Bayesian Level Sets and Texture Models for Image Segmentation and Classification with Application to Non-Invasive Stem Cell Monitoring

by
Nathan Christopher Lowry

B.S.M.E., Rice University (2001)
M.S.M.E., Rice University (2004)

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Doctor of Science in Aeronautics and Astronautics at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

© Nathan Christopher Lowry, MMXIII. All rights reserved.
The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author

Department of Aeronautics and Astronautics

May 10, 2013

Certified by

Youssef M. Marzouk
Class of 1942 Associate Professor, MIT
Thesis Supervisor

Certified by

Ramf S. Mangoubi
Principal Member of the Technical Staff, C.S. Draper Laboratory
Thesis Supervisor

Accepted by

Eytan H. Modiano
Professor of Aeronautics and Astronautics, MIT
Chair, Graduate Program Committee
Thesis Committee Members

Youssef M. Marzouk, Ph.D.
Class of 1942 Associate Professor
Massachusetts Institute of Technology

Rami S. Mangoubi, Ph.D.
Principal Member of the Technical Staff
C.S. Draper Laboratory

Laurent Demanet, Ph.D.
Assistant Professor of Applied Mathematics
Massachusetts Institute of Technology

Mukund N. Desai, Ph.D.
Distinguished Member of the Technical Staff
C.S. Draper Laboratory

Paul J. Sammak, Ph.D.
Scientific Review Officer
National Institute of Health
Bayesian Level Sets and Texture Models for Image Segmentation and Classification with Application to Non-Invasive Stem Cell Monitoring

by

Nathan Christopher Lowry

Submitted to the Department of Aeronautics and Astronautics on May 10, 2013, in partial fulfillment of the requirements for the degree of Doctor of Science in Aeronautics and Astronautics

Abstract

Image segmentation and classification, the identification and demarcation of regions of interest within an image, is necessary prior to subsequent information extraction, analysis, and inference. Many available segmentation algorithms require manual delineation of initial conditions to achieve acceptable performance, even in cases with high signal to noise and interference ratio that do not necessitate restoration. This weakness impedes application of image analysis to many important fields, such as automated, mass scale cultivation and non-invasive, non-destructive high-throughput analysis, monitoring, and screening of pluripotent and differentiated stem cells, whether human embryonic (hESC), induced pluripotent (iPSC), or animal.

Motivated by this and other applications, the Bayesian Level Set (BLS) algorithm is developed for automated segmentation and classification that computes smooth, regular segmenting contours in a manner similar to level sets while possessing a simple, probabilistic implementation similar to that of the finite mixture model EM. The BLS is subsequently extended to harness the power of image texture methods by incorporating learned sets of class-specific textural primitives, known as textons, within a three-stage Markov model. The resulting algorithm accurately and automatically classifies and segments images of pluripotent hESC and trophectoderm colonies with 97% and 91% accuracy for high-content screening applications and requires no previous human initialization. While no prior knowledge of colony class is assumed, the framework allows for its incorporation. The BLS is also validated on other applications, including brain MRI, retinal lesions, and wildlife images.

Thesis Supervisor: Youssef M. Marzouk
Title: Class of 1942 Associate Professor, MIT

Thesis Supervisor: Rami S. Mangoubi
Title: Principal Member of the Technical Staff, C.S. Draper Laboratory
Then Samuel took a stone, and set it between Mizpeh and Shen, and called the name of it Ebenezer, saying, Hitherto hath the LORD helped us.

I Samuel 7:12.
Acknowledgments

I am most fortunate to have an incredible thesis advisor at Draper, Dr. Rami Mangoubi. Rami believed I had potential when we worked together on an industry project and graciously invited me to become his student as he began the leadership of his stem cell image analysis R01 grant. Since then, he has been a keen mentor and strong advocate for my studies and research. This thesis would not have been written without his guidance, and I feel fortunate for the opportunity to continue collaborating with him after the conclusion of graduate study. Rami’s insistence that research originates in real problems and requires new methods speaks to his high standards.

In addition, I would like to thank my advisor at MIT, Prof. Youssef Marzouk, who graciously agreed to advise a student researching image processing. His cheerful and patient guidance was invaluable in navigating MIT.

Thank you to Dr. Mukund Desai, not only for serving on my thesis committee but for the many conversations and brainstorming sessions. The texture content of this thesis has particularly benefited from his insight.

Thank you to Dr. Paul Sammak, who not only provided high-quality image data for analysis but patiently and clearly explained stem cell biology to engineers. The biological application of my research was motivated by the knowledge Paul imparted and his advocacy of image-based biomarkers.

Thank you to Prof. Laurent Demanet for serving on my thesis committee, and to Dr. John Irvine for acting as a reader. My special thanks to Dr. Michael Prange for acting as an emergency reader and Dr. Tarek Habashy for introducing us. I appreciate very much the time and helpful comments you have all provided.

Thank you also to Prof. Gustavo Rohde, who graciously reviewed my defense the night before its delivery. The presentation (and my confidence in giving it) were stronger for his comments.

Thank you to Dr. Igor Linkov and Prof. Matteo Convertino for inviting me to collaborate on their biodiversity research. And thanks also to Dr. Alexander Tkachuk, for introducing us and for many stimulating conversations on decision theory.
I owe thanks to a great many people at Draper Laboratory. I particularly thank Linda Fuhrman, Steve Kolitz, and Sharon Donald for generously supporting my thesis; the laboratory is fortunate to have officers who value the development of research skills. Thanks to Pam Toomey, an artist with a gift for designing conference posters which are both informative and aesthetically appealing. Many thanks also to Heidi Perry. I have often heard from Rami how she values good researchers, and I am grateful that, after I expressed an interest in continuing at Draper Laboratory following my studies, she acted with alacrity to ensure that I would be rehired. Thanks also to Dr. Leena Singh for her interest in my research and to Dr. Piero Miotto, whose advice and recommendation at the beginning of my studies was especially helpful.

Thank you to all my friends for your support. I particularly thank Matt Agen, Josh Ketterman, and Nick Mulherin, for conversation, games, and good humor. And, I specially thank Rob Rodger, an old and true friend who recently gave up a weekend to sit by a table in the woods so that I might be engaged.

Thank you also to my future in-laws, the Gabrielse family, who have welcomed me into their family and kindly attended my defense in support.

I am very, very grateful for the steadfast love and support my family has given me during my years in graduate school. Thank you to my parents, Dr. J. Michael and Jolynn Lowry, and to my sister Sarah, her husband Steven Meek, and their new daughter Grace. Your words of encouragement during pre-quals jitters, in the midst of late-night coding sessions, and this final year of research and writing were a very real source of strength.

And, finally, I would like to thank my wonderful fiancée, Abigail Gabrielse. I resolved to ask her out to dinner the very day I submitted my thesis proposal and proposed to her three weeks prior to my defense. In her kindness and love, she served as the first proof reader for this document and sent many a home-cooked meal with me to the lab. To me she has been a great encouragement and a great joy, and I feel incredibly blessed that we are to be married. Many women do noble things, but you surpass them all. Abigail, sans toi, à qui sourire? Pour qui vivre?
Assignment

Draper Laboratory Report Number T-1744

In consideration for the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Nathan C. Lowry

Date

May 27, 2013

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., with support by NIH grant 1 R01 EB006161-01 and by an internal research and development grant from Draper Laboratory.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.
## Contents

### 1 Introduction

1.1 Need for automated segmentation and classification .......................... 24
1.2 Need for non-invasive biomarkers for stem cell monitoring ................. 26
  1.2.1 Importance of stem cells ................................................. 26
  1.2.2 Current practices ......................................................... 28
  1.2.3 Image analysis as a non-invasive, non-destructive stem cell biomarker 29
1.3 Contributions ................................................................. 29
  1.3.1 Contributions to image segmentation and texture modeling ........... 30
  1.3.2 Contributions to non-invasive, quantitative, image-based stem cell biomarkers ............................................. 32
1.4 Organization ........................................................................... 32

### 2 Level Sets and Expectation-Maximization: A Comparative Analysis 35

2.1 Notation and conventions ....................................................... 36
2.2 Level sets ............................................................................. 38
  2.2.1 History of level sets ........................................................ 38
  2.2.2 Region-based level sets .................................................... 40
  2.2.3 Further development ....................................................... 44
2.3 Expectation Maximization (EM) of finite mixture models ................. 45
  2.3.1 Development and theory .................................................. 46
  2.3.2 Finite mixture model EM ................................................. 47
  2.3.3 Use in image segmentation .............................................. 49
2.4 Stochastic level set-like algorithms .......................................... 50
2.5 Conclusions and commentary ........................................ 51

3 Bayesian Level Sets for Image Segmentation .......................... 55
  3.1 Formulating the cost function ..................................... 55
    3.1.1 The cost function ............................................ 56
    3.1.2 Interpretation and insight .................................. 57
  3.2 A Bayesian identity for minimizing the cost function .......... 59
  3.3 A Bayesian level set algorithm for multiclass image segmentation ... 61
  3.4 Geometric and smoothing priors .................................. 65
  3.5 Advantages of the BLS ............................................. 68
  3.6 The BLS and the EM ................................................ 70
  3.7 Example applications of the BLS .................................. 71
    3.7.1 A simple example ............................................. 71
    3.7.2 Wavelet texture-based segmentation of induced pluripotent stem
cells ................................................................. 73
    3.7.3 Noisy brain MRI ................................................. 77
    3.7.4 Massively multiclass segmentation of stem cell nuclei via hard
classification ........................................................ 79
  3.8 Summary .............................................................. 83

