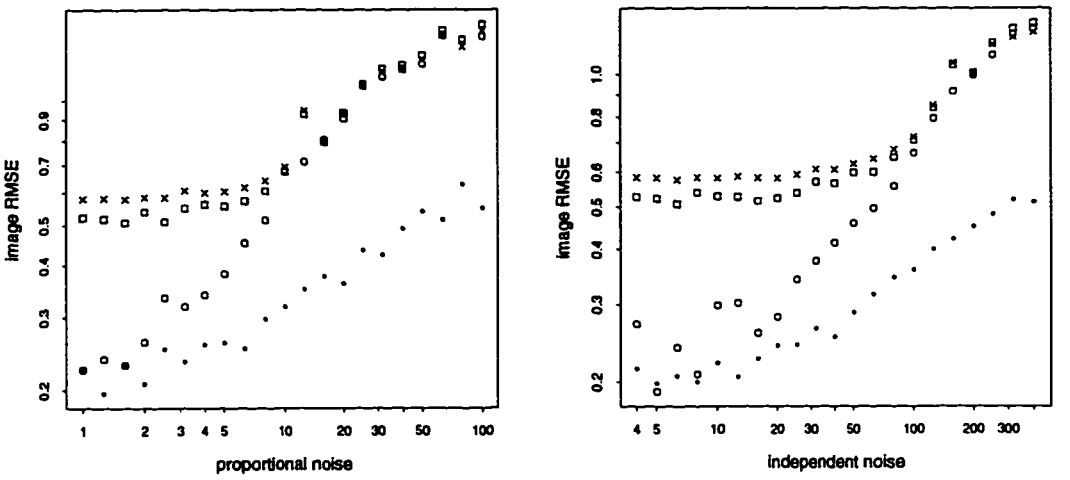


Figure 4.7: Image root mean squared error (RMSE) vs. image noise. The plots are on a log-log scale. The symbols are as follows: (·) estimation with the cell parameters ψ known, (o) estimation with the shape function θ known, (×) estimation with the shape function θ misspecified, and (□) estimation of both the shape function and the cell parameters.

image

$$\sqrt{\frac{1}{m}\sum_{k=1}^{m}(f_k - \bar{f})^2}\,,$$

where $\bar{f} = (1/m)\sum_{k=1}^{m} f_k$, to provide a meaningful scale. A scaled RMSE of 1, for example, means that the estimated image varies from the true image as much as the true image varies within itself.

When image noise is low, image RMSE is about the same whether the cell parameters or the shape function is known, and is about two or three times higher either when the shape function is misspecified or estimated in a two step manner. Also, the two step procedure provides about a 10 percent improvement in RMSE over the case when the shape function is assumed to be constant. When image noise is high, image RMSE is about two or three times as high when the cell parameters are estimated than when they are known. This implies that image recovery in the presence of large

amounts of noise is more sensitive to cell parameter estimation than shape function estimation. Also, the impact on image recovery of not knowing the cell parameters is more substantial than the impact of not knowing the shape function.

The plots in Figure 4.7 are all on a log-log scale and appear approximately linear for the cases when either the cell parameters or the shape function is known. This coincides with related situations [22, 67, 95, 96] in that

$$MSE \approx \kappa\sigma^{2\tau},$$

where σ is the noise level and κ and τ can be estimated by linear regression. Note that $\tau = 1$ in the parametric case. The rate of estimation with the cell parameters known was estimated to be 0.25 ± 0.01 for proportional noise and 0.23 ± 0.01 for independent noise. The rate of estimation with the shape function known was estimated to be 0.47 ± 0.02 for proportional noise and 0.42 ± 0.02 for independent noise. These rates are consistent with what might be expected in a nonparametric situation. Rate of estimation is similar for each type of noise. The rate is about twice as high when the shape function is known than when the cell parameters are known. This indicates that, in the range of noise studied, the rate of image estimation is closest to parametric when the shape curve is known. For the cases when neither the cell parameters nor the shape function are assumed known, the plots appear nonlinear on the log-log scale. This is probably because bias is dominating in these cases at low noise levels.

The algorithm appears to recover the image quite well even in the presence of noise and estimation behaves as would be expected for a method of regularization procedure. This gives hope that the cell parameters, the shape function and the functionals of interest will also be reasonably well estimated. This is investigated in the next three sections.

4.3 *Characteristic Shape Function*

Figure 4.8 illustrates the true shape function and the estimated shape functions when the data is the image in Figure 4.2(b). The two estimates are for the cases when the cell parameters are known and estimated. When the cell parameters are known, the estimated shape curve is quite close to the true curve. When the cell parameters are estimated, the shape curve estimate is not excellent; however, it is reasonable except for the area between about 220° and 300°. This is largely due to the fact that the orientation for some of the cells is off by 180° and the curve is not invariant to 180° rotations. The fact that a large portion of the curve is estimated reasonably well provides hope that a full iterative procedure would do even better.

Figure 4.9 illustrates plots of shape function root integrated squared error (RISE) against proportional noise, independent noise and number of cells. RISE is defined to be

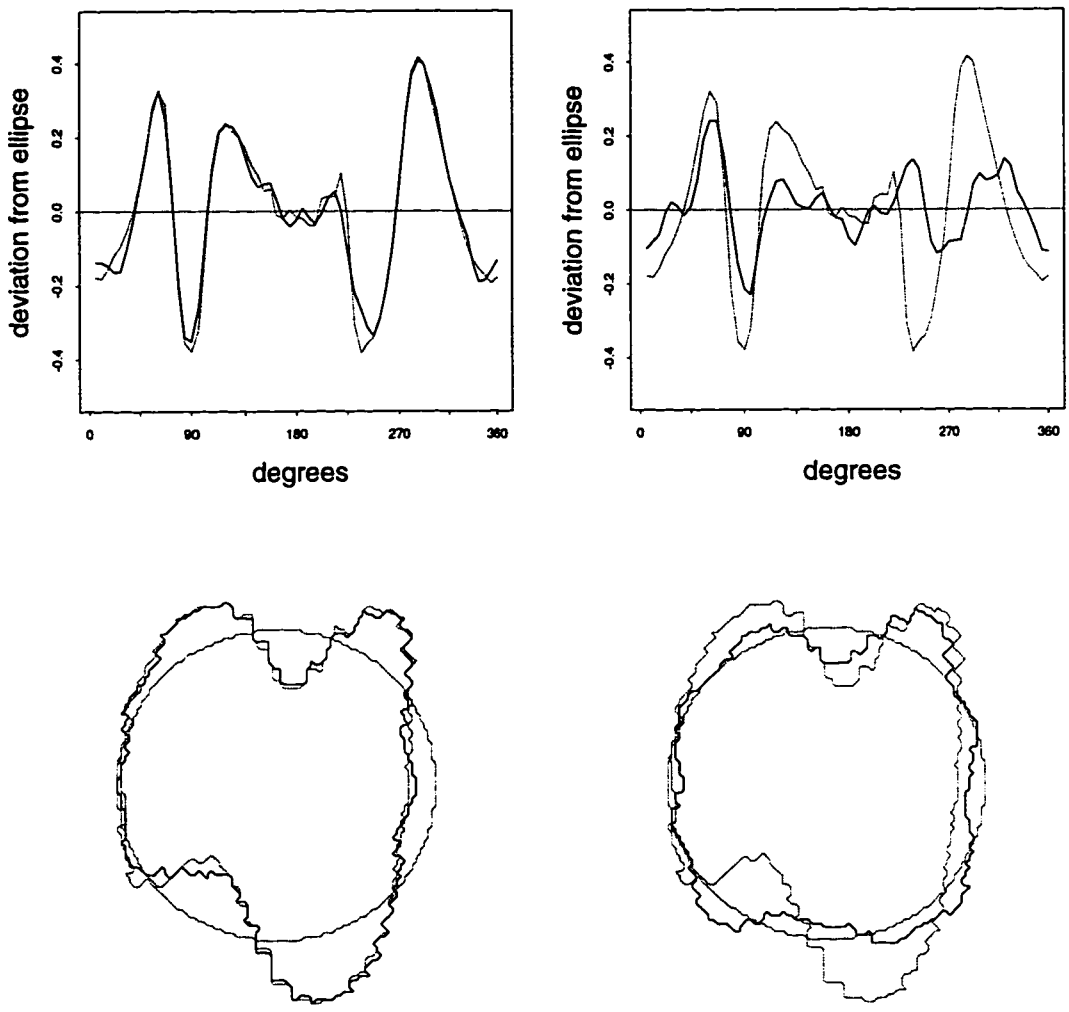
$$\sqrt{\frac{1}{2\pi}\int_0^{2\pi}\left\{\hat{\theta}(t)-\theta(t)\right\}^2 dt},$$

where $\hat{\theta}(t)$ and $\theta(t)$ are respectively the estimated and true values of the shape curve at t. RISE was also calculated with $\hat{\theta}$ rotated 180° and the minimum of these two values was used. In the plots these are divided by

$$\sqrt{\frac{1}{2\pi}\int_0^{2\pi}\left\{\theta(t)-\bar{\theta}\right\}^2 dt},$$

where $\bar{\theta} = (1/2\pi)\int_0^{2\pi}\theta(t)dt$, to provide a meaningful scale. For example, a scaled RISE of 1 means that the estimated shape function varies from the true shape function as much as the true function varies from the constant function corresponding to an ellipse.

When image noise is low or the number of cells is high, RISE is about 5 or 6 times as high when the cell parameters are estimated compared to when they are known. When image noise is high or the number of cells is low, RISE is similar for the two cases, although definitely lower when the cell parameters are known. The



(a) cell parameters known

(b) cell parameters estimated

Figure 4.8: True and estimated curves when the data is the image in Figure 4.2(b). In each case the solid line is the estimate and the dotted line is the truth. The constant function is also given as a dotted line for reference.

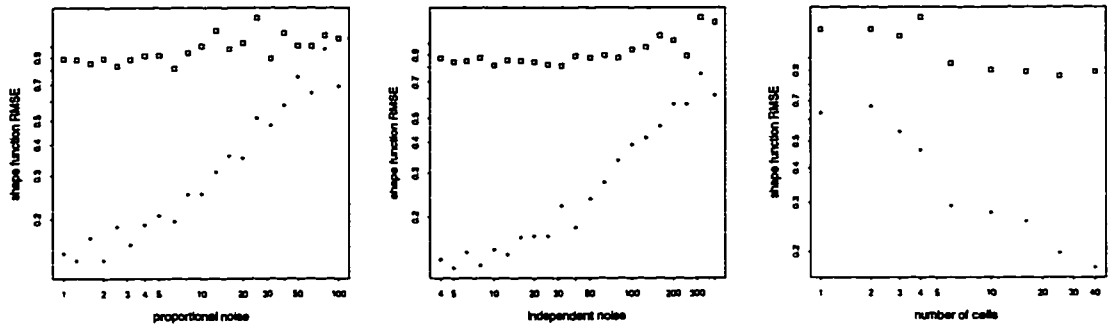


Figure 4.9: Characteristic shape function root integrated squared error (RISE) vs. image noise and number of cells. The plots are on a log-log scale. The symbols are (·) when the cell parameters ψ known and (□) when the cell parameters are estimated.

plots in Figure 4.9 all appear approximately linear on the log-log scale. Rates of estimation were estimated by linear regression. When the cell specific parameters are known, the rate of estimation with respect to proportional noise, independent noise and number of cells is estimated by linear regression to be 0.42 ± 0.02, 0.40 ± 0.02 and 0.40 ± 0.04 respectively. The rates are similar for each type of noise and number of cells, indicating that shape function estimation converges at about the same speed with respect to each of these aspects. When the cell specific parameters are estimated, the rate of estimation with respect to proportional noise, independent noise and number of cells is estimated by linear regression to be 0.06 ± 0.01, 0.07 ± 0.01 and 0.13 ± 0.03 respectively. These rates are much lower, indicating that the quality of the cell parameter estimates has a large effect on shape function estimation.

4.4 Cell Specific Parameters

Figures 4.10 and 4.11 show plots of cell parameter root mean squared error (RMSE) against proportional and independent noise. For all parameters except orientation,

RMSE is defined to be

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\psi}_i - \psi_i)^2},$$

where $\hat{\psi}_i$ and ψ_i are respectively the estimated and true cell parameter (either location (x), location (y), size, aspect ratio or intensity) and n is the number of cells. In the case of orientation, RMSE is defined to be

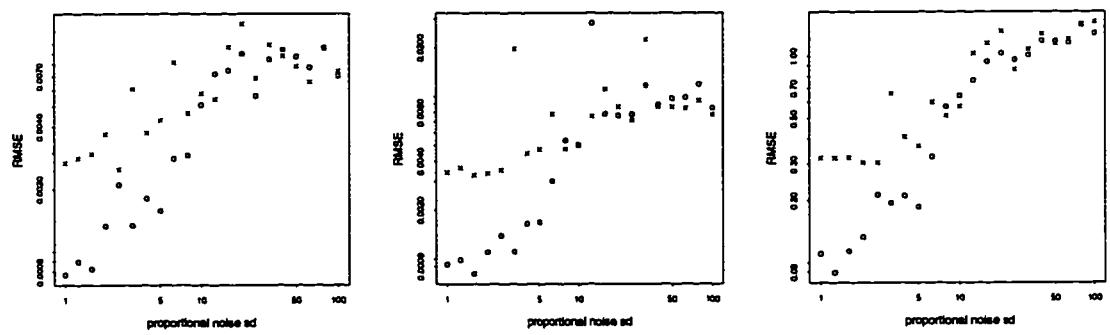
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\{(\hat{\phi}_i - \phi_i + 90°) \bmod 180° - 90°\}^2},$$

where mod is the modulo operator. This metric is used because 90° is the greatest relevant difference when the shape function is constant, since 180° rotations are identical for ellipses This allows a more fair comparison between the methods when the shape function is known and assumed constant. RMSE is scaled differently for each parameter. For location the scale factor is $\sqrt{(d_1^2 + d_2^2)/2}$, where d_1 and d_2 are the image dimensions. For size, aspect ratio and intensity the scale factor is

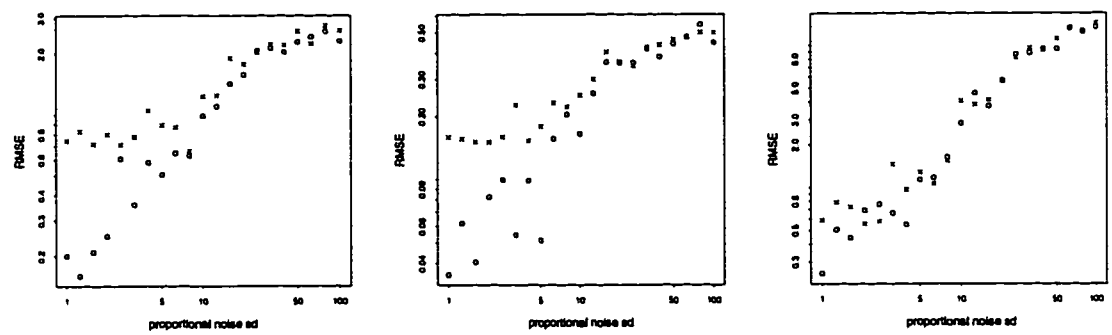
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\psi_i - \bar{\psi})^2},$$

where $\bar{\psi} = (1/n)\sum_{i=1}^{n}\psi_i$ and ψ_i is the appropriate parameter for cell i. For orientation the scale factor is simply 90°. A scaled RMSE value of 1 for location means that the average estimate is off by the distance across the image, for orientation it means the average estimate is off by 90°, and for the rest of the parameters it means the estimates vary from the truth as much as the true parameters vary among themselves. Note that since the two step algorithm is a refinement of the case when the shape function is misspecified, the cell parameter estimates are identical for these two cases.

Cell parameter RMSE shows the same basic pattern for both types of image noise. At high noise levels, RMSE for location, size, aspect ratio and orientation are similar whether the shape curve is known or assumed to be constant; however, at low noise levels, RMSE is much lower when the shape function is known. This parallels the situation for image recovery and indicates that part of the improvement

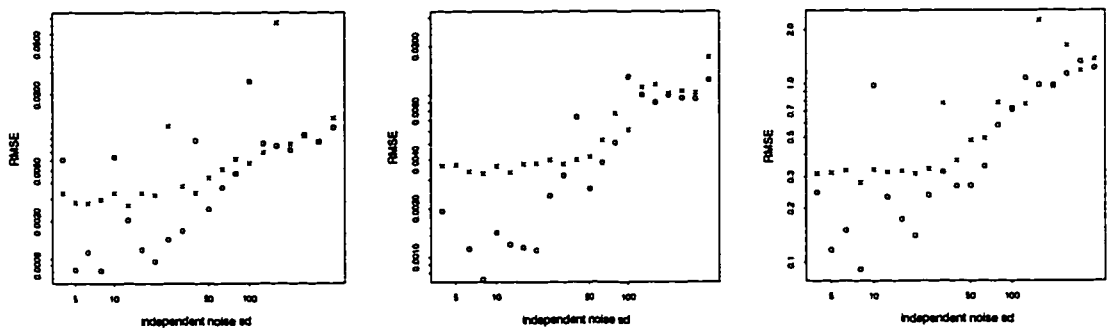


(a) X location (μ_1) (b) Y location (μ_2) (c) Size (α)

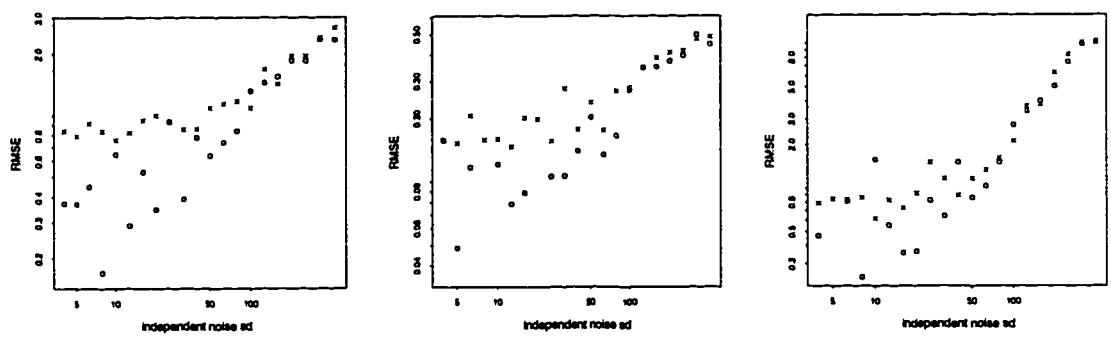


(d) Aspect ratio (β) (e) Orientation (ϕ) (f) Intensity (γ)

Figure 4.10: Cell specific parameter root mean squared error (RMSE) vs. proportional image noise. The plots are on a log-log scale. The symbols are (o) when the shape function is known and (×) when the shape function θ misspecified.



(a) X location (μ_1) (b) Y location (μ_2) (c) Size (α)



(d) Aspect ratio (β) (e) Orientation (ϕ) (f) Intensity (γ)

Figure 4.11: Cell specific parameter root mean squared error (RMSE) vs. independent image noise. The plots are on a log-log scale. The symbols are (o) when the shape function is known and ($\times$) when the shape function θ misspecified.

Table 4.1: Rate of Cell Specific Parameter Estimation from linear regression on the data shown in Figures 4.10 and 4.11. Standard errors ranged from 2 to 12.

noise type	θ assumption	μ_1	μ_2	α	β	ϕ	γ
proportional noise	known	0.59	0.66	0.65	0.63	0.61	0.91
	constant	0.28	0.24	0.39	0.34	0.31	0.81
independent noise	known	0.49	0.65	0.50	0.47	0.49	0.76
	constant	0.37	0.34	0.40	0.24	0.26	0.59

in image recovery when the shape function is known can be attributed to improved cell parameter estimates. RMSE for intensity is nearly indistinguishable between the two methods, indicating that estimation of the intensity parameters is not much affected by knowledge of the shape function.

All of the RMSE plots for the cell parameters appear approximately linear on the log-log scale for each estimation situation, and the rate of estimation is estimated by linear regression. These estimates are given in Table 4.1. Rate of intensity estimation is highest for all the situations, probably due to the linearity of the parameter. The rates are substantially higher when the shape function is known in accordance with the figures. In the case when the shape function is known, the rate of estimation is slightly higher in the proportional noise case. In all cases, the rate of cell parameter estimation is higher than the rate for image recovery. Some of the plots for the case when the shape function is misspecified appear a bit nonlinear on the log-log scale. This is probably because bias dominates at lower noise levels.

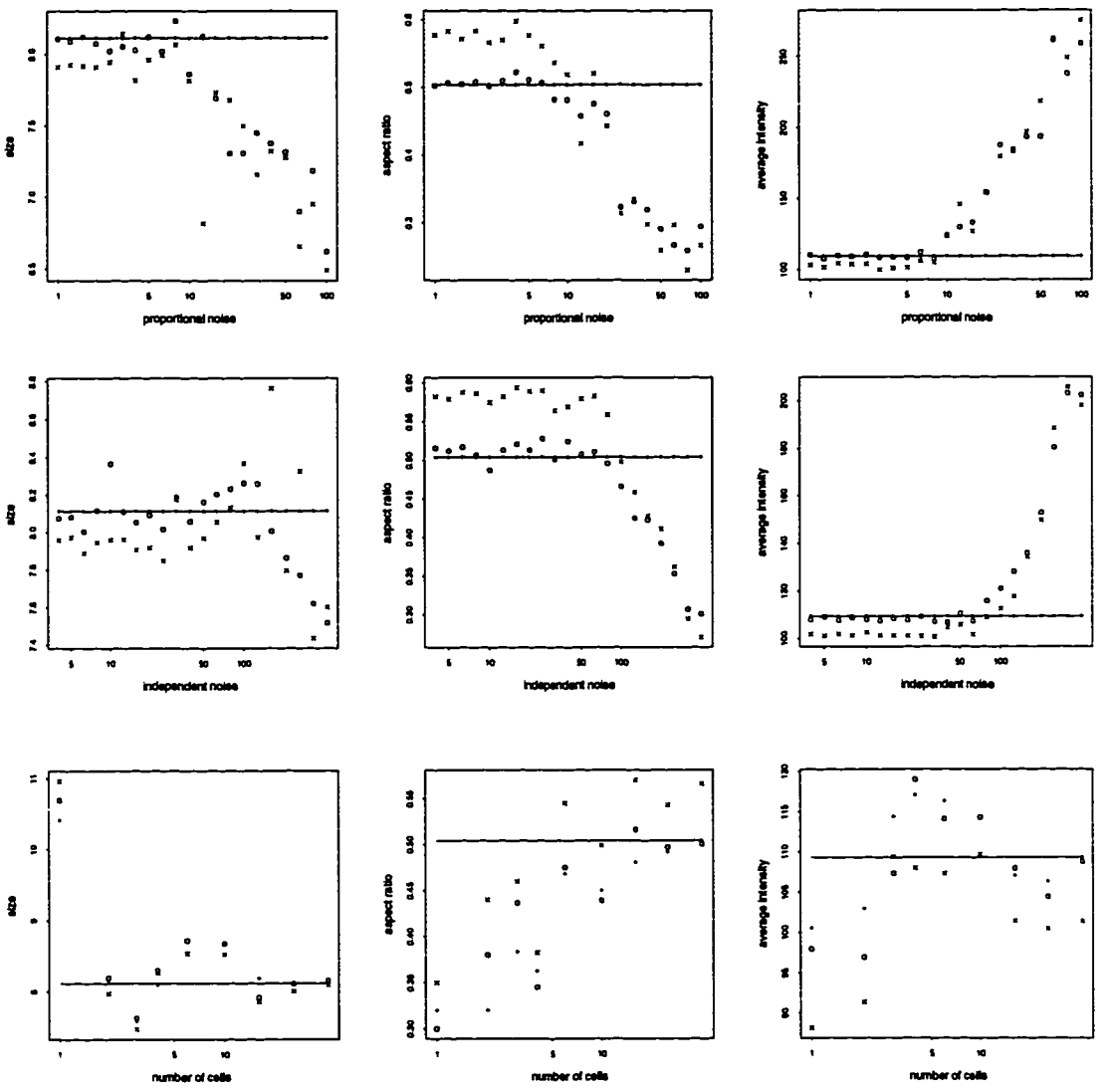
These results indicate that cell specific parameter estimation behaves well with respect to proportional noise and independent noise. Estimation is better when the shape function is known, indicating that the two parts of the algorithm are inter-dependent and an iterative procedure might perform even better than the two step procedure.

4.5 *Quantities of Interest*

For inference in cytometry, certain functionals of the shape function and cell parameters are often used. Some commonly used functionals in cytometry are area, perimeter, shape factor, average intensity and integrated intensity. Estimation of average functional values is evaluated with respect to each type of image noise and number of cells.

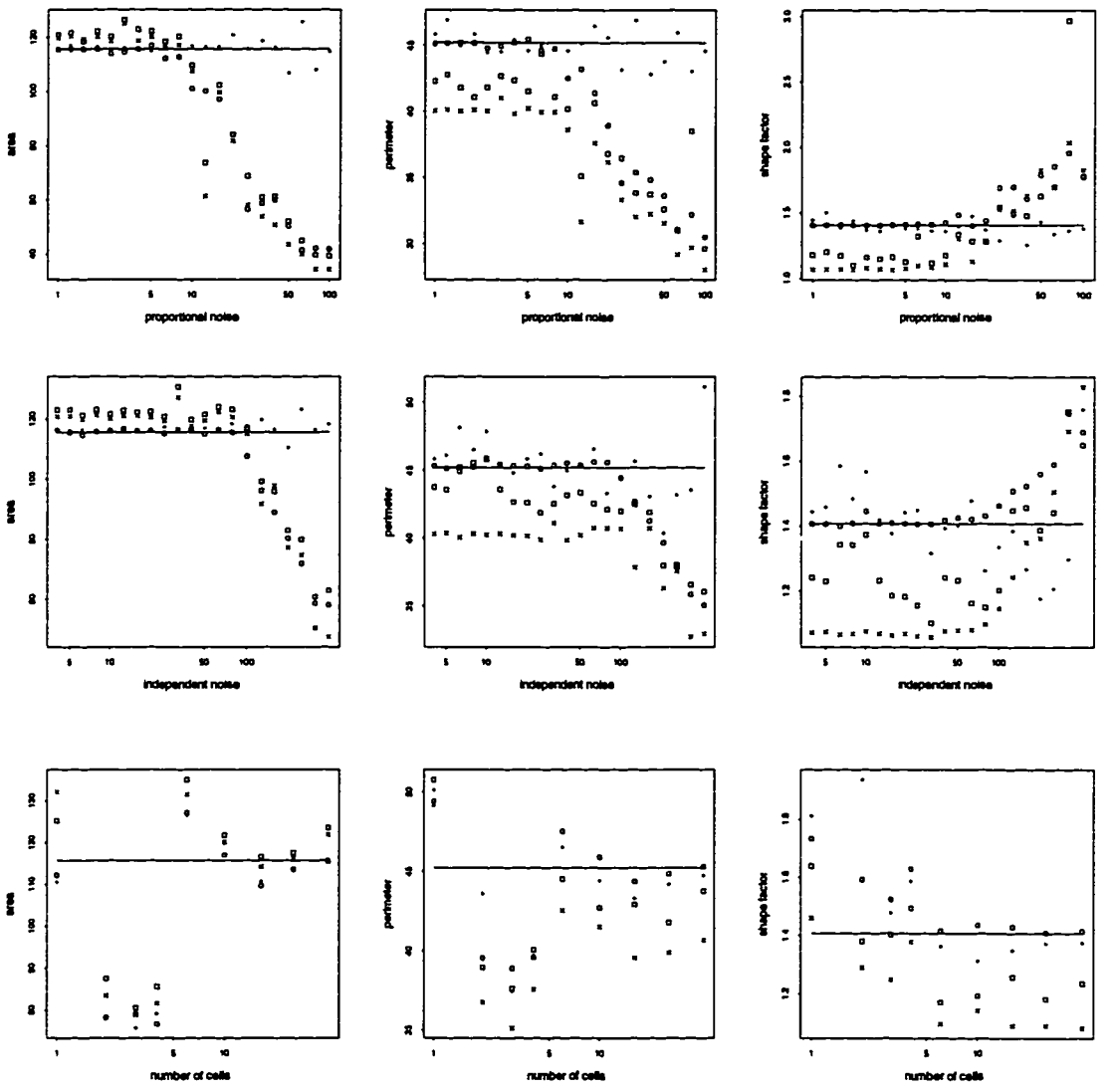
Figures 4.12, 4.13 and 4.14 give plots of estimates of the average quantities against each type of image noise. A line is drawn on each plot at the true mean value of the functional. The patterns for the two types of image noise are very similar. For all of the functionals, the estimates are best in the case when the cell parameters are known. (This is true by definition for the average cell parameters size (α), aspect ratio (β) and intensity (γ), which are known exactly in this case.) There is little if any bias for any of the functionals, and the variability is relatively small, with some increase with noise. For the cases when the cell parameters are estimated, a threshold appears in the plots at about 10 for proportional noise and about 100 for independent noise. Below the threshold, the bias is fairly constant. Once above the threshold, the bias increases with noise.

Below the threshold, the effect of assumed shape function is large. The estimates when the shape function is known are basically unbiased and about as good as when the cell parameters are known; however, the estimates when the shape function is assumed constant have about the same variability, but are biased. If the shape function is misspecified to be a constant, then size is biased slightly downward, aspect ratio upward, area slightly upward, perimeter dramatically downward, and average intensity slightly downward, causing shape factor to be biased dramatically downward and integrated intensity slightly downward. It is not clear why area and integrated intensity are biased; however, it does make sense that the biases are in opposite directions because underestimating average intensity can be compensated for somewhat by in-



(a) Size (b) Aspect ratio (c) Intensity

Figure 4.12: Average cell parameter estimates vs. proportional noise, independent noise and number of cells. The parameters are size (α), aspect ratio (β) and intensity (γ) from the model. The symbols are ($\cdot$) when the cell parameters are known, (o) when the shape function is known, ($\times$) when the shape function is misspecified as constant or both the shape function and the cell parameters are estimated. The solid line indicates the true mean value.



(a) Area (b) Perimeter (c) Shape factor

Figure 4.13: Average functional estimates vs. proportional noise, independent noise and number of cells. The functionals are area, perimeter and shape factor as defined in the text. The symbols are (·) when the cell parameters are known, (o) when the shape function is known, (×) when the shape function is misspecified as constant, and (□) when both the shape function and the cell parameters are estimated. The solid line indicates the true mean value.

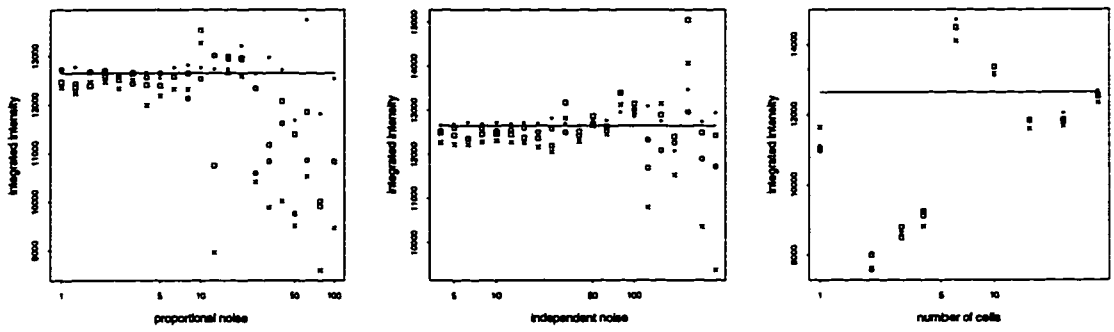


Figure 4.14: Average integrated intensity estimates vs. proportional noise, independent noise and number of cells. The symbols are (·) when the cell parameters are known, (o) when the shape function is known, (×) when the shape function is misspecified as constant, and (□) when both the shape function and the cell parameters are estimated. The solid line indicates the true mean value.

creasing cell size and thus average area. The large downward bias in perimeter makes sense because, for all objects of a given area, ellipses have the smallest perimeter. Therefore, as long as size is well estimated, perimeter should be underestimated if the shape function is misspecified as a constant. The fact that perimeter is a function of cell size may, or may not, have something to do with why area is slightly overestimated. For the two step procedure, the cell parameter estimates are identical to the case when the shape function is misspecified as a constant. The estimates are slightly more biased, but the bias in perimeter, shape factor and integrated intensity is reduced by almost a half.