4 The Power of Texture for Image Classification ....................... 85
  4.1 Historical review of image texture modeling ...................... 86
    4.1.1 Spatial domain neighborhood models ........................ 88
    4.1.2 Frequency domain scale models ............................. 90
  4.2 Wavelet-based texture classification of stem cell patches ....... 91
    4.2.1 Wavelet-based texture analysis ............................. 92
    4.2.2 Application to nuclei classification ........................ 96
    4.2.3 Application to colony classification via windowing ........ 100
  4.3 Texture features for stem cell colony segmentation .............. 104
    4.3.1 Localizing texture models for image segmentation .......... 104
    4.3.2 Wavelet-based features (dcmag, dcmx, and wave) .......... 106
4.3.3 Other filter-based features (abar, gauss, log) .......................... 108
4.3.4 Entropy-based features (hdiff, hloc) ................................. 110
4.3.5 Compressive sensing features (cs) .................................... 112
4.3.6 Comparative summary .................................................. 113
4.4 Summary ....................................................................... 115

5 Texture-based Segmentation using Bayesian Level Sets ............ 117
5.1 The texture-based Bayesian Level Set algorithm .................. 117
  5.1.1 Three-stage Markov texture model ................................. 118
  5.1.2 Adapting the three-stage Markov model in the BLS framework 122
  5.1.3 Geometric prior: combine smoothing and mixture ............ 123
  5.1.4 Conjugate data priors: Normal-inverse-Wishart and Dirichlet. 125
  5.1.5 The texture-based Bayesian Level Set algorithm ............... 129
5.2 Training ........................................................................ 130
  5.2.1 Feature selection ......................................................... 132
  5.2.2 Feature reduction via Principle Component Analysis (PCA) . 136
  5.2.3 Texton learning via GMM-EM clustering ......................... 139
5.3 Initialization ................................................................. 139
5.4 Case Study: Zebras ....................................................... 141
5.5 Summary ....................................................................... 142

6 Texture-Based BLS Applied to Non-Invasive Stem Cell Colony Seg-
mentation and Classification ......................................................... 143
6.1 Motivation and impact ..................................................... 143
6.2 Virtual staining ............................................................. 145
6.3 Data collection .............................................................. 145
6.4 Colony image segmentation ............................................. 145
  6.4.1 Experimental procedure .............................................. 146
  6.4.2 Numerical results ....................................................... 146
  6.4.3 Evolution of a single BLS segmentation ......................... 149
  6.4.4 Texton training for differentiated colonies ...................... 149
6.5 Multi-class image phantom segmentation .................................. 161
6.6 Recognizing anomalous texture classes ....................................... 163
6.7 Conclusions ........................................................................... 164

7 Summary, Contributions, and Suggestions for Future Work .......... 165
  7.1 Contributions ......................................................................... 165
  7.2 Suggestions for future work ................................................... 167

A Notation .................................................................................. 169

B Derivations for smoothing conditions ........................................ 171
  B.1 Diffusion from energy ............................................................ 171
  B.2 Curvature from total variation ............................................... 172
  B.3 Curvature from contour length .............................................. 173

C Derivation of the GMM-EM ....................................................... 175
  C.1 Derivation of the GMM-EM update equations ......................... 175
    C.1.1 Expectation step ............................................................... 176
    C.1.2 Maximization step ............................................................ 176
  C.2 Derivation with Dirichlet and Normal-Inverse Wishart priors ..... 177
  C.3 Derivation of the three-stage GMM-EM .................................. 179
    C.3.1 Expectation step ............................................................... 180
    C.3.2 Maximization step ............................................................ 181
List of Figures

1-1 Image segmentation defined. ........................................... 25
1-2 Current state of the art in stem cell quality assessment. .......... 27
1-3 A vision for automated human embryonic stem cell (hESC) cultivation. 28
1-4 Stem cell colony segmentation. ........................................ 30

2-1 Demonstration of level set segmentation. .................................. 40
2-2 Two-stage finite mixture model. ......................................... 47
2-3 Smoothing prevents topological fracture in the segmentation of noisy images. ........................................ 51

3-1 A simple example demonstrating the rapid convergence of the multiclass BLS. ........................................ 72
3-2 BLS segmentation of iPSC colonies using automated initial conditions. 74
3-3 Comparison of smoothing priors for BLS segmentation. ............ 76
3-4 Retinal lesion segmentation. ........................................... 77
3-5 Comparison of BLS and FMM-EM for MRI phantom segmentation. .. 78
3-6 Noise analysis for BLS and FMM-EM for MRI phantom segmentation. 80
3-7 Massively multiclass BLS segmentation of fixed, early differentiating cells. ........................................ 81

4-1 Image texture as statistical fluctuation in image gray scale intensity. 87
4-2 The generalized Gaussian distribution. .................................. 93
4-3 Nuclei classification via wavelet texture modeling. .................... 97
4-4 Comparing GLCM and wavelets for cell nucleus classification. .... 98
4-5 Adaptive windowing for classification of small, irregularly shaped regions. 99
4-6 Window-based stem cell colony classification using wavelet texture models. 101
4-7 Impulse response of DWT and DT-CWT filters. 107
4-8 The filters for abarσ and hdiffσ. 109
4-9 Texture feature comparison. 114
5-1 Three stage Markov texture model 118
5-2 A zebra's stripes are an example of texture. 119
5-3 A demonstration of the Markov texture model. 120
5-4 In texture-based segmentation, the most likely texton varies within the class. 121
5-5 The mixture prior suppresses outlier classes to prevent misclassification. 125
5-6 Data priors prevent misclassification due to overtraining. 126
5-7 Pluripotent and differentiated training images. 131
5-8 Mutual information-guided feature selection. 133
5-9 The hdiff texture feature is critical for successful segmentation near colony boundaries. 133
5-10 dcmax is generally the most informative wavelet texture feature. 134
5-11 Segmentation precision limits neighborhood scale. 135
5-12 The predictive power of individual compressive sensing texture features is low. 136
5-13 PCA feature reduction. 137
5-14 Data reduced via PCA. 138
5-15 Pluripotent textons derived via the GMM-EM. 140
5-16 Zebra image segmentation using the texture-based BLS. 141
6-1 Pluripotent colony segmentation accuracy. 147
6-2 Differentiated colony segmentation accuracy. 148
6-3 Simultaneous segmentation and classification of the training set. 150
6-4 Simultaneous segmentation and classification of pluripotent stem cell colonies (D-L). .............................................. 151
6-5 Simultaneous segmentation and classification of pluripotent stem cell colonies (M-U). .............................................. 152
6-6 Simultaneous segmentation and classification of pluripotent stem cell colonies (V-BB). .............................................. 153
6-7 Simultaneous segmentation and classification of differentiated stem cell colonies (D-L). .............................................. 154
6-8 Simultaneous segmentation and classification of differentiated stem cell colonies (M-S). .............................................. 155
6-9 Evolution of a single BLS segmentation: pluripotent colony segmentation label map. .............................................. 156
6-10 Evolution of a single BLS segmentation: pluripotent colony geometric prior. .............................................. 157
6-11 Evolution of a single BLS segmentation: pluripotent textons in feature dimensions 1 and 2. .............................................. 158
6-12 Differentiated colonies inappropriate for segmentation via texture-based BLS (T-W). .............................................. 159
6-13 Retraining differentiated texton priors to recognize diffuse colonies. .............................................. 160
6-14 Segmentation of mixed-class stem cell colony phantoms. .............................................. 162
6-15 Screening for anomalous texture classes in post-processing. .............................................. 163
List of Tables

2.1 Comparative advantages of Level Sets and the FMM-EM. 53

3.1 Pseudocode for the Bayesian Level Set (BLS) algorithm. 62

3.2 Comparative advantages of Level Sets, the FMM-EM, and the BLS. 68

4.1 The history of image texture modeling. 89

5.1 Pseudocode for a texture-based BLS algorithm. 129

6.1 Numerical results for simultaneous segmentation and classification of stem cell colonies. 146
Chapter 1

Introduction

The primary objective of this thesis is the formulation of an algorithm for automated simultaneous image segmentation and classification. The first algorithm proposed is the general purpose Bayesian Level Set (BLS), which produces smooth, regular borders just as level sets do and yet possesses a simple, probabilistic implementation similar to that of finite mixture model Expectation-Maximization. The second algorithm harnesses the power of image texture modeling for segmentation by incorporating learned sets of class-specific textural primitives within the BLS via a three-stage Markov process. Early versions of the former and latter algorithms are seen in [63] and [64]. The texture-based BLS is then applied to successfully automate the segmentation and classification of a set of unknown pluripotent and differentiated stem cell colony images with accuracy 97% and 91%, respectively, effectively demonstrating that this approach serves as a non-invasive, non-destructive, automatable, statistical biomarker for stem cell classification and quality assessment that may serve as a surrogate to microscopy or chemical staining [72, 102]. These two algorithms are also validated on other applications, including brain MRI phantoms, stem cell nuclei, and wildlife images.
1.1 Need for automated segmentation and classification

*Image segmentation* is the process of automatically dividing an image into its constitutive disjoint regions, such as distinguishing the zebra in the foreground of the image in Figure 1-1 from the plain comprising the image background. As each region may be considered a separate class, image segmentation is essentially a labeling problem that assigns to each pixel or coordinate within an image a label based on the distinct region to which it belongs. Segmentation is thus a necessary prelude to many other image analysis applications, including biomedical ones. For instance, in image-based high-throughput screening, segmentation might automatically demarcate a cell colony from its growth medium prior to analyzing or grading colony quality according to criteria such as image texture or shape. Likewise, in brain Magnetic Resonance Imaging (MRI) classification, brain tissues are first segmented as a preliminary step before identifying or classifying tumors or diseased tissue. Non-medical applications might include biome identification in satellite imaging [24] or video motion tracking or surveillance, in which the same object must be located and accurately segmented across a time series of video frames. As the applications of segmentation are rich and varied, a wide variety of segmentation algorithms exist, including thresholding [84], region growing [44], Expectation-Maximization [59, 60], phase fields [52, 91], graph cuts [9, 10], level sets [18, 115], etc.