Above the threshold, the pattern is similar for all quantities except integrated intensity. Bias increases dramatically and variability increases slightly with noise and the shape function specification appears to have little effect. For integrated intensity, bias increases slightly while variability increases dramatically. The bias for area, perimeter and integrated intensity is downward, while for shape factor and average intensity it is upward. A possible explanation is that intensity peaks due to noise cause average intensity to be overestimated, resulting in lower estimates for area. The

bias in area is large enough to bias shape factor upward, even though perimeter is biased downward. The bias is also a bit larger for area than average intensity, causing integrated intensity to be biased downward.

These results indicate that knowing the characteristic shape function most dramatically improves estimation of morphological measurements such as perimeter and shape factor, but also improves estimation of measurements such as area and intensity. It is also apparent that the two step estimation procedure improves average functional estimation when compared to the case when the shape function is held constant. This gives promise that a fully iterated procedure might perform even better.

The biases in the functionals indicate a great deal of interdependence between the various parameters and the possible presence of unidentifiability. The biases are only apparent when the image noise is much higher than what is normally seen in confocal fluorescence microscopy. Even so, it would be interesting to study the sources of these biases.

4.6 Thoughts on Theory

The estimation procedure for the cell image model described in this dissertation is a nonlinear method of regularization (MOR) procedure. Theoretical analysis for this class of estimators has proven to be rather difficult and is outside the scope of this dissertation. This section reviews some work related to this problem and attempts to motivate future work on this type of analysis. A factor that compounds the problem for the cell image model is that the estimate includes not only a nonparametric estimate of the shape curve, but six parameters (five of them nonlinear) for each cell. Discussion in this section is restricted to the problem of estimating the shape curve when the cell parameters are known.

Recall from Section 3.1 the objective functional being minimized:

$$l_{n,m,\lambda,h}(\theta) = \frac{1}{m}\sum_{k=1}^{m}\left\{z(x_k) - \sum_{i=1}^{n} f_i(x_k|h,\theta)\right\}^2 + \lambda\int_0^{2\pi}\{\theta''(t)\}^2 dt.$$

The penalty functional being used makes it convenient to consider minimizing the objective functional over the set of curves θ in the Sobolev space $W_2^2[0,2\pi]$ given by

$$\theta \in W_2^2[0,2\pi] \text{ if } \theta \text{ and } \theta' \text{ are absolutely continuous and } \int_0^{2\pi}\{\theta''(t)\}^2 dt < \infty.$$

If the inner product for $\theta, \phi \in W_2^2[0,2\pi]$ is given by

$$< \theta, \phi >= \int_0^{2\pi}\theta(t)\phi(t)dt + \int_0^{2\pi}\theta''(t)\phi''(t)dt,$$

then $W_2^2[0,2\pi]$ is a Hilbert space.

There are many results for linear problems. It can be shown [72, 104] that the MOR estimator minimizes both average and maximum mean squared error if the smoothing parameter is known. There are methods for computing bias and variability in certain situations [56, 72]. Methods for obtaining approximate confidence intervals have been developed for spline models [65, 104]. Consistency results and asymptotic rates of estimation for linear MOR estimators have been obtained in a variety of settings [24, 43, 67, 91, 95, 96].

The model considered in this dissertation is nonlinear and none of these results directly applies. It has been proposed [25, 73] to analyze nonlinear MOR estimators by linearizing and analyzing the linearized estimator. Consistency results and rates of estimation have been obtained [25, 73] for a general class of nonlinear MOR estimators. Much work remains to be done in this area, including extending bias, variability and confidence interval calculations to the nonlinear case. Once again, this is out of the scope of the present work and mentioned only to provide a basic assessment of the situation and to hopefully stimulate interest.

Chapter 5

AN ILLUSTRATION: TANGIER DISEASE

Image data from a Tangier disease experiment [70] are used to illustrate the methods developed in this dissertation. The mechanisms related to Tangier disease provide clues to help understand some of the causes of cardiovascular disease. Some of these mechanisms may be manifested by changing the shape of certain cells in the body. Background on cholesterol, Tangier disease and their relationship is given. The image data was analyzed using a standard method. This method is described and the results for a subset of the images are reported. These images are reanalyzed using the estimation procedure described in Chapter 3. The results are given as well as a comparison of the two methods. The image that was used to illustrate segmentation methods in Chapter 1 is also analyzed using both methods.

5.1 Tangier Disease

Tangier disease is characterized by deficiency in high density lipoproteins (HDL) and apolipoprotein A-I (apo A-I). The disease is caused by a rare autosomal codominant allele. Autosomal means that it is caused by a gene located on one of the non-sex chromosomes. Codominant means that people with no Tangier alleles have no disease effect, people with one allele are called heterozygotes and have a moderate effect, and people with two alleles are called homozygotes and have the full disease effect. Usually it is the homozygotes that are considered to have Tangier disease. The name of the disease comes from Tangier Island, a small Chesapeake Bay island in Virginia, home of the first two Tangier disease patients [6]. As of last year, about 54 people

with homozygous Tangier disease have been identified throughout the world [90]. Patients with homozygous Tangier disease typically have low HDL (5% normal), apo A-I (1% normal), cholesterol (30% normal), and low density lipoproteins (LDL) (40% normal) [32]. Tangier patients also have cholesterol deposition in various macrophage-containing tissues, including the spleen, liver, lymph nodes, thymus, skin, and cornea [6]. Macrophages are in the class of cells known as scavenger cells. Clinically, the patients are found to have enlarged orange tonsils, enlarged liver and spleen, and neuropathy [32, 90]. The enlarged organs are a direct consequence of the cholesterol deposition, but the reason for neuropathy is not known at this time. Tangier patients aged 35-65 years have an increased risk of cardiovascular disease (45% vs. 5% in controls); however, this risk is not nearly as high as the extremely low HDL levels would predict. It is not completely understood why this is so, but it is possibly due to the low LDL levels [90].

Cholesterol is a white, waxy substance found in the body, and is essential for cell growth and viability. It supports membrane biosynthesis and acts as a structural component of cell membranes [40, 42]. Cholesterol comes from dietary sources and is also synthesized in the liver. Lipoproteins are responsible for maintaining a constant cholesterol level in the body by acting as the cholesterol transportation system [39, 40]. Lipoproteins are a combination of lipid, a fatty substance insoluble in water, and protein, a sequence of amino acids. The two major classes of lipoproteins are low density (1.019-1.063 g/mL) and high density (1.063-1.25 g/mL) [32, 34, 88].

Low density lipoproteins (LDL) are the primary cholesterol carriers, carrying about 2/3 of the body's cholesterol. LDL's main function is to deliver cholesterol to cells. This is a fairly well-understood process. LDL delivers cholesterol by attaching to cell surface receptors [39, 40]. These receptors are proteins that attract LDL and enable the cell to absorb LDL and get the cholesterol. Since LDL functions to deliver cholesterol to cells, it is not surprising, then, that high levels of LDL in the body are associated with cardiovascular disease. This is how cholesterol being carried

by LDL (i.e. LDL-cholesterol) got the name "bad cholesterol".

High density lipoproteins (HDL) are the other major cholesterol carriers, carrying about 1/3 of the body's cholesterol. HDL are smaller than LDL. The main function of HDL is to take cholesterol from cells and remove it from the body. This process is not as well understood as the LDL cholesterol delivery [39, 40]. An important benefit of the cholesterol removal is the prevention of arterial cholesterol buildup [69]. It is not surprising, then, to find that high HDL levels are associated with low risk of cardiovascular disease, hence the name "good cholesterol".

The process of removing cholesterol from the body is called reverse cholesterol transport. This can be broken down into three steps [33, 39, 63]: the transfer of cholesterol from the cells to HDL, transfer to the HDL core, and delivery to the liver for removal. One way the first step occurs is called cholesterol efflux [71], the extraction of cholesterol stored in intracellular pools. It is believed that cholesterol efflux occurs through the action of cell surface receptors [40, 69, 71].

It is not clear exactly why low HDL levels are associated with cardiovascular disease. It is not necessarily just because more HDL in the body causes more cholesterol to be removed. More likely, it is a problem with the cholesterol removal process. Nascent HDL need cholesterol to grow. If they don't get enough cholesterol from cells, they do not grow into mature HDL. Thus perhaps the low HDL count is an effect rather than the cause. Tangier patients provide a unique opportunity to study this. Tangier disease patients accumulate excess cholesterol in tissues containing macrophages. Macrophages are different from most cells. In most cells, the LDL cholesterol delivery system is suppressed when the cell has enough cholesterol. This is not so in macrophages. Macrophages continue to collect cholesterol even when overloaded and are thus dependent on a cholesterol removal system. Since Tangier disease patients collect excess cholesterol only in macrophage-containing tissues, it seems the defect must be in the cholesterol removal pathway.

Recent studies have focused on the protein apolipoprotein A-I (apo A-I) [34].

Apo A-I plays an important role in the structure and growth of HDL and mediates cholesterol removal. Even though the cholesterol is accumulating in macrophages, fibroblasts are studied because they are much easier to culture and their HDL apolipoprotein - mediated cholesterol removal is similar to macrophages. Cholesterol removal mediated by apo A-I has been found to be defective in the fibroblasts of Tangier disease patients.

5.2 A Morphology Experiment

Morphological abnormalities have been found in Tangier cells that correlate with the cholesterol removal defect [81]. The abnormalities involve the Golgi apparatus in fibroblasts and in macrophages. The Golgi apparatus sits in the cytoplasm of the cell and usually consists of 3 to 5 stacked sacs in normal cells. In Tangier cells there are 5 to 7 sacs in the stack. This causes the Tangier cells to generally be larger.

An experiment was conducted to quantitatively assess the existence of morphological abnormalities in Tangier fibroblasts [70]. The morphological quantities used in the analysis are area, perimeter and shape factor, where shape factor is defined to be

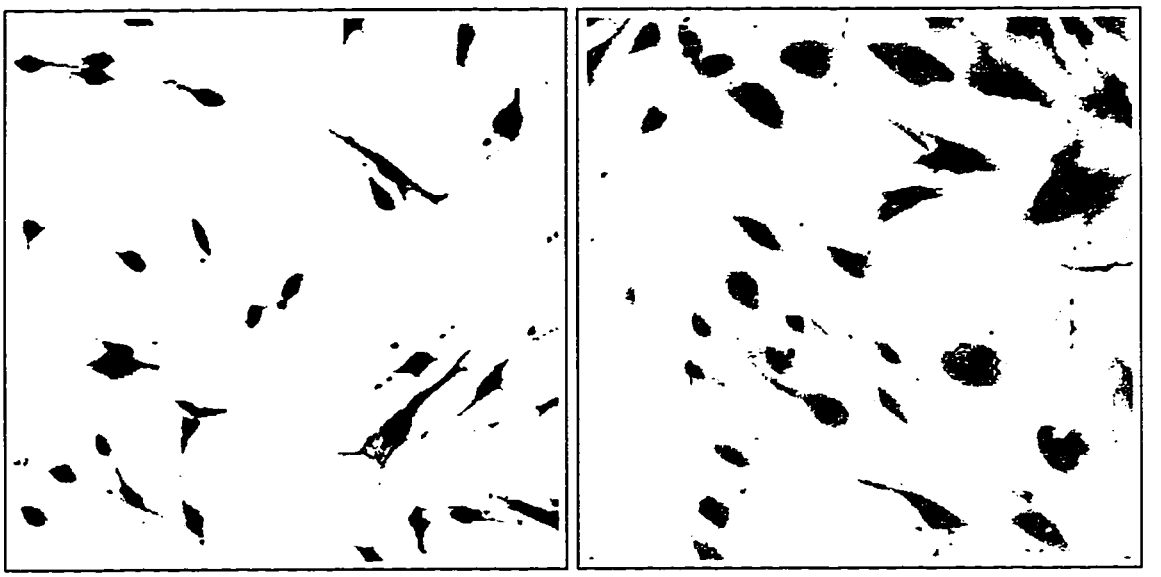
$$\text{shape factor} = \frac{(\text{perimeter})^2}{4\pi\text{area}}.$$

In the experiment there are cells from two Tangier patients and two normal (non-Tangier) patients. The labels for the Tangier patients are IM and JKW and for the normal patients are 282 and 289. For each patient, some cells are transformed with a virus, giving eight cell lines in all. Four to five images were taken of a cell culture from each cell line. Each image represents a 540×540 μm frame on the cell culture. It was expected that there would be differences between the normal and Tangier patients, and that the viral transformation would mask these differences. It was also expected that there might be differences between the primary (untransformed) and transformed cells for both normal and Tangier patients. The results of the analysis using all of

the collected data will be reported elsewhere [70]. For purposes of illustration, four images were selected and are given in Figure 5.1. The cell lines shown are normal primary (patient 282), Tangier primary (patient IM), normal transformed (patient 282) and Tangier transformed (patient IM). Darker colors indicate more fluorescent intensity. The normal cells are a bit more spindle-shaped than the Tangier cells. The transformed cells appear to have more distinct edges and are a bit less stringy than the primary cells. There appears to be little difference between transformed normal and transformed Tangier. A standard analysis method and the cell image model developed in this dissertation were applied to these four images. The results of these analyses are given and compared. The image that was used to illustrate segmentation methods in Chapter 1 is from a normal primary (patient 289) cell line. The two analysis methods are also applied to this image, and a brief discussion of all the methods is given.

5.3 Analysis Using a Standard Method

The cell images were analyzed using a DASY 9000 Workstation (Meridian Instruments, Okemos, MI). The method involved thresholding, manual separation, elimination of border cells and artifacts, parameter calculation and rechecking. Thresholding was set at 16, so that pixels with values between 0 and 15 were set to zero intensity and pixels with values between 16 and 255 unchanged. Cells were separated from each other and the image boundary manually by drawing lines on the computer screen with a mouse. Objects were then defined by the software to be any contiguous set of non-zero pixels. Objects smaller than 300 μm^2 or touching the image boundary were then eliminated and the remainder were analyzed. The cells were counted and then area, perimeter and shape factor were computed. It is unclear what method the software used to account for pixel discretization. The results were checked and the analysis was redone if anything was obviously incorrect. For example, if a cell was



(a) normal primary

(b) Tangier primary

(c) normal transformed

(d) Tangier transformed

Figure 5.1: Tangier images for illustration. The cell lines shown are normal primary, Tangier primary, normal transformed and Tangier transformed.

Table 5.1: Morphological results for the four illustrative images using the DASY 9000 Workstation software.

		area		perimeter		shape factor	
cell line	# of cells	mean	sd	mean	sd	mean	sd
normal primary	26	650	472	152	87	2.97	1.61
Tangier primary	28	1860	1330	222	109	2.33	1.27
normal transformed	11	1630	1260	216	95	2.60	1.15
Tangier transformed	10	1820	880	223	69	2.29	0.64

too large or the number of cells too small, then further manual separation was done.

Results for the three morphological quantities are given in Table 5.1. Mean area for the cells in the image from the Tangier primary cell line is much higher than for those from normal primary. Additionally, perimeter is higher and shape factor is a bit lower. Mean shape factor is a bit lower and area is a bit higher for Tangier transformed than normal transformed, but perimeter is about the same. Mean area for normal transformed is much higher than for normal primary. Perimeter is also higher, and shape factor is a bit lower. All three quantities are about the same for Tangier primary and Tangier transformed. Standard deviations are large in almost all cases, reflecting large variability between cells within each image as well as skewness in the data.

5.4 Analysis Using the Cell Image Model

The methods described in this dissertation were used to estimate the same morphological quantities given by the DASY software. The estimation algorithm, in its current form, uses too much computer memory to analyze the full 270 × 270 pixel image. Each image was reduced to 256 × 256 and then averaged to form a 128 × 128

Table 5.2: Morphological results for the four illustrative images using the cell image model developed in this dissertation.

		area		perimeter		shape factor	
cell line	# of cells	mean	sd	mean	sd	mean	sd
normal primary	26	432	216	89	24	1.56	0.46
Tangier primary	28	1310	920	137	45	1.30	0.26
normal transformed	13	1710	830	182	38	1.69	0.59
Tangier transformed	12	1560	380	186	22	1.86	0.69

image. The parameter estimates were then converted to fit the full image. The blurring parameter was set at 1.5 FWHM in pixel units on the 128×128 image, and the smoothing parameter was set at $\exp(5) \approx 148$.