While these algorithms vary, most attempt to balance the image data and region geometry in determining a final segmentation. Image data quantifies the variation in local appearance between regions. In the absence of geometric criteria, however, purely data-driven approaches often result in the misclassification of pixels due to noise and outliers. Consideration of region geometry accounts for the observation that, within the human field of vision, the location of an object is grouped in one region of space and its perceptual interface with its background is generally smooth or rounded. Some approaches, including most Markov random field and graph cut algorithms, address this consideration by inducing pairwise dependence between the
classification of a pixel and its immediate neighbors. Level set algorithms attain even smoother results over larger neighborhoods by penalizing the curvature of the segmenting contour. While these techniques may be powerful, they are generally far more difficult to implement than algorithms such as Expectation-Maximization, which typically either forgoes geometric criteria or is limited to static prior atlases. A simpler and more computationally efficient approach to implementing geometric criteria in segmentation is needed.

A further complication is the tight connection between segmentation and another problem often performed separately in image processing literature: classification, recognizing the identified region and assigning it to a class based on a library of prior examples. Together, segmentation and classification constitute a chicken-and-egg type problem for many non-trivial images. If the classes are stipulated (e.g. pluripotent cells and growth media in a cell colony image), it would be possible to segment an image according to the characteristics indicative of those classes. Likewise, the statistics of regions in a segmented image would enable classification. Performing these tasks simultaneously, however, is often quite challenging.

This thesis builds upon and extends insights from prior approaches to formulate a segmentation and classification algorithm that fuses the powerful geometric

---

1 A segmentation atlas assigns a prior probability to each class at each pixel and is commonly used for problems in which the general shape of solution is known, such as brain MRI segmentation or object recognition.
criteria of level sets to a simple, probabilistic implementation similar to Expectation-Maximization. The resulting Bayesian Level Set (BLS) returns segmentations with smooth, regular borders, natively extends to multiple image regions, and optionally incorporates data priors in order to simultaneously segment and classify images.

1.2 Need for non-invasive biomarkers for stem cell monitoring

The motivating application for the development of the BLS is automating the segmentation and classification of stem cell colony images. While stem cells offer many exciting possibilities for therapeutic and research use, one of the key bottlenecks in developing this technology is the lack of a procedure for automated stem cell cultivation. Current state-of-the-art in stem cell monitoring requires either manual inspection by a trained microscopist, which is costly, time consuming, and necessarily subjective, or chemical staining, which is destructive and renders a colony unfit for further use. This alternatives are illustrated in Figure 1-2. Image processing offers the possibility of a non-invasive, non-destructive, statistical biomarker for stem cell quality assessment and monitoring, which may potentially serve as an enabling technology for automated stem cell cultivation as in Figure 1-3.

1.2.1 Importance of stem cells

Due to their pluripotency and longevity, pluripotent stem cells have been of intense interest to biomedical researchers in the fifteen years since the 1998 discovery of a practical procedure for the cultivation of human embryonic stem cells (hESC) by researchers under the direction of Dr. James Thomson at the University of Wisconsin-Madison [114]. Stem cell lines are capable of long-term self renewal and are effectively immortal when cultivated under laboratory conditions. More importantly, however, hESCs are pluripotent and have yet to specialize into any of the specific cell types that comprise the human body. By contrast, stem cells which have begun to specialize
Chemical staining is rapid, automatable, and precise, but destructive and renders the colony unfit for further research or therapeutic use; microscopy is non-invasive and non-destructive but non-quantitative and requires a trained microscopist to analyze results by eye.

More recently, the 2006 discovery by Takahashi and Yamanaka of a procedure to create induced pluripotent stem cells (iPSC) [113] by effectively “de-differentiating” adult cells into a pluripotent state offers the potential to mitigate the ethical and political concerns surrounding the manufacture of pluripotent cells from human embryos.

The pluripotency of hESCs and iPSCs offers exciting possibilities to biomedical researchers. In addition to basic research in human embryology and early development, potential therapeutic and research applications of stem cells include:

Tissue growth. The possibility for clinician-directed differentiation of stem cells into specific types might allow for cell-based therapies in which tissues are produced in the laboratory and then transplanted into a human body to replace damaged or destroyed cells without triggering an immune response. Potential applications include Parkinson’s disease [6], diabetes [67], and heart disease [129].

Drug discovery. Another possible near-term use of pluripotent stem cells is high-throughput screening in drug research and discovery. In such procedures, mass testing is used to discover potential druggable compounds and assess their potency and side effects or toxicity by treating an array of diseased and healthy cells and tissues with a wide variety of test compounds. Again, directed differentiation of stem cells might
allow for precise mass production of the targeted cells and tissues [101].

1.2.2 Current practices

At present, there are an estimated twelve thousand stem cell colonies under laboratory cultivation; therapeutic application of stem cell technology mandates increasing this quantity by at least an order of magnitude. A chief impediment to doing so is the current lack of an automatable, non-invasive, and non-destructive means of monitoring, testing, and cultivating cell colonies as illustrated in Figure 1-3. Stem cells are comparatively fragile, and differentiated cells tend to induce nearby pluripotent cells to differentiate. Therefore, each hESC or iPSC colony must be inspected every two days in order to remove damaged cells and pockets of spontaneous differentiation that might lead to a chain reaction causing the entire colony to differentiate [37, 85].

Unfortunately, neither of the two current state-of-the-art methods for stem cell inspection is suitable to mass cultivation. While biochemical and immunochemical staining tests exist that are rapid, consistent, and automatable, results are non-quantitative and the applied chemical factors are destructive, killing the cells and
rendering them unsuitable for further research or clinical use. In practice, therefore, researchers rely on visual inspection of stem colonies via brightfield microscopy, interpreting the state of the colony from its morphological properties. While this procedure is non-invasive and allows for further use of the colony, it is necessarily subjective and non-quantitative. More important, however, are the time and expense associated with training microscopists and having them manually inspect cell colonies. Simply put, there are not enough of these skilled individuals for them to check each of the many thousands of colonies necessary for widespread cultivation, and if there were, their time could be put to better use.

1.2.3 Image analysis as a non-invasive, non-destructive stem cell biomarker

Image analysis presents a potential solution, if segmentation and classification methods were developed to automate the inspection tasks now performed by microscopists. Automated mass screening of cell colonies might be enabled by algorithms sufficiently accurate to identify the type and approximate location and size of cell colonies and flag questionable or differentiating colonies for further analysis. The hardware infrastructure for large scale, automated image collection of cell cultures already exists in many laboratories. Computer-based image processing is therefore a potentially enabling technology for automated non-invasive, non-destructive monitoring of stem cell colony cultivation in a manner that is scalable and exceeds the rate of human-possible classification.

1.3 Contributions

The key contributions of this thesis fall into two major categories:

1. *Image segmentation and texture modeling*. This thesis introduces a Bayesian Level Set (BLS) algorithm for image segmentation, which combines the advantages of two existing segmentation algorithms: level sets and the finite mixture

2. Non-invasive, non-destructive image-based stem cell biomarkers. This thesis demonstrates that image texture constitutes a non-invasive, non-destructive biomarker for segmenting and classifying stem cell images, which may enable future automated stem cell monitoring and cultivation.

1.3.1 Contributions to image segmentation and texture modeling

Major contributions

1. A new Bayesian Level Set (BLS) algorithm for image segmentation (chapter 3).

The key intuition underlying this iterative approach to segmentation is that a geometric smoothing prior computed from the \textit{a posteriori} classification may be used to encourage local similarity in labeling, with the result that the BLS
produces the smooth, regular segmenting contours typical of level sets using a Bayesian approach akin to that employed by Finite Mixture Model Expectation-Maximization (FMM-EM).

Compared to the FMM-EM, the geometric smoothing prior of the BLS:

- noticeably reduces misclassification due to outliers, and
- returns far more natural segmenting contours.

Compared to level sets, the BLS:

- is simpler in implementation, requiring neither the solution of partial differential equations nor ancillary algorithms,
- natively extends to an arbitrary number of classes without complicated mathematical machinery, and
- requires far more lenient initial conditions, demonstrating a capture radius which exceeds that typical of level sets.

Earlier versions of this algorithm were demonstrated in [63, 64].

2. Simultaneous segmentation and classification via a texture-based BLS (chapter 5). By incorporating image texture modeling into the BLS framework via a three-stage Markov texture model, texture may be used to automatically identify the types of regions within an image that may be then segmented via the BLS. This algorithm incorporates all the other advantages of the BLS and is used to automate the simultaneous segmentation and classification of a set of pluripotent and differentiated stem cell images as described in section 1.3.2.

Minor contributions

Traditional image texture models are of little use for segmentation as they depend on statistics calculated over rectangular, texturally homogeneous windows. For segmentation, texture features must be calculated at each coordinate, rather than over a fixed window, and must necessarily be modeled near the interface between the
textures to be segmented, precluding homogeneity. Extension of the BLS algorithm to texture-based segmentation generated two minor contributions to image texture modeling:

1. A framework for localizing image texture models for segmentation (section 4.3.1). Texture feature dimensionality is reduced by retaining only the maximal response over orientation at a particular scale. Localized statistics are then computed by utilizing Gaussian smoothing to compute a weighted local average.

2. An adaptive windowing method for texture-based classification of small, irregularly shaped image regions (section 4.2.2). Applications include cell nuclei [65].

1.3.2 Contributions to non-invasive, quantitative, image-based stem cell biomarkers

By automating the simultaneous segmentation and classification of a series of pluripotent and differentiated stem cell colony images, it is demonstrated that image texture serves as a non-invasive, non-destructive biomarker for assessing stem cell pluripotency status. This quantitative, statistical biomarker serves as a surrogate or potential replacement for destructive chemical staining that renders a colony unsuitable for further use in cultivation, therapy, or research. By eliminating the need for direct human observation of the majority of stem cell colonies, image texture-based screening offers the potential for quantitatively monitoring stem cell quality at a greatly increased spatial and temporal resolution relative to current practices in microscopy or chemical staining. This contribution may eventually enable automated cultivation of stem cells and their derivatives.

1.4 Organization

The organization of this thesis is as follows.

Chapter 2 describes the history of and introduces the mathematical background underlying level sets and Finite Mixture Model Expectation-Maximization (FMM-
EM), the two prior segmentation algorithms most influential to the development of Bayesian level sets. After describing these algorithms and deriving their update equations from corresponding utility functions, the relative advantages and difficulties of the two approaches are assessed.

Chapter 3 formulates the basic Bayesian Level Set (BLS) algorithm. The BLS is first formulated by demonstrating that the level set minimizing conditions may be reached via a Bayesian implementation similar to the FMM-EM by imposing a prior derived from curvature-based smoothing of the \textit{a posteriori} classification. Subsequently, the similarity of the BLS to the EM is explored and various means are noted of extending and modifying the prior via geometric transformations of the \textit{a posteriori} classification. The advantages of the BLS with respect to convergence rate, multi-region segmentation, and ease of initialization are then demonstrated on a variety of problems including basic segmentation of iPSC colonies, retinal lesions, MRI brain phantom images, and stem cell nuclei.

Chapter 4 introduces image texture and demonstrates its suitability for classifying stem cell colony images. This chapter first summarizes the history of image texture methods, distinguishing between approaches which define textural neighborhoods in the spatial or frequency domains. It next shows that texturally homogeneous stem cell images may be classified very accurately using wavelet texture methods, a species of frequency-domain approach. Finally, it introduces a framework for localizing image texture for use in segmentation and describes a variety of suitable texture models.

Chapter 5 incorporates the texture methods described in chapter 4 within the BLS framework introduced in chapter 3 via a three-stage Markov texture model. After imposing data priors in order to allow for simultaneous segmentation and classification, the chapter discusses the training of the algorithm via feature selection, reduction, and clustering with reference to the classification of stem cell colonies.

Chapter 6 demonstrates the application of this algorithm to the simultaneous segmentation and classification of stem cell colonies. Pluripotent results are near perfect at 97% accuracy, and differentiated results are accurate to within 91%.
Chapter 7 concludes with a list of contributions and suggestions for future work.
Chapter 2

Level Sets and Expectation-Maximization: A Comparative Analysis

Consider some image consisting of one or more disjoint foreground regions within a background region. Image segmentation is the process whereby computer vision algorithms are used to distinguish these regions. As each region may be considered a separate class, segmentation is thus a type of classification or labeling problem within the realm of machine learning; the algorithm must return the sort of information that would enable the user to assign a class label to each pixel or voxel in the image. Segmentation algorithms may therefore be usefully divided according to the type of information returned. **Hard classifiers** assign a class label directly to each coordinate, while **soft classifiers** report the probability that each coordinate belongs to each class. It is also useful to distinguish segmentation from the edge detection problem solved by algorithms such as [14, 106]. Segmentation directly classifies image coordinates whereas edge detection identifies coordinates located at sharp transitions in image properties. As object boundaries are not uniformly sharp, edge fields may not form connected contours and may contain artifacts such as extraneous lines, so that labeling based on the edge field is often a non-trivial task.

Image segmentation is one of the more extensively researched fields in computer
vision, with a wide variety of algorithms developed for specific purposes and objectives. Since neither time nor space permit an exhaustive review of the entire field which would do justice to such diverse methods as thresholding [84], graph cuts [9, 10], Markov random field methods [31, 41], region growing [44], or watershed techniques [124], this review will focus instead on the two methods most germane to this thesis: Level Sets and Finite Mixture Model Expectation-Maximization (FMM-EM).

2.1 Notation and conventions

Before reviewing segmentation or formulating a novel algorithm for this purpose, it is helpful to first formulate a standardized set of notation and conventions.

Data and coordinates

The coordinate system of an image is immediately imposed by the location of pixels or voxels in the image. Any particular coordinate $x$ thus exists in domain $\Omega$.

The hidden data is the true classification $Z$, which assigns to each pixel $x \in \Omega$ a class or label $c$ from set $C$. All classes in $C$ are disjoint, so that $Z(x) = c$ implies that $Z(x) \neq d$ for $d \in C - c$.

Observations $Y$ consist of the observed data for each pixel $x \in \Omega$, whether image intensity, some function thereof, or a vector derived from various imaging modalities. It is assumed that $Y(x)$ is generated stochastically from $Z(x)$ according to some set of class-specific distribution parameters $\theta_Z$.

The set of all parameters is $\Theta$, which includes all distribution parameters $\theta_Z$ and any additional parameters noted in the text.

Probabilities

For ease of expression and compactness in notation, probabilities will be abbreviated via the following $q$-notation, so that $P^k(Z(x) \mid Y(x); \theta_Z)$, the probability density at $x$ may refer to a voxel or coordinate in a higher dimensional space. While the term pixel will be used for the sake of concreteness, the three principal algorithms described in this thesis (level sets, the FMM-EM, and the BLS) may also be used for higher-dimensional segmentation.
iteration $k$ that pixel $x$ takes on class $Z(x)$ when conditioned on data $Y(x)$ subject to parameters $\theta_Z$ may be denoted compactly via subscripts:

$$q^k_{Z|Y}(x) = P^k(Z(x) | Y(x); \theta_Z)$$

(2.1)

In general, dependence on parameters $\theta_Z$ will be left implicit when doing so is judged unlikely to result in confusion. In the event that it is necessary to refer to the probability of assignment to a single class $c \in \mathcal{C}$, this will be substituted as:

$$q^k_{c|Y}(x) = P^k(Z(x) = c | Y(x); \theta_c)$$

(2.2)

Likewise, $P^k(Z(x))$, the prior probability of class $Z$, and $P^k(Y(x) | Z(x); \theta_Z)$, the probability density of the data $Y$ at pixel $x$ conditioned on class $Z$ and parameters $\theta_Z$ are:

$$q^k_Z(x) = P^k(Z(x))$$

(2.3)

$$q^k_{Y|Z}(x) = P^k(Y(x) | Z(x); \theta_Z)$$

(2.4)

**Signal to Noise and Interference Ratio (SNIR)**

The Signal to Noise and Interference Ratio (SNIR) in decibels (dB) will be computed as:

$$10 \log_{10} \frac{s}{\sigma}$$

(2.5)

where $s$ is the signal intensity, which is equal to the minimum difference in mean intensity between image regions, and $\sigma$ is the noise standard deviation. Thus, an image consisting of two regions separated by a unit change in intensity ($s = 1$) corrupted by noise with standard deviation of $\sigma = 0.5$ has SNIR of 3.0103 dB.
2.2 Level sets

2.2.1 History of level sets

Origins in computational physics

Level set methods were initially developed by Osher and Sethian [83] for use in computational physics as an aid to modeling front propagation in applications such as flame expansion, fluid shockwaves, crystal growth, etc. Prior to the advent of level sets, such problems had been solved via the so-called “Lagrangian” approach in which marker particles along the front’s location were evolved according to the curve evolution equation and the front’s own velocity field, with the front itself reconstructed in post-processing using spline functions. While this reconstruction is subject to interpolation error, the method is also acutely vulnerable to topological problems; disconnected particles create ambiguities during region merging and splitting which could only be addressed using a variety of ad hoc techniques.

The key insight behind level set methods was to dispense with marker particles by embedding the evolving front into the zero level set\(^2\) (hence the name) of an auxiliary surface \(\phi\) represented at lattice points over the entire region of interest. If the surface \(\phi\) is then evolved so that its zero level set obeys the curve evolution equation for the evolving front, the implicit location of the front can be uniquely and accurately determined at each time step while natively accounting for topological changes. This so-called “Eulerian” approach ushered in a minor revolution in the modeling of curve evolution. Algorithmic research stimulated by this approach includes application-specific finite-differencing methods to improve accuracy [19, 111], narrow-band solvers, which speed convergence and reduce computational burden by evolving \(\phi\) only along lattice points near the advancing front [1, 86], and the modeling of multiphase motion by using multiple level set functions coupled via constraint equations [131]. The publishing of standard reference works [82, 105] approximately a decade ago confirmed the maturity of the field.