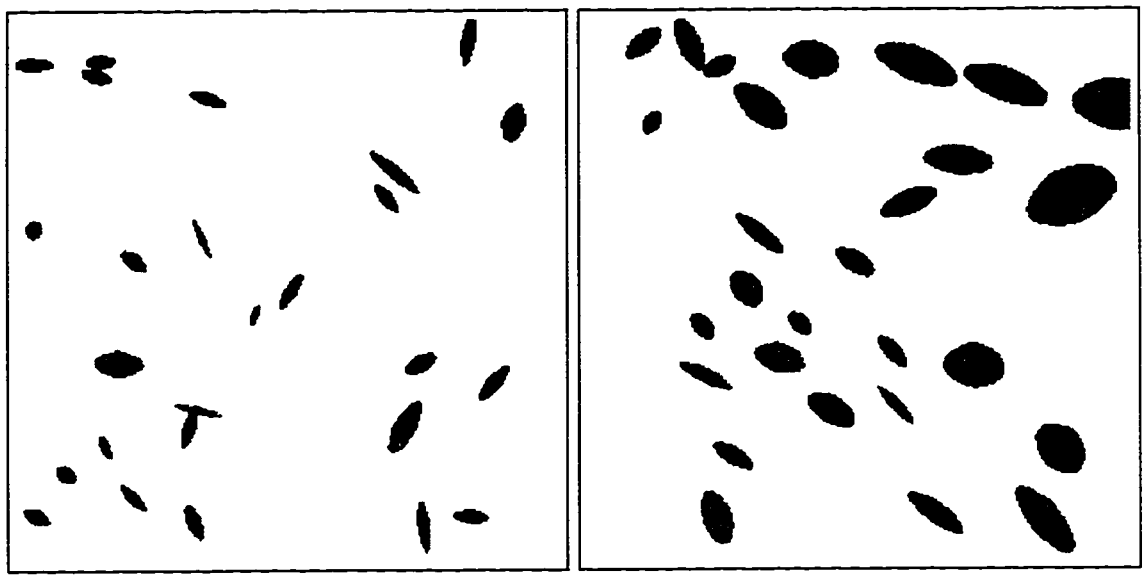
The results are given in Table 5.2. Many of the results are the same as those obtained with the previous method; however, there are some notable differences. The comparisons between Tangier primary and normal primary are about the same as before. The differences in mean area and shape factor between Tangier transformed and normal transformed shift to the opposite direction but are still small, while perimeter is still about the same. Between normal transformed and normal primary, the difference in mean perimeter is more pronounced, the difference in shape factor is still small but in the opposite direction, and area is still about the same. All of the quantities are higher for Tangier transformed than Tangier primary instead of being the same. Almost all estimates are lower for the cell image model, including mean area, mean perimeter, mean shape factor, and the standard deviations.

Most of these differences would have little effect on inference, except for the differences between Tangier transformed and Tangier primary, and the lower standard deviations. These differences make sense. The DASY method considers all contiguous non-zero pixels after thresholding and manual separation to be part of the cell,

while the cell image model estimates a smooth boundary based on ellipses and a characteristic shape curve averaged over all cells on the image. Additionally, intensity is estimated to be constant over the cell and the image is deblurred. The low intensity strings extending from the cells are thus included as part of the cell when using the DASY method, but not necessarily when using the cell image model. This makes the mean value and variability of the quantities higher for the DASY method. This also probably causes the differences between Tangier transformed and Tangier primary to be nonexistent using the DASY method and pronounced using the cell image model, because the Tangier primary image has large low intensity regions at the outer regions of the cells while the Tangier transformed image has much less. The cell image model is preferable if the strings and low intensity regions are considered to be artifacts, and it is a smooth approximation if they are considered to be part of the cell. Note that it is not possible to compare the results by cell because the DASY software does not identify the cells nor their locations.

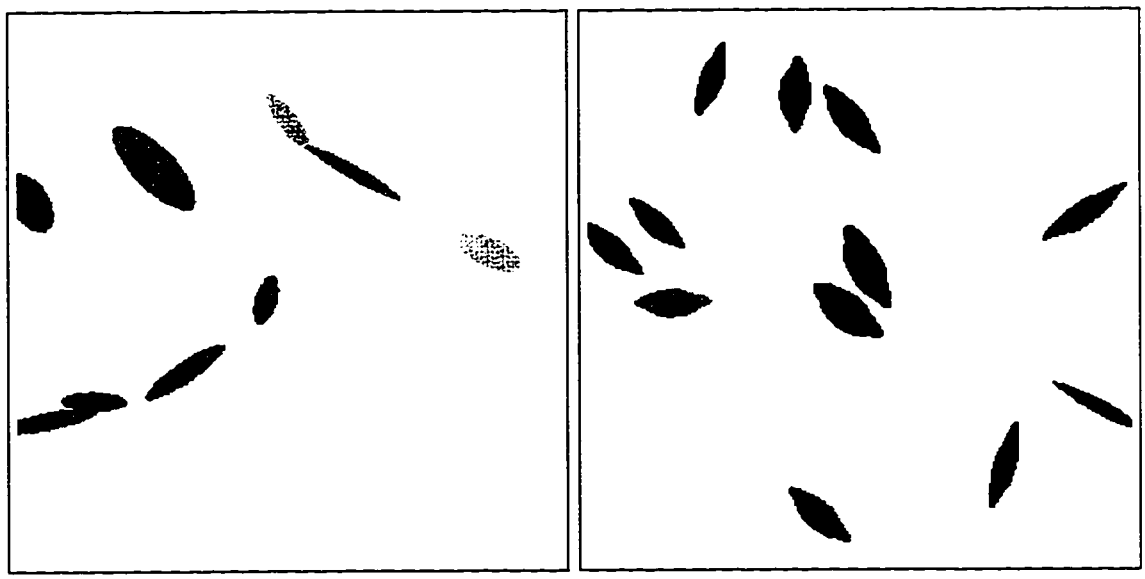
A major advantage for the method developed in this dissertation over the DASY method is the amount of effort required by the user. The DASY method requires threshold selection and a potentially time consuming manual separation process. The cell image model only requires identification of initial values for cell centers and a rough guess for average cell size and aspect ratio. Additionally, the threshold and manual separation are fixed and subject to human error, while the cell centers, and the rough estimates of average size and aspect ratio, are updated and optimized by the algorithm.

Another advantage of the cell image model is the ability to obtain estimates of the image and characteristic shape function. These are shown for the four images in Figures 5.2 and 5.3. The estimated images reveal the smoothing aspects of the algorithm. There is no blur, there are no extended strings, and cell intensities are constant over the cell. For most of the cells the estimates appear quite good. The shape function estimates indicate that the shape is less complex for Tangier primary



(a) normal primary

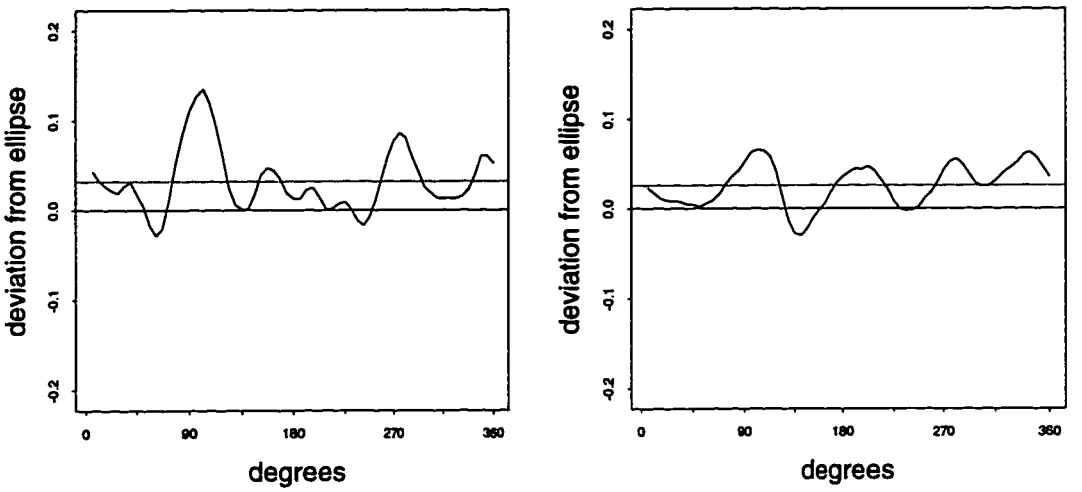
(b) Tangier primary



(c) normal transformed

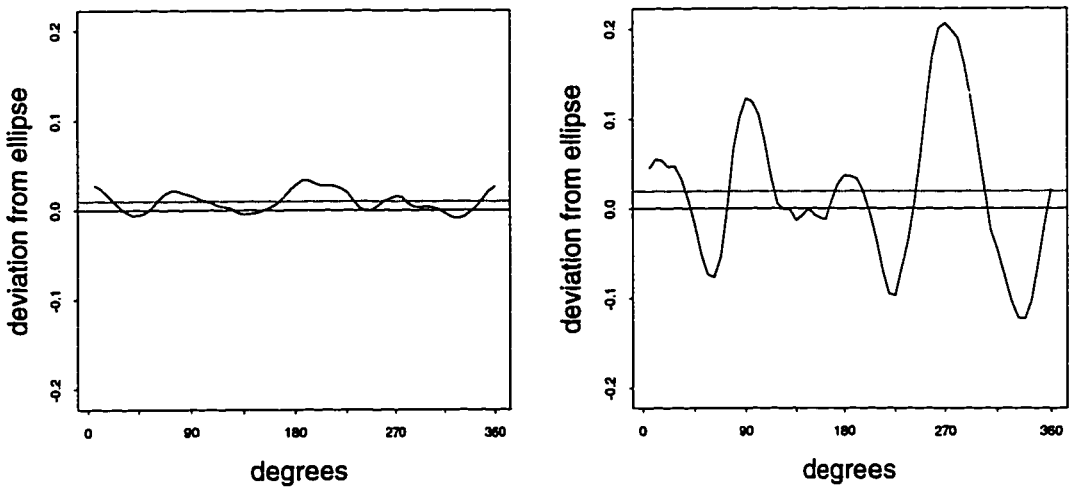
(d) Tangier transformed

Figure 5.2: Estimated images when the cell image model is applied to the images in Figure 5.1.



(a) normal primary

(b) Tangier primary



(c) normal transformed

(d) Tangier transformed

Figure 5.3: Estimated characteristic shape curves for the Tangier images. The solid line at zero indicates the shape curve for the ellipse with the estimated cell parameters. The dotted line indicates the mean value of the curve.

Table 5.3: Morphological results for the example image used in Chapter 1 using the DASY 9000 Workstation software (method 1) and the cell image model developed in this dissertation (method 2).

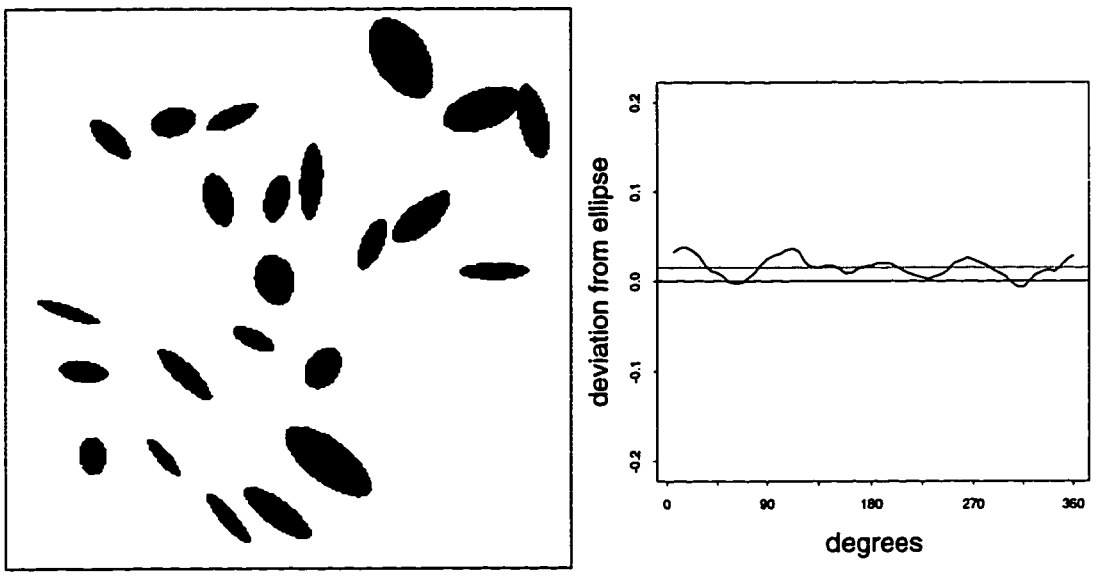
		area		perimeter		shape factor	
method	# of cells	mean	sd	mean	sd	mean	sd
1	21	2320	1390	271	109	2.76	1.47
2	23	1280	820	143	38	1.44	0.35

or normal transformed than for normal primary, and more complex for Tangier transformed than Tangier primary or normal transformed. There is also bias in all the estimates, in the sense that the average value for the curve is not zero. This indicates that the size parameter would need to be adjusted to remove this bias in the curve. Note that iteration to convergence would be expected to eliminate this bias.

5.5 Analysis of Segmentation Example Image

The DASY method and the cell image model were used to analyze the same image used in Chapter 1 to illustrate image segmentation methods. This image is of normal primary (patient 289) cells and is illustrated in Figure 1.4. The results for the morphological parameters are given in Table 5.3. Mean area, perimeter and shape factor are all lower for the cell image model method, consistent with the findings for the four images analyzed in the previous sections. Mean area and perimeter are both much higher for this image than for the normal primary (patient 282) image analyzed in the previous sections. This indicates large variability between cell from different patients with the same disease status.

The cell image model provides further information as well. The estimated image and characteristic shape function are given in Figure 5.4. The estimated image appears reasonable for most of the cells. As before, the estimated shape curve is biased



(a) estimated image

(b) estimated shape curve

Figure 5.4: Estimated image and shape curve for the Tangier image used to illustrate segmentation methods in Chapter 1. In (b), the solid line at zero indicates the shape curve for the ellipse with the estimated cell parameters. The dotted line indicates the mean value of the curve.

upward in the sense that the average deviation from an ellipse is greater than zero. The curve is plotted on the same scale as the curves in the previous section, revealing a less complex characteristic shape than the primary cells from the other normal patient. Estimates of the cell specific parameters are given in Table 5.4. These parameter estimates, along with the shape function estimate, were used to compute the estimated image and the morphological quantities of interest.

The basic segmentation methods illustrated in Chapter 1 provide significantly less information than either method demonstrated in this chapter. Recall the thresholding, edge detection and uniform region detection methods illustrated in Figures 1.5, 1.6 and 1.7 respectively. The thresholded image has too many small artifacts and clustered cells to be useful on its own. The DASY method allows the user to choose

Table 5.4: Estimated cell specific parameters for the Tangier image used to illustrate segmentation methods in Chapter 1.

x location	y location	size	aspect ratio	orientation	intensity
20	191	21	0.58	30	98
46	231	20	0.48	110	101
52	256	19	0.38	192	63
127	237	16	0.24	91	137
100	201	17	0.40	130	119
113	177	13	0.39	156	148
82	147	18	0.29	357	146
90	130	11	0.49	345	148
91	100	13	0.56	18	137
50	107	13	0.32	114	156
52	77	11	0.62	104	116
60	46	13	0.39	45	131
131	128	12	0.78	18	140
160	118	11	0.38	64	157
175	153	11	0.74	142	146
221	155	25	0.41	53	121
247	130	20	0.32	55	143
249	105	16	0.23	42	157
219	72	11	0.27	41	167
178	83	18	0.28	47	123
218	37	9	0.74	5	121
176	33	12	0.44	83	160
147	26	15	0.19	71	153

a threshold, remove artifacts manually, and manually separate clustered cells. The method is thus an example of thresholding followed by human interaction. Edge detection algorithms have a tendency to pick up artifacts and variations within cells. No attempt was made to post-process these algorithms. Uniform region detection algorithms are perhaps most similar to the cell image model, in that each estimated region has constant intensity and deblurring is easily incorporated. The main problem is that no information about the size and shape of cells is incorporated and thus some of the identified regions are definitely not unique cells.

5.6 Summary

The cell image model provides a reasonable fit to cell images. Estimates of differences in morphological parameters between cell lines are comparable to a standard method. The differences between the estimates obtained using the two methods make sense due to the averaging and deblurring aspects of the model. Furthermore, the cell image model requires less user input and provides more information. The cell image model is thus a useful tool for analyzing fluorescence cell images.

Chapter 6

DISCUSSION

6.1 Summary

Fluorescence image cytometry is a common diagnostic method used for the analysis of cells and cellular components in tissue or culture. Fluorescence imaging is useful in cytometry because fluorescent stains are highly sensitive and specific to objects of interest and imaging allows for direct spatial and morphological measurements. Image segmentation plays a key role in quantitative analysis of fluorescence cell images. The basic segmentation methods were not designed for cell images and thus do not perform very well when cells are clustered or varied in size or intensity. Several methods have been attempted to improve upon these basic methods, with varying degrees of success.

In this dissertation, a statistical approach is developed for quantitative analysis of fluorescence cell images. A cell image model is used that incorporates three main components: a nonparametric specification for the characteristic shape of the cells, several parameters which are allowed to be different for each cell, and an image representation that incorporates blurring and noise. Results from a fluorescent bead study and the fluorescence microscopy literature indicate that the image representation is reasonable.

The cell image model is a nonlinear convolution and parameter estimation is an ill-posed inverse problem. The method of regularization is used to obtain estimates of the model parameters. The characteristic shape function and the cell specific parameters are estimated separately. A modified Gauss-Newton procedure is used for each portion of the algorithm.

The properties of the model are evaluated using simulation studies. Estimation of the true image, characteristic shape function, cell specific parameters, and certain functionals of interest appears to be consistent with respect to image noise. In other words, the mean squared error between the estimate and the truth gets smaller as image noise is reduced. Estimation of the shape function and certain functionals of interest also appears consistent with respect to the number of cells on the image. The quality of the shape function estimate depends on the quality of the cell parameter estimates and *vice versa*. The two parts of the algorithm are thus interdependent.

The cell image model is illustrated using data from a Tangier disease morphology experiment. The study of Tangier disease provides clues to understanding certain causes of cardiovascular disease. The cell image model provides reasonable estimates and compares well with a standard method. Area and perimeter estimates are lower than those obtained using the standard method. This is explained by the averaging and deblurring aspects of the model. Since less user input is needed, and more information is obtained, the cell image model is a useful tool for quantitative analysis of fluorescence cell images.

6.2 Future Work

There are some extensions to the model that could be explored. The characteristic shape function, currently fixed to be the same for all cells on the image, could be allowed to vary, either by estimating a separate curve for each cell or incorporating random effects specifications. Random effects might also be used on the cell specific parameters. Fluorescence intensity is often higher in the nucleus than in the rest of the cell, and the model fit would likely improve if this were incorporated.

Some extensions to the estimation algorithm could be explored. First, a method of iteration between the two main parts of the algorithm could be incorporated, probably improving on the two step procedure implemented here. A global optimization pro-

cedure, such as simulated annealing, might perform better than the Gauss-Newton procedure. It might help to optimize the smoothing and blurring parameters, although computation is likely to be prohibitive using any of the usual methods. To increase the size of the image that can be analyzed, a regional updating algorithm could be implemented, rather than updating the entire image at once. Methods such as thresholding, mathematical morphology, peak finding algorithms, clustering and cross validation might be implemented to initialize the algorithm and determine the number of cells. For three dimensional data, methods from stereology [66, 94] might be explored, or a 3D version of the model could be developed.

There is some related work that would enhance the algorithm as well. Methods could be developed for classification of cells based on the parameters of the model. Finally, the estimation algorithm for the model could be incorporated into a user-friendly software package.

BIBLIOGRAPHY

[1] D. A. Agard et al. Fluorescence microscopy in three dimensions. In *Methods in Cell Biology*, volume 29, chapter 13. Academic Press, New York, 1989.

[2] R. S. Aikens, D. A. Agard, and J. W. Sedat. Solid-state imagers for microscopy. In *Methods in Cell Biology*, volume 29, chapter 16. Academic Press, New York, 1989.

[3] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration through deformable templates. *Journal of the American Statistical Association*, 86:376–387, 1991.

[4] H. Ancin et al. Advances in automated 3-D image analysis of cell populations imaged by confocal microscopy. *Cytometry*, 25:221–234, 1996.

[5] J. J. Art and M. B. Goodman. Rapid scanning confocal microscopy. In *Methods in Cell Biology*, volume 38, chapter 2. Academic Press, New York, 1993.

[6] G. Assmann, G. Schmitz, and H. B. Brewer, Jr. Familial high density lipoprotein deficiency: Tangier disease. In *The Metabolic Basis of Inherited Disease*, chapter 50. McGraw-Hill, New York, 1989.

[7] A. J. Baddeley and M. N. M. van Lieshout. Stochastic geometry models in high-level vision. In K. V. Mardia and G. K. Kanji, editors, *Statistics and Images 1*, chapter 11. Carfax, Abingdon, Oxfordshire, U.K., 1993.

[8] B. Bailey et al. Enhancement of axial resolution in fluorescence microscopy by standing-wave excitation. *Nature*, 366:44–48, 1995.

[9] J. D. Banfield and A. E. Raftery. Ice floe identification in satellite images using mathematical morphology and clustering about principle curves. *Journal of the American Statistical Association*, 87(417):7–16, 1992.

[10] J. Barba, L. Yuan, and J. Gil. Cell contour extraction on multi-threshold images. *Pathology, Research and Practice*, 188:449–453, 1992.

[11] G. T. Bartoo. *Quantitative neuropathological measures in Alzheimer's disease.* PhD thesis, University of Washington, Seattle, WA, 1995.

[12] V. B. Berezin, G. Skladchikova, and E. Bock. Evaluation of cell morphology by video recording and computer-assisted image analysis. *Cytometry*, 27(2):106–116, 1997.

[13] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[14] G. Bigras et al. Cellular sociology applied to neuroendocrine tumors of the lung: quantitative model of neoplastic architecture. *Cytometry*, 24:74–82, 1996.

[15] I. O. Bohachevsky, M. E. Johnson, and M. L. Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28:209–217, 1986.

[16] F. L. Bookstein. Size and shape spaces for landmark data in two dimensions (with discussion). *Statistical Science*, 1:181–242, 1986.

[17] S. Bose and F. O'Sullivan. A region-based segmentation method for multichannel image data. *Journal of the American Statistical Association*, 92(437):92–106, 1997.

[18] P. B. Brakenhoff and P. Barends. Confocal scanning light microscopy with high aperture immersion lenses. *Journal of Microscopy*, 117(2):219–232, 1979.

[19] N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25, 1993.

[20] J. F. Brinkley. A flexible, generic model for anatomic shape: Application to interactive two-dimensional medical image segmentation and matching. *Computers and Biomedical Research*, 26:121–142, 1993.

[21] W. A. Carrington et al. Superresolution three-dimensional images of fluorescence in cells with minimal light exposure. *Science*, 268:1483–1487, 1995.

[22] R. J. Carroll and P. Hall. Optimal rates of convergence for deconvolving a density. *Journal of the American Statistical Association*, 83:1184–1186, 1982.

[23] V. Chalana. *Deformable Models for Segmentation of Medical Ultrasound Images.* PhD thesis, University of Washington, Seattle, WA, 1996.

[24] D. D. Cox. Approximation of method of regularization estimators. *The Annals of Statistics*, 16:694–712, 1988.

[25] D. D. Cox and F. O'Sullivan. Asymptotic analysis of penalized likelihood and related estimators. *The Annals of Statistics*, 18:1676–1695, 1990.

[26] J. E. Darnell, H. Lodish, and D. Baltimore. *Molecular Cell Biology.* Scientific American Books, New York, 1986.

[27] Z. Darzynkiewicz et al. Cytometry in cell necrobiology: analysis of apoptosis and accidental cell death (necrosis). *Cytometry*, 27:1–20, 1997.

[28] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[29] E. R. Dougherty, Y. Chen, and A. Waks. Bayesian morphological peak estimation and its application to chromosome counting via fluorescence *in situ* hybridization. *Pattern recognition*, 29(6):987–996, 1996.

[30] A. I. Dow et al. Automatic multiparameter fluorescence imaging for determining lymphocyte phenotype and activation status in melanoma tissue sections. *Cytometry*, 25:71–81, 1996.

[31] C. H. Edwards, Jr. and D. E. Penney. *Calculus and Analytic Geometry.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986.

[32] J. Emmerich et al. Familial HDL deficiency due to marked hypercatabolism of normal apo A-I. *Arteriosclerosis and Thrombosis*, 13:1299–1306, 1993.

[33] G. Franceschini, J. P. Werba, and L. Calabresi. Drug control of reverse cholesterol transport. *Pharmacology and Therapeutics*, 61:289–324, 1994.

[34] G. A. Francis, R. H. Knopp, and J. F. Oram. Defective removal of cellular cholesterol and phospholipids by apolipoprotein A-I in Tangier disease. *Journal of Clinical Investigation*, 96:78–87, 1995.

[35] N. S. Friedland and A. Rosenfeld. Compact object recognition using energy-function-based optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):770–777, 1992.

[36] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984.

[37] A. L. Givan. *Flow Cytometry: First Principles.* Wiley-Liss, New York, 1992.

[38] I. J. Good and R. A. Gaskins. Nonparametric roughness penalties for probability densities. *Biometrika*, 58:255–277, 1971.

[39] A. J. R. Habenicht, U. Janβen-Timmen, R. Ziegler, and G. Schettler. Lipoproteins and their functions. *The Clinical Investigator*, 72:639–652, 1994.

[40] L. Husten. Cholesterol to go. *Discover*, pages 30–31, December 1990.

[41] L. Ji. Fully automatic chromosome segmentation. *Cytometry*, 17:196–208, 1994.

[42] W. J. Johnson et al. Cholesterol transport between cells and high-density lipoproteins. *Biochimica et Biophysica Acta*, 1085:273–298, 1991.

[43] I. M. Johnstone and B. W. Silverman. Speed of estimation in Positron Emission Tomography and related inverse problems. *The Annals of Statistics*, 18:251–280, 1990.

[44] D. Kim et al. Semiautomated nuclear shape analysis of prostatic carcinoma and benign prostatic hyperplasia. *Analytical and Quantitative Cytology and Histology*, 16:400–414, 1994.

[45] R. Kiss et al. Relationship between DNA ploidy level and tumor sociology behavior in 12 nervous cell lines. *Cytometry*, 20:118–126, 1995.

[46] L. G. Koss. Quantitative and analytical cytology in historical perspective. *Journal of Cellular Biochemistry, Supplement*, 19:23–27, 1994.

[47] J. S. J. Lee et al. A processing strategy for automated papanicolaou smear screening. *Analytical and Quantitative Cytology and Histology*, 14(5):415–425, 1992.

[48] T. Leonard. Density estimation, stochastic processes and prior information (with discussion). *Journal of the Royal Statistical Society, Series B*, 40:113–146, 1978.

[49] C. Lesty et al. An application of mathematical morphology to analysis of the size and shape of nuclei in tissue sections of non-Hodgkin's lymphoma. *Cytometry*, 7:117–131, 1986.

[50] K.-Y. Liang and S. L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73:13–22, 1986.

[51] C. E. Liedtke, T. Gahm, F. Kappei, and B. Aeikens. Segmentation of microscopic cell scenes. *Analytical and Quantitative Cytology and Histology*, 9(3):197–211, 1987.

[52] M. J. Lindstrom and D. M. Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46:673–687, 1990.

[53] S. J. Lockett et al. Automated fluorescence image cytometry: DNA quantification and detection of chlamydial infections. *Analytical and Quantitative Cytology and Histology*, 13(1):27–44, 1991.

[54] S. J. Lockett and B. Herman. Automatic detection of clustered, fluorescent-stained nuclei by digital image-based cytometry. *Cytometry*, 17:1–12, 1994.

[55] S. J. Lockett, K. Jacobson, and B. Herman. Application of 3D digital deconvolution to optically sectioned images for improving the automatic analysis of fluorescent-labeled tumor specimens. *SPIE Biomedical Image Processing and Three-Dimensional Microscopy*, 1660:130–139, 1992.

[56] R. Maitra. *Variability Estimation in Linear Inverse Problems.* PhD thesis, University of Washington, Seattle, WA, 1996.

[57] L. A. Mancl and B. G. Leroux. Efficiency of regression estimates for clustered data. *Biometrics*, 52:500–511, 1996.

[58] K. V. Mardia and I. L. Dryden. The statistical analysis of shape data. *Biometrika*, 76(2):271–281, 1989.

[59] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis.* Academic Press, London, 1979.

[60] K. V. Mardia and W. Qian. A Bayesian method for compact object recognition from noisy images. In D. M. Titterington, editor, *Complex Stochastic Systems and Engineering*, IMA Conference Series Number 54, pages 155–166. Clarendon Press, Oxford, 1995.

[61] L. C. Martin. *The Theory of the Microscope.* Elsevier Publishing Co., Inc., New York, 1966.

[62] B. Matsumoto, editor. *Cell Biological Applications of Confocal Microscopy*, volume 38 of *Methods in Cell Biology.* Academic Press, New York, 1993.

[63] N. E. Miller. HDL metabolism and its role in lipid transport. *European Heart Journal*, 11(Supplement H):1–3, 1990.

[64] A. Moragas et al. Textural analysis of lymphoid cells in serous effusions: A mathematical morphologic approach. *Analytical and Quantitative Cytology and Histology*, 15(3):165–170, 1993.

[65] D. Nychka. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83:1134–1142, 1988.

[66] D. Nychka et al. Cross-validated spline methods for the estimation of three-dimensional tumor size distributions from observations on two-dimensional cross sections. *Journal of the American Statistical Association*, 79:832–845, 1984.

[67] D. W. Nychka and D. D. Cox. Convergence rates for regularized solutions of integral equations from discrete noisy data. *The Annals of Statistics*, 17(2):556–572, 1989.

[68] A. B. Olshen and F. O'Sullivan. Camouflaged deconvolution with application to blood curve modeling in fdg pet studies. *submitted to Journal of the American Statistical Association*, 1997.

[69] J. F. Oram et al. The role of HDL receptors in removal of cellular cholesterol. In L. L. Gallo, editor, *Cardiovascular Disease 2.* Plenum Press, New York, 1995.

[70] J. F. Oram et al. Reduction in apolipoprotein-mediated removal of cellular lipids by viral transformation of human fibroblasts and its reversion by cAMP. Lack of effect with Tangier Disease cells. *to be sumitted to Journal of Cell Biology*, 1997.

[71] J. F. Oram, A. J. Mendez, and E. L. Bierman. The HDL receptor. In A. L. Catapano, F. Bernini, and A. Corsini, editors, *High Density Lipoproteins: Physiopathology and Clinical Relevance*, pages 87–97. Raven Press, Ltd., New York, 1993.

[72] F. O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1:502–527, 1986.

[73] F. O'Sullivan. Convergence characteristics of methods of regularization estimators for nonlinear operator equations. *SIAM Journal of Numerical Analysis*, 27:1635–1649, 1990.

[74] F. O'Sullivan. A study of least squares and maximum likelihood for image reconstruction in positron emission tomography. *The Annals of Statistics*, 23(4):1267–1300, 1995.

[75] J. Pladellorens et al. Using mathematical morphology to determine left ventricular contours. *Physics in Medicine and Biology*, 38(12):1877–1894, 1993.

[76] W. H. Press et al. *Numerical Recipies in C.* Cambridge University Press, Cambridge, 1992.

[77] J. H. Price, E. A. Hunter, and D. A. Gough. Accuracy of least squares designed spatial FIR filters for segmentation of images of fluorescence stained cell nuclei. *Cytometry*, 25(4):303–316, 1996.

[78] W. Qian, D. M. Titterington, and J. N. Chapman. An image analysis problem in electron microscopy. *Journal of the American Statistical Association*, 91:944–952, 1996.

[79] M. Revenu et al. An automatic system for the classification of cellular categories in cytological images. *SPIE Intelligent Robots and Computer Vision XII*, 2055:32–43, 1993.

[80] R. S. Riley, E. J. Mahin, and W. Ross. *Clinical Applications of Flow Cytometry.* Igaku-Shoin, New York, 1993.

[81] H. Robeneck and G. Schmitz. Abnormal processing of golgi elements and lysosomes in Tangier disease. *Arteriosclerosis and Thrombosis*, 11(4):1007–1020, 1991.

[82] K. Rodenacker et al. Cytometry in histological sections of colon carcinoma. *Pathology, Research and Practice*, 188:556–560, 1992.