\(^2\)More formally, the zero level set of \(\phi\) is \(\{x \mid \phi(x) = 0\}\).
Extension to image processing

It was only natural that level sets should be harnessed for image segmentation. Based on the 1988 work of Kass, Witkin, and Terzopolous [51], active contours or “Snakes” were segmenting images by evolving particles whose location marked the location of the segmenting contour in a manner analogous to the Lagrangian approach in computational physics, many of which utilized curvature-based terms for regularization similar to the ones found in natural front propagation processes.

When applied to image segmentation, the zero level set of $\phi$ becomes the segmenting contour, and the sign of $\phi$ (positive or negative) picks out the two regions in the image as illustrated in Figure 2-1. The level set function $\phi$ is shown in (d)-(k). The zero-level set of this function, marked in red, corresponds to the yellow segmenting contours in (a)-(c). Likewise, regions of negative $\phi$ indicate the darker foreground objects, whereas $\phi$ is positive in the lighter background regions. That is, at pixel $x$:

$$\phi(x) > 0 \Rightarrow x \in \text{Region 0}$$
$$\phi(x) < 0 \Rightarrow x \in \text{Region 1}$$

Level set methods are thus hard classifiers; pixels are labeled definitively as belonging to one region or another rather than being assigned a probability of class inclusion. Note that $\phi$ in Figure 2-1 is a signed distance function, meaning that $|\phi(x)| = d$, where $d$ is the distance from pixel $x$ to the zero level set of $\phi$, which implies that $|\nabla \phi| = 1$ everywhere save at points or ridges equidistant to two points along the contour. As smoothness in $\phi$ allows for accurate location of the segmenting contour and unitary gradient magnitude facilitates accurate calculation of the curvature term (see below), the signed-distance function is the standard choice for $\phi$ in level set algorithms.

Early level set-based image segmentation adapted the framework of Osher and Sethian to solve problems previously posed to active contours. In these edge-based methods, inflationary and attractive terms were used to drive the segmenting contour to locations identified by edge detectors in pre-processing [16]. The second key breakthrough came with the near-simultaneous work of Chan and Vese [18] and Tsai
Figure 2-1: Demonstration of level set segmentation. In (a)-(c), the yellow contour indicates the zero level set of \( \phi \), which evolves so that it separates the dark foreground regions from the light background. In (d)-(f) and (g)-(k), the corresponding level set function \( \phi \) is shown in profile; red marks the zero level set. Figure from J. Kim [53].

[115] who developed a methodology for using level sets to segment images based on statistics of the identified regions.

### 2.2.2 Region-based level sets

The 2001 paper of Chan and Vese [18] was seminal in introducing region-based active contours to the image processing community, and their formulation remains characteristic of the entire field. For the pedagogical benefits of concreteness, their result will first be derived before interpreting it and discussing the implications of this method.

**Derivation of the Chan-Vese algorithm**

Begin with the famous Mumford-Shah functional for simultaneous segmentation and smoothing [76, 77]:

\[
\min_{f, \Gamma} E_{MS}(f, \Gamma)
\]  

(2.6)
where
\[ E_{MS}(f, \Gamma) = \int_{\Omega} (Y(x) - f(x))^2 \, dx + \alpha \int_{\Omega \setminus \Gamma} \|\nabla f(x)\|^2 \, dx + \beta |\Gamma| \] (2.7)

\( f \) is a smoothed version of the input image \( Y \), and \( \Gamma \) is a set of curves dividing the image into various regions. The first term in the functional enforces data fidelity, penalizing deviation of \( f \) from \( Y \), while the second, smoothing term penalizes the gradient of \( f \) except on the segmenting curve \( \Gamma \), where \( f \) is allowed to vary discontinuously to represent the transition from one image region to another. The final term penalizes the length of the segmenting curve. Weight parameters \( \alpha \) and \( \beta \) modulate the interaction of these three terms. Note that, in the absence of \( \Gamma \), (2.7) reduces to a Wiener or Kalman smoother.

Perhaps the most common application of the Mumford-Shah functional is that of Ambrosio and Tortorelli [3]. By implementing the contour \( \Gamma \) as an edge detection problem in order to penalize smoothing near image discontinuities, this approach enables the restoration and enhancement of noisy images with powerful results for medical imaging [32, 34].

The Chan-Vese model, however, instead adapts the Mumford-Shah functional for image segmentation by parameterizing the segmenting curve \( \Gamma \) according to the zero-level set of an auxiliary function \( \phi \). By further assuming an image with two regions of constant intensity \( \mu_0 \) and \( \mu_1 \), the Mumford-Shah functional becomes [18]:

\[ \min_{\phi, \mu_0, \mu_1} E_{CV}(\phi, \mu_0, \mu_1) \] (2.8)

where
\[ E_{CV}(\phi, \mu_0, \mu_1) = \int_\Omega H(\phi(x)) (Y(x) - \mu_0)^2 \, dx \] (2.9)
\[ + \int_\Omega (1 - H(\phi(x))) (Y(x) - \mu_1)^2 \, dx \]
\[ + \beta \int_\Omega \|\nabla H(\phi(x))\| \, dx + \gamma \int_\Omega H(\phi(x)) \, dx \]

41
\( H(\cdot) \) is the Heaviside step function:

\[
H(\phi(x)) = \begin{cases} 
1 & \phi(x) \geq 0 \quad (x \in \text{Region 0}) \\
0 & \phi(x) < 0 \quad (x \in \text{Region 1}) 
\end{cases} 
\]  
(2.10)

Under the piecewise-constant model, the first (data fidelity) term in (2.7) corresponds to the first two terms in (2.9), the second (smoothing) term of (2.7) is precisely zero, and the third (contour length) term in (2.7) is equivalent to the third term in (2.9), which picks out the length of the zero-level set of \( \phi \). The fourth term in (2.9) is new and was added to penalize the existence of region 0 where \( \phi \geq 0 \) according to the user-specified parameter \( \gamma \).

To find minimizing conditions, take the variation with respect to \( \phi \), understanding that the variation of the Heaviside step function \( H(\cdot) \) is the Dirac delta function \( \delta(\cdot) \):

\[
0 = \delta(\phi(x)) \left[ (Y(x) - \mu_0)^2 - (Y(x) - \mu_1)^2 - \beta \kappa(\phi(x)) + \gamma \right] \quad \text{on } x \in \Omega 
\]  
(2.11)

\[
= \frac{\delta(\phi(x)) \partial \phi(x)}{\|\nabla \phi(x)\| \partial n(x)} \quad \text{on } x \in \partial \Omega 
\]  
(2.12)

where \( \partial \Omega \) is the boundary of \( \Omega \) and \( n \) is its normal. \( \kappa(\phi) \) is the curvature of the iso-contours of \( \phi \) and is equivalent to the divergence of its normal:

\[
\kappa(\phi(x)) = \nabla \cdot \left( \frac{\nabla \phi(x)}{\|\nabla \phi(x)\|} \right) 
\]  
(2.13)

The first, second, and fourth terms in (2.11) follow immediately from (2.9), while the derivation of the third term and the Von Neumann conditions are given in appendix B.3. Arising as it does from a penalty on the length of the segmenting contour,\(^3\) this curvature term leads to minimizing solutions with smooth, regular borders between regions. This is perhaps the key advantage of level sets as opposed to competing methodologies [112].

By introducing an artificial time parameter \( t \), (2.11) may be solved via continuous-\(^3\)
\(^{\text{i.e. the set } \{ x \mid \phi(x) = 0 \}.} \)
time gradient descent using the following partial differential equation [18]:

$$\frac{\partial}{\partial t} \phi(x) = \delta(\phi(x)) \left[ - (Y(x) - \mu_0)^2 + (Y(x) - \mu_1)^2 + \beta \kappa(\phi(x)) - \gamma \right]$$  \hspace{1cm} (2.14)

An example of Chan-Vese segmentation is shown in figure 2-1.

**Region-based level sets**

The Chan-Vese algorithm derived above is a specific example of an algorithm family known as *region-based level sets*, which pose the segmentation data fidelity term according to a log-probability. In the case of Chan-Vese, this is:

$$\log P(Y(x) \mid Z(x) = c; \theta_c) \propto - (Y(x) - \mu_c)^2 \quad c \in [0, 1] \quad (2.15)$$

$\theta_c$ are the parameters governing the distribution of the data in class $c$. In the case of Chan-Vese, $\theta_c = \mu_c$. The generalized functional is:

$$\min_{\phi, \Theta} E_{RC}(\phi, \Theta) \quad \text{(2.16)}$$

where

$$E_{RC}(\phi, \Theta) = - \int_{\Omega} H(\phi(x)) \log P(Y(x) \mid Z(x) = 0; \theta_0) \, dx$$

$$- \int_{\Omega} (1 - H(\phi(x))) \log P(Y(x) \mid Z(x) = 1; \theta_1) \, dx$$

$$+ \beta \int_{\Omega} \|\nabla H(\phi(x))\| \, dx + \gamma \int_{\Omega} H(\phi(x)) \, dx \quad \text{(2.17)}$$

which has the update equation:

$$\frac{\partial}{\partial t} \phi(x) = \delta(\phi(x)) \left[ \log \frac{P(Y(x) \mid Z(x) = 0; \theta_0)}{P(Y(x) \mid Z(x) = 1; \theta_1)} + \beta \kappa(\phi(x)) - \gamma \right] \quad \text{(2.18)}$$

As $\phi$ (and hence the classification) is evolved according to the log-likelihood of the data conditioned on the class, this is simply a level set implementation of Zhu-Yuille region-competition [132]. This is a powerful framework, because it allows for level
set-based segmentation under most any model in which observed data can be assigned a probability conditioned on its being in a given region, and the functional $E_{RC}$ was quickly given a probabilistic interpretation:

$$E_{RC} = P(Y | Z) P(Z)$$

(2.19)

The first two terms in (2.17) correspond to the probability $P(Y | Z)$ of the data given the class, while the third and fourth terms notionally correspond to some prior probability $P(Z)$ on the classification.

**Determining region parameters $\theta$**

Heretofore, there has been no discussion concerning the origin of the parameters $\Theta$ used to model the data distributions in (2.17). Likewise, in deriving the Chan-Vese algorithm, it was simply assumed that the mean region intensities $\mu_0$ and $\mu_1$ were known. In practice, however, these parameters are seldom known a priori, and so the parameters of region-based level set algorithms must be continually re-estimated between updates to $\phi$ based on the current segmentation. While alternating between updates to the segmentation $\phi$ and the parameters $\theta$ is sometimes justified by appealing to “the EM principle,” level sets are not EM algorithms and do not share their guaranteed convergence. Indeed, level sets often display limit cycle behavior due to the interplay between parameter updates based on a hard classifier. Specifically, one set of parameter estimates causes a particular group of pixels to be re-assigned; updated parameters based on that re-assignment then cause that same group of pixels to be re-re-assigned back to their original class, and the cycle continues.

**2.2.3 Further development**

After more than a decade of use, region-based level methods are a standard component of the image processing toolkit owing to their topological flexibility, the smooth, regular segmentation borders they return, and the extensive body of literature that has expanded the range of utilizable data models $P(Y | Z)$. A complete summary
of the field's extensive literature would require a monograph in its own right, but a few highlights may serve to illustrate the richness of this field. Beginning with the original, piecewise constant model of Chan and Vese [18] and the mean separation energy of Tsai [115] which maximized \((\mu_0 - \mu_1)^2\), a great deal of subsequent work expanded the range of distributions admissible to the data, including the family of exponential distributions [57], segmenting or model matching based on divergence (i.e. Bhattacharyya and Kullback-Leibler) flows [39, 75], and information-theoretic methods minimizing entropy [46] or maximizing the mutual information [54] of empirical image distributions based on Parzen estimators. Other work localized model parameter estimation in order to segment perceptible objects with spatially-varying statistics, such as a shadowed object whose intensity varies from one side of an image to another [27, 56], and allowed for shape-based object recognition via priors modeled using level set functions based on reference shapes [40, 99, 116]. Of particular interest to the present research are texture-based extensions [11, 25], including wavelet models [4, 58] and time-dynamic textures modeled via Kalman filtering [36], and methods modeling vector data via a sum of log-likelihoods in (2.18), effectively imposing a Naïve Bayesian [73] model on the data.4

2.3 Expectation Maximization (EM) of finite mixture models

This section provides a brief account of the origin and theoretical underpinnings of the Expectation-Maximization (EM) family of algorithms, introduces Finite Mixture Models (FMM) and provides their EM update equations, and describes some of the more significant uses of the EM in biomedical image segmentation.

4In practice, this approach is somewhat problematic. Under the Naïve Bayesian framework, as the number of data terms increase they begin to dominate over the smoothing term.
2.3.1 Development and theory

Expectation Maximization, or EM, is an approach to solving an entire family of problems first formalized and given its name in the seminal 1977 paper by Dempster, Laird, and Rubin [30]. Begin with a probability density function \( P(Y, Z \mid \Theta) \) which is dependent on observations \( Y \), latent or unknown variables \( Z \), and unknown parameters \( \Theta \). Were there no \( Z \), it would be possible to recover an approximate \( \Theta \) using maximum likelihood estimate (MLE) methods. Instead, the EM algorithm solves this joint estimation (\( \Theta \)) and identification (\( Z \)) problem using the following two-stage iterative process:

1. **Expectation** or E-step. At iteration \( k \), first calculate the expectation of \( P(Y, Z \mid \Theta) \) with respect to \( Z \), conditioned on \( \Theta^{k-1} \), the parameter estimate at the end of the previous iteration.

\[
U(\Theta \mid \Theta^{k-1}) = E_{Z \mid Y, \Theta^{k-1}} [P(Y, Z \mid \Theta)]
\] (2.20)

2. **Maximization** or M-step. Now, set \( \Theta^k \) to the value which maximizes \( U(\Theta \mid \Theta^{k-1}) \).

\[
\Theta^k = \arg \max_{\Theta} U(\Theta \mid \Theta^{k-1})
\] (2.21)

Subject to the corrections given in [128], this procedure can be shown to be non-decreasing in the sense that \( U(\Theta^{k+1} \mid \Theta^k) \geq U(\Theta^k \mid \Theta^{k-1}) \). As the expected likelihood \( U(\Theta \mid \Theta^{k-1}) \) may be non-convex, the EM is not guaranteed to converge to a global minimum.

Furthermore, [78] provides an information-theoretic interpretation for this procedure. Let \( Q \) be any proposed distribution for \( Z \). The EM algorithm is equivalent to minimizing the Kullback-Leibler divergence between \( Q \) and the true probability density function \( P(Y, Z \mid \Theta) \):

\[
\min_{Q, \Theta} D(Q \mid \mid P(Y, Z \mid \Theta)) = \max_{Q, \Theta} E_Q [\log P(Y, Z \mid \Theta)] + H(Q)
\] (2.22)
Figure 2-2: The two-stage process that produces the observations in the Finite Mixture Model or FMM. The first event in the process selects an unknown class $Z(x)$. The second event produces an observation $Y(x)$ from a probability density function whose parameters are dependent upon $Z(x)$.

where $H(Q)$ is the entropy of $Q$. EM is equivalent to solving this as an alternating maximization-maximization problem, solving first for the optimal $Q$ with fixed $\Theta$ and thence for the optimal $\Theta$ with fixed $Q$.

A great deal of subsequent literature has been written on the EM algorithm, analyzing its convergence rate and introducing various extensions. See [74] for a review.

2.3.2 Finite mixture model EM

The Finite Mixture Model EM (FMM-EM) [74] is one of the most common variants of the EM and enjoys extensive use in image processing and segmentation.

The Finite mixture model

Assume some set $\Omega$ of observations $Y$, each of which is independent of the others and produced by a two-stage process as illustrated in Figure 2-2. At index $x \in \Omega$, the first event assigns to the unknown class $Z(x)$ a label from some finite set $C$ according to a multinomial distribution with unknown parameters $\{\pi_c\}_{c \in C}$. Thus:

$$P(Z(x) = c) = \pi_c \quad \forall c \in C \tag{2.23}$$

To each $c \in C$ is associated an unknown set of parameters $\theta_c$ that characterize some known probability density function. In the second stage, observation $Y(x)$ is generated according to this probability density function with the parameters $\theta_c$ corre-
sponding to the class chosen in $Z(x)$:

$$P(Y(x) \mid Z(x) = c) = P(Y(x) ; \theta_c)$$ (2.24)

Since the set $\Omega$ is a mixture of observations $Y$ from a finite set of classes $\mathcal{C}$, this is known as a finite mixture model (FMM).

**Solution via EM**

In terms of the EM algorithm described in the previous section, observations $Y(x)$ are known for $x \in \Omega$, unknown latent variables are class assignments $Z(x)$, and unknown parameters include both the $\pi_c$ characterizing class assignment and $\theta_c$ characterizing observations for $c \in \mathcal{C}$. Denote the entire set of parameters $\Theta = \{\pi_c, \theta_c\}_{c \in \mathcal{C}}$.

The complete log likelihood of the FMM, which assumes known $Z$, is:

$$L(Y, \Theta) = \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} \delta(Z(x) = c) \log \pi_c P(Y(x) ; \theta_c)$$ (2.25)

where $\delta(Z(x) = c)$ is an indicator function which triggers on the class assignment:

$$\delta(Z(x) = c) = \begin{cases} 1 & Z(x) = c \\ 0 & \text{otherwise} \end{cases}$$ (2.26)

Now, assume some candidate probability density function $Q$ conditioned on $Y$ and $\Theta$ which assigns $P(Z(x) = c) = q_c(x)$; it follows that $E_Q[\delta(Z(x) = c)] = q_c(x)$. Thus, under distribution $Q$, the expected log-likelihood is:

$$E_Q[L(Y, \Theta)] = \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} E_Q[\delta(Z(x) = c)] \log \pi_c P(Y(x) ; \theta_c)$$ (2.27)

$$= \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} q_c(x) \log \pi_c P(Y(x) ; \theta_c)$$ (2.28)

Derivation of the EM update equations is given in appendix C.1. At iteration $k$, the
E-updates are:

\[ q_{c}^{k+1}(x) = \frac{\pi_{c}^{k}P(Y(x); \theta_{c}^{k})}{\sum_{c \in C} \pi_{c}^{k}P(Y(x); \theta_{c}^{k})} \] (2.29)

The M-updates for \( \pi_{c} \) are:

\[ \pi_{c}^{k} = \frac{1}{|\Omega|} \sum_{x \in \Omega} q_{c}^{k}(x) \] (2.30)

M-updates for \( \theta_{c} \) are dependent on choice of distribution, but for Normal data:

\[ \mu_{c}^{k} = \frac{\sum_{x \in \Omega} q_{c}^{k}(x)Y(x)}{\sum_{x \in \Omega} q_{c}^{k}(x)} \] (2.31)

\[ J_{c}^{k} = \frac{\sum_{x \in \Omega} q_{c}^{k}(x)(Y(x) - \mu_{c}^{k})(Y(x) - \mu_{c}^{k})^{T}}{\sum_{x \in \Omega} q_{c}^{k}(x)} \] (2.32)

where \( \mu_{c} \) and \( J_{c} \) are the mean and covariance matrix, respectively. A Finite Mixture Model in which \( P(Y | Z) \) is Normal is known as a Gaussian Mixture Model or GMM.