[83] B. Roysam et al. Unsupervised noise removal algorithms for three-dimensional confocal fluorescence microscopy. *Micron and Microscopica Acta*, 23(4):447–461, 1992.

[84] B. Roysam et al. Algorithms for automated characterization of cell populations in thick specimens from 3-d confocal fluorescence microscopy data. *Journal of Microscopy*, 173(2):115–126, 1994.

[85] M. Rudemo, I. Skovgaard, and H. Stryhn. Maximum likelihood estimation of curves in images. Technical Report 90-4, Department of Mathematics and Physics, Royal Veterinary and Agricultural University, Denmark, 1990.

[86] M. Rudemo and H. Stryhn. Approximating the distribution of maximum likelihood contour estimators in two-region images. Technical Report 91-2, Department of Mathematics and Physics, Royal Veterinary and Agricultural University, Denmark, 1991.

[87] V. Russack. Image cytometry: Current applications and future trends. *Critical Reviews in Clinical Laboratory Sciences*, 31(1):1–34, 1994.

[88] G. Schmitz et al. The role of HDL in reverse cholesterol transport and its disturbances in Tangier disease and HDL deficiency with xanthomas. *European Heart Journal*, 11(Supplement E):197–211, 1990.

[89] G. A. F. Seber and C. J. Wild. *Nonlinear Regression.* John Wiley & Sons, New York, 1989.

[90] C. Serfaty-Lacrosniere et al. Homozygous Tangier disease and cardiovascular disease. *Atherosclerosis*, 107:85–98, 1994.

[91] P. Speckman. Spline smoothing and optimal rates of convergence in nonparametric regression problems. *The Annals of Statistics*, 13:970–983, 1985.

[92] K. R. Spring and R. J. Lowy. Characteristics of low light level television cameras. In *Methods in Cell Biology*, volume 29, chapter 15. Academic Press, New York, 1989.

[93] E. H. K. Stelzer, I. Wacker, and J. R. De Mey. Confocal fluorescence microscopy in modern cell biology. *Seminars in Cell Biology*, 2:145–152, 1991.

[94] D. C. Sterio. The unbiased estimation of number and sizes of arbitrary particles using the disector. *Journal of Microscopy*, 134(2):127–136, 1984.

[95] C. J. Stone. Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, 8(6):1348–1360, 1980.

[96] C. J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053, 1982.

[97] P. Suetens et al. Image segmentation: methods and applications in diagnostic radiology and nuclear medicine. *European Journal of Radiology*, 17:14–21, 1993.

[98] D. L. Taylor and E. D. Salmon. Basic fluorescence microscopy. In *Methods in Cell Biology*, volume 29, chapter 13. Academic Press, New York, 1989.

[99] D. L. Taylor and Y. L. Wang, editors. *Fluorescence Microscopy of Living Cells in Culture, Part B: Quantitative Fluorescence Microscopy - Imaging and Spectroscopy*, volume 30 of *Methods in Cell Biology*. Academic Press, New York, 1989.

[100] A. D. H. Thomas, T. Davies, and A. R. Luxmoore. The Hough transform for locating cell nuclei. *Analytical and Quantitative Cytology and Histology*, 14(4):347–353, 1992.

[101] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 5:1035–1038, 1963.

[102] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Halsted Press, New York, 1977.

[103] G. Wahba. Bayesian "confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society, Series B.*, 45:133–150, 1983.

[104] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, Pennsylvania, 1990.

[105] J. E. Wampler and K. Kutz. Quantitative fluorescence microscopy using photomultiplier tubes and imaging detectors. In *Methods in Cell Biology*, volume 29, chapter 14. Academic Press, New York, 1989.

[106] Y. L. Wang and D. L. Taylor, editors. *Fluorescence Microscopy of Living Cells in Culture, Part A: Fluorescent Analogs, Labeling Cells, and Basic Microscopy*, volume 29 of *Methods in Cell Biology*. Academic Press, New York, 1989.

[107] S. J. Wright et al. Introduction to confocal microscopy and three-dimensional reconstruction. In *Methods in Cell Biology*, volume 38, chapter 1. Academic Press, New York, 1993.

[108] I. T. Young. Image fidelity: Characterizing the imaging transfer function. In *Methods in Cell Biology*, volume 30, chapter 1. Academic Press, New York, 1989.

[109] L. Zamai et al. Supravital exposure to propidium iodide identifies apoptotic cells in the absence of nucleosomal DNA fragmentation. *Cytometry*, 23:303–311, 1996.

[110] X. Zheng and M. Kolb. A particle segmentation method based on nucleation and growth of the background. *Journal of Microscopy*, 173(2):165–172, 1994.

[111] H. Ziezold. Mean figures and mean shapes applied to biological figure and shape distributions in the plane. *Biometrical Journal*, 4:491–510, 1994.

Appendix A

PROGRAM LISTING

This appendix includes an overview and listing of the computer codes used to implement the estimation algorithm. All programs were written in the C language. The main algorithm calls the cell parameter estimation algorithm and then the shape function estimation algorithm in a two step manner. The shape function estimation algorithm and the cell parameter estimation algorithm are described in Chapter 3. The image convolution algorithm calculates the cell function and the derivatives for the cell parameters. A listing of all these codes is given. Various C functions were written for utility purposes. Fortran routines from various sources are also utilized. A brief description is given of the C utility functions and the Fortran routines.

To assist readers familiar with programming languages, but not C, a few notes are given. Much of the code is analogous to what would be written in other languages such as Fortran or S. The use of the terms `void` and `float` before the function name refers to what the function returns as a value. A void function returns nothing, and is thus analogous to a Fortran subroutine or an S function returning `invisible()`. The functions `calloc`, `vector`, `matrix`, `f3tensor` and `free` all deal with dynamic memory allocation and are not essential to understanding the algorithm. This should provide enough help, at least for Fortran and S programmers, to read a good portion of the code. For more help consult any basic C programming book.

A.1 Main Algorithm

In this section, the code for the function **cellalg** is given. This function contains the two step procedure that calls the shape function estimation and cell parameter estimation algorithms.

```
/******************/
/*                */
/* void cellalg() */
/*                */
/******************/
/* combine shape function and cell parameter estimation algorithms

parameters: (* indicates changed by cellalg)
       z, d1,d2 (image and dimensions (horiz,vertical))
      *eta = shape estimate
      *psi = {mu1,mu2,a,b,phi,gmma} (cell specific parameters)
       h = Gaussian blurring parameter
       lambda,nl = vector of smoothing parameters and # of them
       nc = number of cells
       ne = number of coefficients in eta
*/

void cellalg(z,eta,psi,h,lambda,d1,d2,nc,ne,nl)
     float **z, *eta, **psi, h, *lambda;
     int d1, d2, nc, ne, nl;
{
  FILE *ofl;
  int i,j,k,l,m, *pte, ni, nstep;
  float **Omega, **wm, usg=0, leta;
#define NP 6
#define PI 3.141593

  Omega=matrix(ne,ne);
  wm=matrix(ne,ne);

/* make Omega regularization matrix */

  for(j=0;j<ne;++j)
    for(k=0;k<ne;++k){
      if(j==k) wm[j][k]=(-2);
      else if(abs(j-k)==1) wm[j][k]=1;
    }
  wm[0][ne-1] = 1;
  wm[ne-1][0] = 1;
  mtranm(wm,Omega,ne,ne);

/* two step algorithm
   first cell specific parameters
   then characteristic shape function
*/

  pte=(int *)calloc(7,sizeof(int));
  for(i=0;i<6;i++)
    pte[i]=1;
  pte[6]=6;
  ni=30; nstep=13;
  estpsi(psi,eta,h,z,d1,d2,nc,ne,pte,ni,nstep);
```

```
  free(pte);

  estshape(psi,eta,h,z,lambda,Omega,&leta,d1,d2,nc,ne,nl);

  free(*Omega); free(Omega);
  free(*wm); free(wm);

} /* end cellalg() */
```

A.2 Shape Function Estimation

In this section, the code for the functions **estshape**, **etacalc**, **lchoose**, **divdif**, **stepfn**, **pseudoval**, **lsecalc** and **objfn** is given. These functions contain the algorithms for estimating the characteristic shape function. The function **estshape** calls **etacalc** to compute updates for various values of the smoothing parameter, then calls **lchoose** to select the best smoothing parameter. The function **etacalc** calls **divdif** to calculate the divided difference approximation to the derivatives, **stepfn** to choose the step length at each iteration, **pseudoval** to compute the pseudovalues given to **lsecalc**, which calculates the least squares estimate, and **objfn** to calculate the objective function.

```
/*******************/
/*                 */
/* void estshape() */
/*                 */
/*******************/
/* iterate shape function estimation over smoothing parameters and
   choose best smoothing parameter

parameters: (* indicates updated by estshape)
   psi - matrix (6 x nc) of cell specific parameters:
            mu1,mu2 - cell location in pixel units
              alpha - size in pixel units
               beta - aspect ratio
                phi - orientation angle [0,360]
              gamma - intensity
  *eta - vector (ne) containing shape curve
   h - Gaussian blurring parameter (standard deviation)
       might switch to fwhm
   z - matrix (d1 x d2) containing image
   lambda - vector (nl) of smoothing parameters
   Omega - matrix (ne x ne) containing regularization matrix
  *leta - value of objective function ||z-c(eta)||^2 + d1d2Le'Oe
   d1,d2 - image dimensions
   nc - number of cells
   ne - dimension of shape curve vector
   nl - number of smoothing parameters
files in:  etatrue.dat
*/

void estshape(psi,eta,h,z,lambda,Omega,leta,d1,d2,nc,ne,nl)
     float **psi, *eta, h, **z, *lambda, **Omega, *leta;
```

```
     int d1, d2, nc, ne, nl;
{
  int i, j, k, l, m;
  float *etastart, **etasm, *letasm;
  FILE *ifl, *ofl;

  etastart=vector(ne);
  etasm=matrix(nl,ne);
  letasm=vector(nl);

  for(i=0;i<ne;++i)
    etastart[i]=eta[i];

/* iterate over smoothing parameter */

  for(m=0;m<nl;m++){
    for(i=0;i<ne;++i)
      eta[i]=etastart[i];
    etacalc(psi,eta,h,z,lambda[m],Omega,&letasm[m],d1,d2,nc,ne);
    for(i=0;i<ne;i++)
      etasm[m][i]=eta[i];
  } /* end smoothing iteration */

  if(nl>1)
    lchoose(etasm,letasm,&m,nl,ne);
  else
    m=0;
  for(i=0;i<ne;i++)
    eta[i]=etasm[m][i];
  *leta=letasm[m];

  free(etastart);
  free(*etasm); free(etasm);
  free(letasm);

} /* end estshape() */

/*****************/
/*               */
/* void etacalc() */
/*               */
/*****************/
/* calculate eta(lambda) by Gauss-Newton iteration
   use divided difference approximation for derivative */

void etacalc(psi,eta,h,z,lambda,Omega,letasav,d1,d2,nc,ne)
     float **psi, *eta, h, **z, lambda, **Omega, *letasav;
     int d1, d2, nc, ne;
{
  int i, j, k;
  float mse, msle, **cfn, *eta0, **X, **zst, *leta;
  FILE *ofl;

  int ni=12; /* maximum number of iterations */

  cfn=matrix(d1,d2); /* cell function */
  eta0=vector(ne);
  X=matrix(d1*d2,ne); /* derivative for shape */
  zst=matrix(d1,d2); /* pseudovalues */
  leta=vector(ni);

  for(i=0;i<ne;++i)
```

```
    eta0[i]=eta[i];
  for(i=0;i<=ni;i++)
    leta[i]=0;
  mse=1;
  msle=1;

/* Gauss-Newton iteration */

  for(k=1;(k<=ni)&&(mse>1e-05)&&(msle>1e-05);++k){

/* calculate cell function and derivative at eta0 */
/* calculate objective function for first iteration */
/* get pseudovalues for shape --> zst (z^star) */
/* calculate eta, an updated estimate of eta0 */
/* choose step length to minimize l(eta) and get l(eta) */
/* test for convergence */
/* update eta0 */

    imconv(cfn,0,0,d1,d2,nc,psi,2.3548*h,eta0,ne);
    divdif(X,cfn,psi,h,eta0,d1,d2,ne,nc);
    if(k==1) leta[0]=objfn(cfn,z,lambda,eta0,Omega,d1,d2,ne);
    pseudoval(zst,cfn,z,X,eta0,d1,d2,ne);
    lsecalc(eta,X,lambda,Omega,zst,d1,d2,ne);
    stepfn(&leta[k],eta,eta0,cfn,z,lambda,Omega,nc,psi,h,d1,d2,ne);

    mse=0;
    for(i=0;i<ne;++i)
      mse+=(eta[i]-eta0[i])*(eta[i]-eta0[i]);
    mse=mse/ne;
    msle=(leta[k-1]-leta[k])/(1+absv(leta[k]));

    for(i=0;i<ne;++i)
      eta0[i]=eta[i];

  } /* end Gauss-Newton iteration */

  *letasav=leta[k-1];

  free(*cfn); free(cfn);
  free(eta0);
  free(*X); free(X);
  free(*zst); free(zst);
  free(leta);

} /* end etacalc() */

/******************/
/*                */
/* void lchoose() */
/*                */
/******************/
/* select smoothing parameter based on ||etahat-etatrue||^2
   m gets index for chosen smoothing parameter */

void lchoose(etasm,letasm,m,nl,ne)
     float **etasm, *letasm;
     int *m, nl, ne;
{
  float *etatrue, *mseta;
  int i, j, k;
  FILE *ifl, *ofl;
```

```
  ifl=fopen("etatrue.dat","r");
  fscanf(ifl,"%d",&i);
  if(i!=ne){
    fprintf(stderr,"dimension of etatrue != dim of eta0\n");
    fclose(ifl);
  }
  else {
    etatrue=vector(ne);
    for(j=0;j<ne;j++)
      fscanf(ifl,"%f",&etatrue[j]);
    fclose(ifl);

    mseta=vector(nl);
    for(k=0;k<nl;k++){
      mseta[k]=0;
      for(j=0;j<ne;j++)
        mseta[k]+=(etasm[k][j]-etatrue[j])*(etasm[k][j]-etatrue[j]);
      mseta[k]/=ne;
    }
    *m=minvi(mseta,nl);
  }

  free(etatrue);
  free(mseta);

} /* end lchoose() */

/*****************/
/*               */
/* void divdif() */
/*               */
/*****************/
/* calculate divided difference approximation to derivative for shape
   use max(10%,1e-5)
   output: X = derivative matrix */

void divdif(X,cfn,psi,h,eta,d1,d2,ne,nc)
     float **X, **cfn, **psi, h, *eta;
     int d1, d2, ne, nc;
{
  int i,j,k;
  float **cfnd, *etad, d;

  cfnd=matrix(d1,d2);
  etad=vector(ne);

  for(k=0;k<ne;k++)
    etad[k]=eta[k];

  for(k=0;k<ne;k++){
    d=(eta[k]>1e-4)?0.1*eta[k]:1e-5; /* force d > 1e-5 */
    etad[k]+=d;
    imconv(cfnd,0,0,d1,d2,nc,psi,2.3548*h,etad,ne);
    for(i=0;i<d1;i++)
      for(j=0;j<d2;j++)
        X[i*d2+j][k]=(cfnd[i][j]-cfn[i][j])/d;
    etad[k]=eta[k];
  }
  free(*cfnd); free(cfnd);
  free(etad);

} /* end divdif() */
```

```
/*****************/
/*               */
/* float objfn() */
/*               */
/*****************/
/* evaluate l(eta) = ||z-c(eta)||^2 + lambda d1 d2 eta' Omega eta
   z is data and cfn is c(eta) (computed before passing to objfn) */

float objfn(cfn,z,lambda,eta,Omega,d1,d2,ne)
     float **cfn, **z, lambda, *eta, **Omega; int d1, d2, ne;
{
  int i,j,k; float x,*y;

  x=0;
  for(i=0;i<d1;i++)
    for(j=0;j<d2;j++)
      x+=(cfn[i][j]-z[i][j])*(cfn[i][j]-z[i][j]);

  if(lambda>0){
    y=vector(ne);
    for(i=0;i<ne;i++){
      y[i]=0;
      for(j=0;j<ne;j++)
        y[i]+=eta[j]*Omega[i][j];
    }
    for(i=0;i<ne;i++)
      x+=lambda*d1*d2*eta[i]*y[i];
    free(y);
  }

  return x;

} /* end objfn() */

/*******************/
/*                 */
/* void pseudoval() */
/*                 */
/*******************/
/* calculate pseudovalues for Gauss-Newton algorithm */

void pseudoval(zst,cfn,z,X,eta0,d1,d2,ne)
     float **zst, **cfn, **z, **X, *eta0;
     int d1,d2,ne;
{
  int i,j,k;

  for(i=0;i<d1;++i)
    for(j=0;j<d2;++j){
      zst[i][j]=z[i][j]-cfn[i][j];
      for(k=0;k<ne;++k)
        zst[i][j]+=X[i*d2+j][k]*eta0[k];
    }

} /* end pseudoval() */

/******************/
/*                */
/* void lsecalc() */
/*                */
/******************/
```

```
/* calculate least squares update (or estimate) with regularization
   eta = (X'X+lambda*d1*d2*Omega)^(-1) X'zst */

void lsecalc(eta,X,lambda,Omega,zst,d1,d2,ne)
     float *eta, **X, lambda, **Omega, **zst;
     int d1, d2, ne;
{
  int i,j,k,n=d1*d2;
  float **xtx, *evals, **evecs, *wv, **wm, *xtz0,
        alpha=1, beta=0;
  char u='u', t='n';
  FILE *ofl;

/*
  xtx = X(^T)X and later inverse of (xtx + lambda d1 d2 Omega)
  evals for storing eigenvalues of (xtx + lambda d1 d2 Omega) to invert
  evecs are corresponding eigenvectors
  wv,wm = work vector and matrix
  xtz0 = X'zst
  n,alpha,beta,u,t : parameters for ssyrk
*/

/* make xtx + lambda*d1*d2*Omega and invert
   to invert: first get eigenvalues and eigenvectors
   then get inverse as A = xLx', A{^-1} = xL{^-1}x'
   where x is eigenvecs, L is diagonal matrix of eigenvals
   use ssiev from Fortran cmlib (licepack and eispack)
*/

  xtx=matrix(ne,ne);

/* ssyrk from cmlib sblas3 */

  ssyrk_(&u,&t,&ne,&n,&alpha,&X[0][0],&ne,&beta,&xtx[0][0],&ne);
  for(i=0;i<ne;i++)
    for(j=i+1;j<ne;j++)
      xtx[i][j]=xtx[j][i];

  if(lambda>0)
    for(i=0;i<ne;++i)
      for(j=0;j<ne;++j)
        xtx[i][j]+=lambda*d1*d2*Omega[i][j];

  evals=vector(ne);
  evecs=matrix(ne,ne);
  wv=vector(2*ne);
  wm=matrix(ne,ne);
  for(i=0;i<ne;++i)
    for(j=0;j<ne;++j)
      evecs[i][j]=xtx[i][j];
  j=1;
  ssiev_(*evecs,&ne,&ne,evals,wv,&j,&i);
  for(i=0;i<ne;++i)
    for(j=0;j<ne;++j)
      wm[i][j]=evecs[j][i]/(evals[j]+1e-9);
  for(i=0;i<ne;++i)
    for(j=i;j<ne;++j){
      xtx[i][j]=0;
      for(k=0;k<ne;++k)
        xtx[i][j]+=wm[i][k]*evecs[k][j];
    }
  for(i=0;i<ne;++i)
```

```
    for(j=0;j<i;++j)
      xtx[i][j]=xtx[j][i];
  free(evals);
  free(*evecs); free(evecs);
  free(wv);
  free(*wm); free(wm);

/* make xtz0 and eta */

  xtz0=vector(ne);

  for(k=0;k<ne;++k){
    xtz0[k]=0;
    for(i=0;i<d1;++i)
      for(j=0;j<d2;++j)
        xtz0[k]+=X[i*d2+j][k]*zst[i][j];
  }
  for(i=0;i<ne;++i) {
    eta[i]=0;
    for(j=0;j<ne;++j)
      eta[i]+=xtx[i][j]*xtz0[j];
  }
  free(xtz0);
  free(*xtx); free(xtx);

} /* end lsecalc() */

/*****************/
/*               */
/* void stepfn() */
/*               */
/*****************/
/* find an optimal step function in the suggested descent direction
   minimize objective function:
   ||c(eta_rho)-z||^2 + lambda d1 d2 eta_rho' Omega eta_rho
   eta_rho = eta0 + rho(etahat-eta0)

   input:  cfn, z, d1, d2, psi, h, nc, eta0, etahat, ne, lambda, Omega
   output: etahat = eta_rho, rho, l(eta_rho)

   currently chooses from:
   0,1/128,1/64,1/32,1/16,1/8,1/4,1/2,1,2,4,8,16 */

void stepfn(leta,etahat,eta0,cfn,z,lambda,Omega,nc,psi,h,d1,d2,ne)
     float *leta, *etahat, *eta0, **cfn, **z, lambda, **Omega;
     int nc;
     float **psi, h;
     int d1, d2, ne;
{
#define STEPS 13

  float *etarho, *rho, *lofeta, step;
  int i,j,k;
  FILE *ofl;

  etarho=vector(ne);
  rho=vector(STEPS);
  lofeta=vector(STEPS);

  rho[0]=0;
  rho[STEPS/2+2]=1;
  lofeta[0]=leta[-1];
```

```
  lofeta[STEPS/2+2]=objfn(cfn,z,lambda,etahat,Omega,d1,d2,ne);

  for(i=1;i<STEPS;(i==STEPS/2+1)?i+=2:i++){
    rho[i]=pow(2.0,(float)(i-STEPS/2-2));
    for(j=0;j<ne;j++)
      etarho[j]=eta0[j]+rho[i]*(etahat[j]-eta0[j]);
    imconv(cfn,0,0,d1,d2,nc,psi,2.3548*h,etarho,ne);
    lofeta[i]=objfn(cfn,z,lambda,etarho,Omega,d1,d2,ne);
  }

  *leta=minv(lofeta,STEPS);
  step=rho[minvi(lofeta,STEPS)];
  if(step!=1)
    for(i=0;i<ne;i++)
      etahat[i]=eta0[i]+(step)*(etahat[i]-eta0[i]);

  free(etarho);
  free(rho);
  free(lofeta);

} /* end stepfn() */
```