### 2.3.3 Use in image segmentation

The EM algorithm has seen extensive use in image segmentation, particularly in classifying brain MRI images into their constituent white matter, gray matter, and cerebrospinal fluid. This began with the 1996 paper by Wells et al. [125], which assumed a Normal model for the three brain regions with parameters estimated \textit{a priori} and then refined the segmentation by utilizing an EM algorithm to estimate the bias field which formed the residual between the ideal image (based on the segmentation and Normal model with known parameters) and the measured image. This approach was greatly refined by the 1999 companion papers of Van Leemput et al. [59, 60], which modeled the data directly as a GMM with unknown parameters. This work used the EM approach to simultaneously solve for both the unknown model parameters and the weights to a set of basis functions used to model the bias field. In lieu of global mixture parameters, spatially varying mixture parameters were introduced by setting \( \pi \) at each voxel to the values contained in an \textit{a priori} probabilistic atlas. More recent research has incorporated anatomical models into an hierarchical EM classification algorithm in order to precisely identify local brain structures [92].

49
Unlike level sets, however, these approaches do not impose explicit localized dependency between the classification of neighboring voxels. While bias field compensation removes low frequency noise, it cannot address outlier voxels with anomalous intensities due to high frequency and independent noise. The result is that EM MRI segmentation tends to return "island"-type errors in which these outlier voxels are erroneously assigned to a topographically isolated class [112]. Efforts have been made [130] to mitigate this by applying an Ising model [41] as a spatially localizing prior on classification, though the implementation is quite intricate despite imposing no more than pairwise local dependency in classification.

2.4 Stochastic level set-like algorithms

In recent years, algorithms have appeared that emulate the desirable features of level set algorithms – primarily topological flexibility and the smoothing of contour boundaries – within a more rigorously stochastic framework. As noted in (2.19), level sets were soon given a probabilistic interpretation, and information theoretic techniques were introduced in works such as [53], which cast the level set problem as a type of mutual information problem penalized by the length of the segmenting contour, and the so-called phase-field methods [52, 91], which determine a soft classification of the image pixels via methods created to solve the Allen-Cahn equation in computational physics. Another approach is Conditional Random Fields (CRFs) [22, 48, 117], in which a Markov random field is defined in terms of both the pixel classes and an auxiliary surface conditioned on the data. By linking class probability to an auxiliary surface whose likelihood increases with its smoothness, CRFs encourage smoothness in the segmenting contour in a manner similar to level sets.

Most interesting, perhaps, is a stochastic approach based on the logarithm-of-odds formulation which defines the probability of a class in terms of the logistic function of level set \( \phi \) [93]:

\[
P(Z(x) = 0 \mid Y(x); \theta_0) = \frac{1}{1 + e^{-\phi(x)}}
\]  

(2.33)

Using this relation, the varying forms of the level set update equation (2.18) may
be cast as a gradient-descent solver for a likelihood function formulated in terms of mean-field theory [94] or analogously to the EM algorithm [97].

The novel Bayesian Level Set algorithm developed in chapter 3 admits interpretation under this log-odds approach (see section 3.1.2) but eschews gradient descent solvers in favor of a Bayesian implementation.

## 2.5 Conclusions and commentary

While both level set and FMM-EM algorithms enjoy extensive use, the respective advantages of the two algorithms differ noticeably. Both algorithms naturally enjoy topological flexibility, the ability to gracefully split and merge image regions; level sets do this via the principled application of the “Eulerian” framework, while FMM-EM does so trivially by completely dissociating pixel classification from its spatial location. Beyond this, however, the advantages of the two algorithms are largely complementary, as illustrated in Table 2.1.

*Smooth, regular class transitions.* The chief advantage of level sets over the FMM-EM is its curvature-based smoothing term, which regularizes class borders and transitions. Smoothing effectively imposes localized stochastic dependence on classification in accordance with the intuition that neighbors of a given class should influence a pixel
to take on the same class. In the absence of such a smoothing term, the FMM-EM is prone to misclassification and topological fracturing, forming many small, spurious regions, especially in the absence of noise. See Figure 2-3 for an example.

**Soft vs. hard classifiers.** As a soft classifier, the FMM-EM returns the probability of class inclusion for each coordinate and can be shown to have guaranteed convergence to a local minimum. Level sets, by contrast, are hard classifiers and so return less information – class labels rather than class probabilities – and are prone to limit cycle behavior as they approach convergence.

**Multiclass capability.** While nothing within the FMM-EM E-step update (2.29) is tuned to two-classes, the sign of the level set function \( \phi \) must necessarily be either positive or negative. Consequently, the FMM-EM handles multi-region segmentation far more gracefully than the level set framework. Early efforts to implement multi-class segmentation via level sets made use of the theoretical result that \( N \) level set functions might code for \( 2^N \) regions by modulating their signs but were prone to erroneous classification and the formation of extraneous pockets due to notionally paired level sets evolving at different rates [123]. Later research improved accuracy at the expense of assigning each region its own level set function but required additional computational machinery to fill gaps and eliminate overlap within regions [12].

**Computational efficiency.** The FMM-EM updates in discrete steps via Bayes' rule (2.29), as opposed to level sets which must be evolved in continuous time via the solution of a partial differential equation (2.18). Continuous evolution imposes additional complexity in the form of auxiliary algorithms to determine step size, and as stability conditions often require this maximal step size to be quite small, the solution of a level set problem is often computationally expensive and slow to converge. Fast methods inspired by level sets generally dispense with the solution of a partial differential equation by reducing \( \phi \) to a small number of codes marking the contour's interior and exterior and either approximating curve evolution or dispensing with it entirely [107, 110, 127].

**Ease of implementation.** Finally, level sets are typically far more difficult to implement and tune than the FMM-EM.
• Computation of step size is not the only factor requiring an auxiliary algorithm. The desire that \( \phi \) be a signed distance function is necessarily in conflict with the log-likelihood ratio data term, which seeks to perpetually increase \( \phi \) in one region while decreasing it in another. Correcting for this tendency requires either periodic re-initialization and auxiliary smoothing iterations between update steps [111], which noticeably degrades computational efficiency, or the addition of auxiliary regularization terms [61] to (2.17) which in practice stiffen the segmenting contour.

• Implementation of (2.18) on a lattice requires some continuous regularization of the Dirac delta and its companion Heaviside step functions. Once one of the regularization methods is selected, choosing the width of the regularization involves a tradeoff between a narrow window with accurate curve evolution or a wide window which can more easily see region borders. In consequence, level set methods are known for having a short capture radius, leading to applications such as [8] in which level sets are used in post-processing to refine a segmentation derived by an algorithm with a broader view.

• Lastly, a hidden side effect of the multitude of ancillary algorithms is a proliferation of user-defined parameters which must be tuned against one another.

In chapter 3, these two algorithms will be fused into a novel segmentation method, the Bayesian Level Set algorithm, which combines their advantages while mitigating many of their deficiencies.

<table>
<thead>
<tr>
<th>Level Sets</th>
<th>FMM-EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth class transitions</td>
<td>X</td>
</tr>
<tr>
<td>Robust to noise</td>
<td>X</td>
</tr>
<tr>
<td>Soft classifier</td>
<td>X</td>
</tr>
<tr>
<td>Easy implementation</td>
<td>X</td>
</tr>
<tr>
<td>Natively multiclass</td>
<td>X</td>
</tr>
<tr>
<td>Lenient initial conditions</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2.1: Comparative advantages of Level Sets and the FMM-EM.
Chapter 3

Bayesian Level Sets for Image Segmentation

This chapter introduces a novel Bayesian level set (BLS) algorithm for multiclass image segmentation and classification that creates smooth, regular segmenting borders after the fashion of level set algorithms and yet possesses a simple, probabilistic implementation similar to the finite mixture model EM. The advantages of the BLS are discussed in section 3.5, illustrated in Table 3.2, and demonstrated on a variety of examples in section 3.7.