A.3 Cell Parameter Estimation

In this section, the code for the functions estpsi, adjphi, derivpsi and steppsi is given. These functions contain the algorithms for estimating the cell specific parameters. The function estpsi calls adjphi to optimize the orientation angle over a number of selected choices, derivpsi to calculate the derivatives for all the cell parameters, and steppsi to choose the step length. The functions pseudoval, lsecalc and objfn, described in the previous section, are also used by estpsi.

```
/*****************/
/*               */
/* void estpsi() */
/*               */
/*****************/
/* estimate cell specific parameters

parameters: (* indicates updated by estpsi)
  *psi - matrix (6 x nc) of cell specific parameters:
            mu1,mu2 - cell location in pixel units
              alpha - size in pixel units
               beta - aspect ratio
                phi - orientation angle [0,360]
              gamma - intensity
   eta - vector (ne) containing shape curve
   h - Gaussian blurring parameter (standard deviation)
       might switch to fwhm
   z - matrix (d1 x d2) containing image
   d1,d2 - image dimensions
   nc - number of cells
   ne - dimension of shape curve vector
*/
```

```
void estpsi(psi,eta,h,z,d1,d2,nc,ne,pte,ni,nstep)
     float **psi, *eta, h, **z;
     int d1, d2, nc, ne, *pte, ni, nstep;
{
  float *p0, *p1, *lpsi, mse, msl, **cfn, **X, **zst, temp, *gmma, usg=0,
        *pt, **xt;
  int i, j, k, m, nti, ipr, ncu, ngu, n, nctu, ngl;
  FILE *ofl;

#define NP 6

  nti=pte[6]; /* number of parameters to iterate (1-6) */
  ncu=5; /* max number of cells to update at once */

  if((nti<1)||(nti>6)){
    fprintf(stderr,"need pte[6] between 1 and 6 in estpsi\n");
    exit();
  }

  if(ncu>nc) ncu=nc;

  p0=vector(nti*nc);
  p1=vector(nti*nc);
  lpsi=vector(ni);
  cfn=matrix(d1,d2);
  X=matrix(d1*d2,nti*nc);
  zst=matrix(d1,d2);
  gmma=vector(nc);
  pt=vector(nti*ncu);
  xt=matrix(d1*d2,nti*ncu);

  ipr=0;
  for(j=0;j<6;j++){
    if(pte[j]){
      for(i=0;i<nc;i++)
        p0[i+ipr*nc]=psi[j][i];
      ipr+=1;
    }
  }

  for(i=0;i<=ni;i++)
    lpsi[i]=0;
  mse=1;
  msl=1;

/* Gauss-Newton iteration */

  for(k=1;(k<=ni)&&(mse>1e-05)&&(msl>1e-05||mse>1);k++){

/* adjust orientation */

    ngl=0;
    if(k<=10) ngl=1;
    if(k<=5) ngl=2;
    if(k<=3) ngl=3;
    if(k==1) ngl=4;
    if(pte[4]&&ngl>0){
      imconv(cfn,0,0,d1,d2,nc,psi,2.3548*h,eta,ne);
      adjphi(psi,cfn,z,h,eta,d1,d2,ne,nc,ngl);
      for(i=0;i<nc;i++)
        p0[i+(pte[0]+pte[1]+pte[2]+pte[3])*nc]=psi[4][i];
```

```
    }

    imconv(cfn,0,0,d1,d2,nc,psi,2.3548*h,eta,ne);
    derivpsi(X,cfn,psi,h,eta,d1,d2,ne,nc,pte);
    if(k==1) lpsi[0]=objfn(cfn,z,0.0,eta,0,d1,d2,ne);

    ngu=(int)floor((float)nc/(float)ncu+.999);
    for(m=0;m<ngu;m++){
      nctu=0;
      for(n=ncu*m;n<ncu*(m+1)&&n<nc;n++){
        nctu+=1;
        for(ipr=0;ipr<nti;ipr++){
          pt[ipr+(n-m*ncu)*nti]=p0[n+ipr*nc];
          for(i=0;i<d1;i++)
            for(j=0;j<d1;j++)
              xt[i*d2+j][ipr+(n-m*ncu)*nti]=X[i*d2+j][n+ipr*nc];
        }
      }
      pseudoval(zst,cfn,z,xt,pt,d1,d2,nti*ncu);
      lsecalc(pt,xt,0.0,0,zst,d1,d2,nti*ncu);
      for(n=ncu*m;n<ncu*(m+1)&&n<nc;n++)
        for(ipr=0;ipr<nti;ipr++)
          p1[n+ipr*nc]=pt[ipr+(n-m*ncu)*nti];
    }

    steppsi(&lpsi[k],p1,p0,cfn,z,eta,psi,h,d1,d2,ne,nc,pte,nstep);

/* constrain */
/* location in [0..d1,0..d2] */
/* size in [0,d] where d=dimension/2 */
/* aspect ratio in [0,1] */
/* orientation in [0,360) */
/* intensity in [1,400] */

    ipr=0;
    if(pte[0]){
      for(i=0;i<nc;i++){
        if(p1[i+ipr*nc]<0)  p1[i+ipr*nc]=5; /* location */
        if(p1[i+ipr*nc]>d1) p1[i+ipr*nc]=d1-5;
      }
      ipr+=1;
    }
    if(pte[1]){
      for(i=0;i<nc;i++){
        if(p1[i+ipr*nc]<0)  p1[i+ipr*nc]=5;
        if(p1[i+ipr*nc]>d2) p1[i+ipr*nc]=d2-5;
      }
      ipr+=1;
    }
    if(pte[2]){
      for(i=0;i<nc;i++){
        if(p1[i+ipr*nc]<2) p1[i+ipr*nc]=5; /* size */
        if(p1[i+ipr*nc]>(d1+d2)/4) p1[i+ipr*nc]=(d1+d2)/6;
      }
      ipr+=1;
    }
    if(pte[3]){
      for(i=0;i<nc;i++){
        if(p1[i+ipr*nc]>1) p1[i+ipr*nc]=0.8; /* aspect ratio */
        if(p1[i+ipr*nc]<0.05) p1[i+ipr*nc]=0.1;
      }
      ipr+=1;
```

```
    }
    if(pte[4]){
      for(i=0;i<nc;i++){
        while(p1[i+ipr*nc]>=360) p1[i+ipr*nc]-=360; /* orientation */
        while(p1[i+ipr*nc]<0)   p1[i+ipr*nc]+=360;
      }
      ipr+=1;
    }
    if(pte[5]){
      for(i=0;i<nc;i++){
        if(p1[i+ipr*nc]<10) p1[i+ipr*nc]=20;
        if(p1[i+ipr*nc]>400) p1[i+ipr*nc]=300;
      }
      ipr+=1;
    }

/* reset psi */

    ipr=0;
    for(j=0;j<6;j++){
      if(pte[j]){
        for(i=0;i<nc;i++)
          psi[j][i]=p1[i+ipr*nc];
        ipr+=1;
      }
    }
    mse=0;
    for(i=0;i<nti*nc;i++){
      temp=(p1[i]-p0[i]);/* /(p0[i]+1e-6); */
      mse+=sqr(temp);
    }
    mse=mse/(nti*nc);
    msl=(lpsi[k-1]-lpsi[k])/(1+absv(lpsi[k]));

    for(i=0;i<nti*nc;i++)
      p0[i]=p1[i];

  } /* end Gauss-Newton iteration */

  free(p0);
  free(p1);
  free(lpsi);
  free(*cfn); free(cfn);
  free(*X); free(X);
  free(*zst); free(zst);
  free(gmma);
  free(pt);
  free(*xt); free(xt);

} /* end estpsi() */

/******************/
/*                */
/* void steppsi() */
/*                */
/******************/
/* find an optimal step function in suggested descent direction
   minimize objective function: ||c(phi_rho)-z||^2

   input:  cfn, z, d1, d2, psi, h, nc, phi1, phi0, eta, ne,
   output: phihat = phi_rho, rho, l(phi_rho)
```

```
   steps chosen from (as a function of number of steps)
   note: each time add the current number to the previous set
         e.g.: nstep == 1: 0
               nstep == 2: 0,1/4

       nstep: 1 2   3   4   5 6    7 8    9 10
   add to set: 0 1/4 1/2 1/8 1 1/16 2 1/32 4 1/64

       nstep  11 12    13 14    15 16    17 18     19  20
   add to set: 8  1/128 16 1/256 32 1/512 64 1/1024 128 1/2048

   param: 0 - mu_1  - location (x,i)
          1 - mu_2  - location (y,j)
          2 - alpha - size
          3 - beta  - aspect ratio
          4 - phi   - orientation
          5 - gamma - intensity (never used)
*/

void steppsi(lpsi,p1,p0,cfn,z,eta,psi,h,d1,d2,ne,nc,pte,nstep)
     float *lpsi, *p1, *p0, **cfn, **z, *eta, **psi, h;
     int d1, d2, ne, nc, *pte, nstep;
{

  float **psirho, *rho, *lofpsi, step;
  int i,j,k, ipr, nti;
  FILE *ofl;

  nti=pte[6];
  psirho=matrix(NP,nc);

  if(nstep==0){
    ipr=0;
    for(k=0;k<6;k++){
      if(pte[k]){
        for(j=0;j<nc;j++)
          psirho[k][j]=p1[j+ipr*nc];
        ipr+=1;
      }
    }
    imconv(cfn,0,0,d1,d2,nc,psirho,2.3548*h,eta,ne);
    *lpsi=objfn(cfn,z,0.0,eta,0,d1,d2,ne);
    step=1;
  }
  else{
    rho=vector(nstep);
    lofpsi=vector(nstep);

    for(i=0;i<NP;i++)
      for(j=0;j<nc;j++)
        psirho[i][j]=psi[i][j];

    rho[0]=0;
    rho[nstep/2+2]=1;
    lofpsi[0]=lpsi[-1];

    for(i=1;i<nstep;i++){
      rho[i]=pow(2.0,(float)(i-nstep/2-2));
      ipr=0;
      for(k=0;k<6;k++){
        if(pte[k]){
          for(j=0;j<nc;j++)
```

```
            psirho[k][j]=p0[j+ipr*nc]+rho[i]*(p1[j+ipr*nc]-p0[j+ipr*nc]);
          ipr+=1;
        }
      }
      imconv(cfn,0,0,d1,d2,nc,psirho,2.3548*h,eta,ne);
      lofpsi[i]=objfn(cfn,z,0.0,eta,0,d1,d2,ne);
    }

    *lpsi=minv(lofpsi,nstep);
    step=rho[minvi(lofpsi,nstep)];
    if(step!=1)
      for(i=0;i<nti*nc;i++)
        p1[i]=p0[i]+(step)*(p1[i]-p0[i]);

    free(rho);
    free(lofpsi);
  }

  free(*psirho); free(psirho);

} /* end steppsi() */

/*******************/
/*                 */
/* void derivpsi() */
/*                 */
/*******************/
/* calculate derivatives for cell specific parameters
   output: X = derivative matrix */

void derivpsi(X,cfn,psi,h,eta,d1,d2,ne,nc,pte)
     float **X, **cfn, **psi, h, *eta;
     int d1, d2, ne, nc, *pte;
{
  int i,j,k,m, ipr;
  float **cfnd, **psid, d, ***f0, ***f1, ***f2, ***f3, ***f4, ***f5,
        **y1, **psic;
  FILE *ofl;
#define PI 3.141593

  f0=f3tensor(nc,d1,d2);
  f1=f3tensor(nc,d1,d2);
  f2=f3tensor(nc,d1,d2);
  f3=f3tensor(nc,d1,d2);
  f4=f3tensor(nc,d1,d2);
  y1=matrix(d1,d2);
  psic=matrix(6,1);

/* calculate necessary convolutions */

  if(pte[6]>0)
    for(k=0;k<nc;k++){ /* for all parameters */
      for(m=0;m<6;m++)
        psic[m][0]=psi[m][k];
      imconv(f0[k],0,0,d1,d2,1,psic,2.3548*h,eta,ne);
    }
  if(pte[0]||pte[2]||pte[3]||pte[4]) /* for mu1,alpha,beta,phi */
    for(k=0;k<nc;k++){
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          y1[i][j]=i-psi[0][k]+1e-6;
      for(m=0;m<6;m++)
```

```
        psic[m][0]=psi[m][k];
      imconv(f1[k],y1,1,d1,d2,1,psic,2.3548*h,eta,ne);
    }
  if(pte[1]||pte[2]||pte[3]||pte[4]) /* for mu2,alpha,beta,phi */
    for(k=0;k<nc;k++){
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          y1[i][j]=j-psi[1][k]+1e-6;
      for(m=0;m<6;m++)
        psic[m][0]=psi[m][k];
      imconv(f2[k],y1,1,d1,d2,1,psic,2.3548*h,eta,ne);
    }
  if(pte[2]) /* for alpha */
    for(k=0;k<nc;k++){
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          y1[i][j]=pow(i-psi[0][k],2.0)+pow(j-psi[1][k],2.0);
      for(m=0;m<6;m++)
        psic[m][0]=psi[m][k];
      imconv(f3[k],y1,1,d1,d2,1,psic,2.3548*h,eta,ne);
    }
  if(pte[3]) /* for beta */
    for(k=0;k<nc;k++){
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          {
            y1[i][j]=(j-psi[1][k])*cos(psi[4][k]*PI/180)
              -(i-psi[0][k])*sin(psi[4][k]*PI/180);
            y1[i][j]*=y1[i][j];
          }
      for(m=0;m<6;m++)
        psic[m][0]=psi[m][k];
      imconv(f4[k],y1,1,d1,d2,1,psic,2.3548*h,eta,ne);
    }

  ipr=0;

/* derivative for location (x) */

  if(pte[0]){
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=
            ((i-psi[0][k])*f0[k][i][j]-f1[k][i][j])/(h*h);
    ipr+=1;
  }

/* derivative for location (y) */

  if(pte[1]){
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=
            ((j-psi[1][k])*f0[k][i][j]-f2[k][i][j])/(h*h);
    ipr+=1;
  }

/* derivative for size */

  if(pte[2]){
```

```
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=
            f0[k][i][j]*2/psi[2][k]+((i-psi[0][k])*f1[k][i][j]
              +(j-psi[1][k])*f2[k][i][j]-f3[k][i][j])/(h*h*psi[2][k]);
    ipr+=1;
  }

/* derivative for aspect ratio */

  if(pte[3]){
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=
            f0[k][i][j]/psi[3][k]+
              ((cos(psi[4][k]*PI/180)*(j-psi[1][k]) - sin(psi[4][k]*PI/180)*
               (i-psi[0][k]))*(cos(psi[4][k]*PI/180)*f2[k][i][j]
                -sin(psi[4][k]*PI/180)*f1[k][i][j])-f4[k][i][j])/
                  (psi[3][k]*h*h);
    ipr+=1;
  }

/* derivative for orientation */

  if(pte[4]){
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=(f1[k][i][j]*(j-psi[1][k])
                               -f2[k][i][j]*(i-psi[0][k]))/(h*h*180/PI);
    ipr+=1;
  }

/* derivative for intensity */

  if(pte[5]){
    for(k=0;k<nc;k++)
      for(i=0;i<d1;i++)
        for(j=0;j<d2;j++)
          X[i*d2+j][k+ipr*nc]=f0[k][i][j]/psi[5][k];
    ipr+=1;
  }

  free(**f0); free(*f0); free(f0);
  free(**f1); free(*f1); free(f1);
  free(**f2); free(*f2); free(f2);
  free(**f3); free(*f3); free(f3);
  free(**f4); free(*f4); free(f4);
  free(*y1); free(y1);
  free(*psic); free(psic);

} /* end derivpsi() */

/*****************/
/*               */
/* void adjphi() */
/*               */
/*****************/
/* adjust orientation +/-180 if necessary
   hopefully will protect against local minima
```

```
   output: psi is adjusted if necessary
   ngl=number of guess levels:
        0 - none
        1 - 1/2
        2 - add 1/4, 3/4
        3 - add 1/8,...,7/8
        4 - add 1/16,...,15/16
        etc...
*/

void adjphi(psi,cfn,z,h,eta,d1,d2,ne,nc,ngl)
     float **psi, **cfn, **z, h, *eta;
     int d1, d2, ne, nc, ngl;
{
  float **psit, lphi0, **lphi, **phit, *g;
  int i, j, k, ng;
  FILE *ofl;

  ng=pow(2.0,(float)ngl)-1;

  psit=matrix(6,nc);
  lphi=matrix(nc,ng);
  phit=matrix(nc,ng);
  g=vector(ng);

  for(i=0;i<6;i++)
    for(j=0;j<nc;j++)
      psit[i][j]=psi[i][j];

  imconv(cfn,0,0,d1,d2,nc,psit,2.3548*h,eta,ne);
  lphi0=objfn(cfn,z,0.0,eta,0,d1,d2,ne);

  for(j=1;j<=ng;j++)
    g[j-1]=360*j/(ng+1);

  for(i=0;i<nc;i++)
    for(j=0;j<ng;j++){
      phit[i][j]=psi[4][i];
      phit[i][j]+=g[j];
      if(phit[i][j]>360)
        phit[i][j]-=360;
    }

  for(i=0;i<nc;i++)
    for(j=0;j<ng;j++){
      psit[4][i]=phit[i][j];
      imconv(cfn,0,0,d1,d2,nc,psit,2.3548*h,eta,ne);
      lphi[i][j]=objfn(cfn,z,0.0,eta,0,d1,d2,ne);
      psit[4][i]=psi[4][i];
    }

  k=0;
  for(i=0;i<nc;i++){
    j=minvi(lphi[i],ng);
    if(lphi[i][j]<lphi0){
      k++;
      psi[4][i]=phit[i][j];
    }
  }

  free(*psit); free(psit);
  free(*lphi); free(lphi);
```

```
  free(*phit); free(phit);
  free(g);
} /* end adjphi() */
```

A.4 Image Convolution

In this section, the code for the functions imconv, pixel and cellshape is given. These functions contain the algorithms for computing the convolutions necessary for calculating the cell function and derivatives in the shape function and cell parameter estimation algorithms. The function imconv calls pixel to compute the unblurred contribution to a given pixel for each cell, and pixel calls cellshape to incorporate the characteristic shape function in the computation.

```
/*****************/
/*               */
/* void imconv() */
/*               */
/*****************/
/* create image of cells with given parameters, convolved with
   a given function and blurred

parameters: (* indicates changed by imconv)

      *z - d1 x d2 matrix for image
          (or d1*d2 vector, with d2 index counting first, by C convention)
       fy - d1xd2 function to multiply by before blurring
       fyf - flag: 0 means don't use fy
       d1,d2 - horizontal (x), vertical (y) image dimensions, rotated 90^0
       nc - number of cells
       psi - cell specific parameters (x,y,a,b,phi,g) (np x nc)
             x,y - coordinates of center of cell in image
               a - size (length of long semiaxis)
               b - aspect ratio (ratio of perpendicular to long axes)
             phi - orientation [0,360)
               g - intensity
       fwhm - blurring parameter
       eta, J - shape function vector and its dimension
       *wv - "work" vector for gblurr
                 wv[7*2*maxd+15+2*2*maxd], where maxd=max(d1,d2)
                 wr[7*2*maxd+15],wc[2*2*maxd]
*/

void imconv(z,fy,fyf,d1,d2,nc,psi,fwhm,eta,J)
     float **z, **fy; int fyf, d1, d2, nc;
     float **psi, fwhm, *eta; int J;
{
  int i,j,k;
  float *x, *y, *a, *b, *phi, *g;
  float fwhm1=fwhm, fwhm2=fwhm, fwhm3=fwhm;
  int d3=1;
  float maxeta; int l1,l2,h1,h2;
```

```
  float *wr,*wc;

/* assign vectors x,y,a,b,phi,g to parts of matrix psi */

  x=&psi[0][0]; y=&psi[1][0];
  a=&psi[2][0]; b=&psi[3][0];
  phi=&psi[4][0]; g=&psi[5][0];

/* generate "true" image */

  for(i=0;i<d1;++i)
    for(j=0;j<d2;++j)
      z[i][j]=0.0;
  maxeta=maxv(eta,J);
  for(k=0;k<nc;++k){
    /* only need to look in (maxeta*a)^2 region around cell center */
    l1=(int)floor(x[k]-maxeta*a[k]);
    l2=(int)floor(y[k]-maxeta*a[k]);
    h1=(int)ceil(x[k]+maxeta*a[k]);
    h2=(int)ceil(y[k]+maxeta*a[k]);
    if(l1<0) l1=0;
    if(l2<0) l2=0;
    if(h1>d1) h1=d1;
    if(h2>d2) h2=d2;
    for(i=l1;i<h1;i++)
      for(j=l2;j<h2;j++)
        z[i][j] += pixel(x[k],y[k],a[k],b[k],phi[k],g[k],eta,i,j,J);
  }

/* multiply by function */

  if(fyf!=0)
    for(i=0;i<d1;++i)
      for(j=0;j<d2;++j)
        z[i][j]*=fy[i][j];

/* blurring */

  if(fwhm>0.0){
    i=(d1>d2)?d1:d2;
    wr=vector(7*2*i+15);
    wc=vector(2*2*i);
    gblurr_(&fwhm1,&fwhm2,&fwhm3,&d2,&d1,&d3,*z,wr,wc);
    /* switch d1 and d2 due to diffs btwn C and Fortran */
    /* d3 always 1 so doesn't matter here */
    free(wr);
    free(wc);
  }

} /* end imconv() */

/*****************/
/*               */
/* float pixel() */
/*               */
/*****************/
/* compute contribution of cell to a given pixel on the image */

float pixel(x,y,a,b,phi,g,eta,i,j,J)
     float x,y,a,b,phi,g,*eta; int i,j,J;
{
  float u,v; /* horiz,vert dist from center on transformed coord syst */
```

```
  float r;   /* relative distance from cell center (0) to edge (1) */
  float z;   /* pixel intensity */

/* (x,y) becomes pixel location relative to origin
   location of pixel (i,j) is (i+0.5,j+0.5)
   i&j index x&y respectively
*/
  x = i+0.5-x; y = j+0.5-y;
  phi *= PI/180; /* degrees to radians */
  u = x*cos(phi) + y*sin(phi);     /* rotate so angle of first */
  v = (-x*sin(phi)) + y*cos(phi);  /* principal axis is zero */

/* find relative distance (r) of pixel
   from cell center(0) to edge(1)
   note:  (u/a)^2 + (v/(ab))^2 = r^2
*/
  r = sqrt(sqr(u/a)+sqr(v/(a*b)));

/* add shape function */

  z = cellshape(r,g,a,b,u,v,eta,J);
  return z;

} /* end pixel() */

/*********************/
/*                   */
/* float cellshape() */
/*                   */
/*********************/
/* function to add cell shape function to cell simulation
   used in pixel()

  inputs: r - relative distance from cell center to pixel
          g - cell intensity
          a,b - principal semiaxes
          u,v - coordinates of pixel relative to cell center
          eta - cell shape function vector
          J - length of eta
*/

float cellshape(r,g,a,b,u,v,eta,J)
     float r,g,a,b,u,v,*eta; int J;
{
  float z, t;
  int j;
/*
  z - pixel intensity
  t - angle of pixel relative to cell (calculated from a,b,u,v)
  j - correct component of shape function eta
*/

/* prepare t on scale [0,2*pi) */

  t = atan((v/(a*b+1e-6))/(u/(a+1e-6)+1e-6));
  if (u==0 && v==0) t = 0;
  if (u < 0) t += PI;
  if (u>0 && v<0) t += 2*PI;
  while (t >= (2*PI)) t -= 2*PI;
  while (t < 0.0) t += 2*PI;

/* get appropriate component of shape curve vector */
```

```
  if (t>=(2*PI-PI/J)) t -= 2*PI;     /* t in [-pi/J,2pi-pi/J) */
  j = floor((t+PI/J)*J/(2*PI));

/* set pixel value to g if in cell with given shape function */

  z = (r<=eta[j]) ? g : 0;
  return(z);

} /* end cellshape() */
```

A.5 Other Supporting Functions

Various C functions and Fortran routines are utilized. In this section, a brief description is given of the C utility functions and the Fortran routines. The appropriate sources are also noted.

A.5.1 C Utility Functions

Several other functions were written in the C language for utility purposes. The functions minv, minvi, maxv, maxvi, sqr, absv and mtranm are original and vector, matrix and f3tensor are modified from routines in *Numerical Recipes in C* [76]. The functions minv and maxv find the minimum and maximum values of a vector, while minvi and maxvi find the vector locations corresponding to those values. The functions sqr and absv are actually one line definitions to calculate the square and absolute values respectively. The function mtranm returns the matrix equal to the transpose of a matrix times itself: $X^T X$. The functions vector, matrix and f3tensor are the same as the corresponding functions in *Numerical Recipes in C* [76] except that the ability to offset has been eliminated, so only the dimensions need to be specified.

A.5.2 Fortan Subroutines

Fortran subroutines can be incorporated into programs written in the C language, at least on the SunOS and DEC Unix platforms. The subroutines gblurr, ssiev and ssyrk are used in some of the programs listed in the previous sections. The subroutine gblurr is used in the imconv function to perform the Gaussian blurring. This subroutine computes the convolution using Fast Fourier Transform (FFT) method. The

subroutines **cffti**, **cfftf** and **cfftb** from FFTPACK, a fast Fourier transform package by P. N. Swarztrauber, are used in **gblurr**. FFTPACK can currently be obtained by itself from Netlib or as a part of CMLIB from Statlib. The subroutine **ssiev**, by D. K. Kahaner, C. B. Moler and G. W. Stewart, is used for calculating eigenvalues and eigenvectors in **lsecalc** and is part of the LICEPACK package available as part of CMLIB from Statlib. The subroutine **ssyrk**, by J. Dongarra et al., is used for multiplying matrices and is part of the SBLAS3 package available from Netlib.

VITA

James Francis Lymp was born at the United States Naval Weapons Center, China Lake, California on December 9, 1969. He obtained his primary and secondary education from various schools in Hillsboro and Yamhill, Oregon, graduating from Yamhill-Carlton Union High School in 1988. Jim attended Oregon State University in Corvallis from 1988-1992 and received a Bachelor of Science, *cum laude*, in Mathematics. He attended the University of Washington in Seattle from 1992-1997 and received 2 degrees in the Department of Biostatistics (Master of Science, 1994 and Doctor of Philosophy, 1997). Upon completion of his Ph.D. work, Jim plans to pursue a post-doctoral fellowship under the direction of Finbarr O'Sullivan at University College Cork, Ireland.