3.1 Formulating the cost function

The cost function for a two-class Bayesian level set algorithm may be formulated immediately from the cost function of a two-class level set problem, simply by replacing the Heaviside of the level set function $H(\phi)$ with the probability of the class conditioned on the data $q_{Z|Y}$. The remainder of this section will interpret this transformation.
3.1.1 The cost function

Begin with the two-class level set functional from (2.17):

$$\min_{\phi, \Theta} E(\phi, \Theta)$$  

(3.1)

where

$$E(\phi, \Theta) = -\int_\Omega H(\phi) \log P(Y(x) | Z(x) = 0; \Theta) \, d\Omega$$

(3.2)

$$-\int_\Omega H(-\phi) \log P(Y(x) | Z(x) = 1; \Theta) \, d\Omega$$

$$+ \beta \int_\Omega \|\nabla H(\phi(x))\| \, d\Omega + \gamma \int_\Omega H(\phi(x)) \, d\Omega$$

Now, replace the Heaviside of the level set function $\phi$ with the probability of the class conditioned on the data $q_{Z|Y}$:

$$H(\phi) = q_{0|Y}(x)$$

(3.3)

$$H(-\phi) = q_{1|Y}(x) = 1 - q_{0|Y}(x)$$

(3.4)

Under this relation, the cost function becomes:

$$\min_{q_{0|Y}, \Theta} E(q_{0|Y}, \Theta)$$

(3.5)

where

$$E(q_{0|Y}, \Theta) = -\int_\Omega q_{0|Y}(x) \log q_{Y|0}(x) \, d\Omega - \int_\Omega (1 - q_{0|Y}(x)) \log q_{Y|1}(x) \, d\Omega$$

(3.6)

$$+ \beta \int_\Omega \|\nabla q_{0|Y}(x)\| \, d\Omega + \gamma \int_\Omega q_{0|Y}(x) \, d\Omega$$

which discretizes as:

$$E(q_{0|Y}, \Theta) = -\sum_{x \in \Omega} q_{0|Y}(x) \log q_{Y|0}(x) - \sum_{x \in \Omega} (1 - q_{0|Y}(x)) \log q_{Y|1}(x)$$

(3.7)

$$+ \beta \sum_{x \in \Omega} \|\nabla q_{0|Y}(x)\| + \gamma \sum_{x \in \Omega} q_{0|Y}(x)$$
3.1.2 Interpretation and insight

Reparameterization of the level set function

Fundamentally, the relationship in (3.3) is simply a reparameterization of the level set function \( \phi \in (-\infty, \infty) \) to \( q_{0|Y} \in (0, 1) \). Equivalently, since \( q_{0|Y} \) is defined via regularization of the Heaviside function \( H(\phi) \), it is necessarily a monotonically increasing function which maps \( \phi \) from the real number line onto the unit interval \((0, 1)\).

The \textit{a posteriori} \( q_{0|Y} \) satisfies all the criteria of an admissible \( H(\phi) \). As \( \phi \to \infty \) into the interior of class 0, \( q_{0|Y} \to 1 \). Likewise, traveling into the exterior \( \phi \to -\infty \) and \( q_{0|Y} \to 0 \). Most crucially, any admissible regularization must be symmetric about \( \phi = 0 \), implying that after regularization \( H(\phi) = H(-\phi) \) for \( \phi = 0 \). \( q_{0|Y} \) fulfills these conditions since it is similarly symmetric about 1/2 and \( q_{0|Y} = 1/2 \) implies that \( q_{1|Y} = 1 - q_{0|Y} = 1/2 \) as well. This implies that the segmenting contour has been mapped from the zero level set \( \{ x \mid \phi(x) = 0 \} \) onto the equiprobable one-half level set \( \{ x \mid q_{0|Y}(x) = 1/2 \} \). Further properties are determined by the choice of regularization.

One especially elegant choice of regularization for \( H(\phi) \) is the logistic function:

\[
q_{0|Y}(x) = \frac{1}{1 + e^{-\phi(x)}} \tag{3.8}
\]

Solving for \( \phi \) yields the logit function, which is equivalent to the “log-odds” approach of [94]:

\[
\phi(x) = \log \frac{q_{0|Y}(x)}{q_{1|Y}(x)} = \log \frac{q_{0|Y}(x)}{1 - q_{0|Y}(x)} \tag{3.9}
\]

Under this regularization, the level set function \( \phi \) is equal to the \textit{a posteriori} log-likelihood ratio of the two classes, a quantity which will repeatedly appear when deriving the identity in section 3.2. The equivalence in (3.9) is very helpful in developing an intuitive understanding of that derivation.

One further elegant consequence of defining \( H(\phi) \) via the logistic function is that it implies that the regularized delta function is equal to the variance of the classification. Consider that at pixel \( x \), \( q_{0|Y}(x) \) is effectively the parameter of a Bernoulli trial
indicating whether \( x \) takes class 0 or 1. The regularized Dirac delta corresponding to (3.9) is:

\[
\delta(\phi(x)) = \frac{\partial}{\partial \phi(x)} H(\phi(x)) = q_{0|Y}(x) (1 - q_{0|Y}(x)) = q_{0|Y}(x) q_{1|Y}(x)
\]

which is identical to the equation for the variance in a Bernoulli trial.

**Expectation of the level set function**

If the level set function is perfectly accurate, (3.3) is equivalent to replacing the unregularized \( H(\phi) \) with its expectation.

A completely accurate level set function implies that:

\[
\begin{align*}
    x \in \text{Region } 0 & \Rightarrow \phi(x) > 0 \Rightarrow H(\phi(x)) = 1 \\
    x \in \text{Region } 1 & \Rightarrow \phi(x) < 0 \Rightarrow H(\phi(x)) = 0
\end{align*}
\]

In the notation of (2.26), this is equivalent to:

\[
\begin{align*}
    x \in \text{Region } 0 & \Rightarrow Z(x) = 0 \Rightarrow \delta(Z(x) = 0) = 1 \\
    x \in \text{Region } 1 & \Rightarrow Z(x) = 1 \Rightarrow \delta(Z(x) = 0) = 0
\end{align*}
\]

It follows that when \( \phi \) is correctly defined:

\[
H(\phi(x)) = \delta(Z(x) = 0)
\]

Since \( q_{0|Y}(x) \) is defined in the EM context as the expectation of \( \delta(Z(x) = 0) \), it follows that \( q_{0|Y}(x) \) is also the expectation of \( H(\phi(x)) \).

This does not, however, imply that the cost function in (3.7) is an EM for reasons which will be discussed in section 3.6.

\[\text{\textsuperscript{1}}\text{See (2.28).}\]
3.2 A Bayesian identity for minimizing the cost function

To formulate a Bayesian approach to minimizing the cost function in (3.7), begin by deriving an identity. Specifically, after first deriving the minimizing conditions, a geometric prior will be defined that encompasses all terms in the minimizing conditions which have no explicit dependence on the data. This geometric prior will be used to solve for equivalent minimizing conditions in terms of Bayes’ rule.

Minimizing conditions

To find the steady-state conditions for a minimum to the cost function in (3.7), begin by setting its derivative with respect to \( q_{0|Y}(x) \) to zero:

\[
0 = \log \frac{q_{Y|0}(x)}{q_{Y|1}(x)} + \beta \kappa(q_{0|Y}(x)) - \gamma \tag{3.13}
\]

where \( \kappa(\cdot) \) is the curvature function defined in (2.13). The log-likelihood ratio in the first term may be derived from inspection; to derive the curvature term, see appendix B.2. Since curvature is the divergence of the normal, and the normal is invariant under transformation by the logit function, (3.13) is equivalent to the level set minimizing conditions under relationship (3.9).

Solving (3.13) for the implicit, optimal \( q_{0|Y} \) is at best impractical. The level set framework computes a solution via gradient descent, which requires solving partial differential equations. However, this thesis avoids these alternatives by formulating a simpler solution in terms of Bayes’ rule.

Define the geometric prior

To define the geometric prior for the minimizing conditions (3.13), add the steady-state log-likelihood of \( q_{0|Y} \) to \( q_{1|Y} \) to both sides and group to obtain:

\[
\log \frac{q_{0|Y}(x)}{q_{1|Y}(x)} = \log \frac{q_{Y|0}(x)}{q_{Y|1}(x)} + \left( \log \frac{q_{0|Y}(x)}{q_{1|Y}(x)} + \beta \kappa(q_{0|Y}(x)) - \gamma \right) \tag{3.14}
\]
The geometric prior \( q_Z \) is defined by equating its log-likelihood ratio to the quantity within the parentheses in (3.14):

\[
\log \frac{q_0(x)}{q_1(x)} = \left( \log \frac{q_0|Y(x)}{q_1|Y(x)} + \beta \kappa \left( \frac{q_0|Y(x)}{1 - q_0|Y(x)} \right) \right) - \gamma
\]  

(3.15)

Note that the curvature of \( q_0|Y \) has been replaced with the curvature of the log-likelihood ratio of the two classes, which is the logit of \( q_0|Y \). As discussed earlier in this section, these two quantities are equal, and this version makes explicit the relationship between the log-likelihood ratios of the prior \( q_Z \) and the a posteriori \( q_{Z|Y} \). Specifically, the quantity in the parentheses is a first-order approximation to the log-likelihood ratio of steady-state soft decision \( q_{Z|Y} \) after smoothing via flow-by-curvature for time period \( \beta \). The remaining term in the prior is a weight \( \gamma \), which biases classification towards either \( Z(x) = 0 \) or 1 depending on its sign. Thus, while \( q_Z \) has no explicit dependence on the data, the minimizing prior \( q_Z \) is itself an explicit function of the minimizing a posteriori \( q_{Z|Y} \), the probability of the class conditioned on the data.

(3.15) is referred to as a geometric prior because it may be interpreted as a Naïve Bayesian [73] combination of two features with equiprobable priors, a mixture feature \( q_{m,Z} \) and a smoothing feature \( q_{s,Z} \):

\[
q_Z(x) = \frac{q_{m,Z}q_{s,Z}(x)}{\sum_{c \in [0,1]} q_{m,c}q_{s,c}(x)}
\]  

(3.16)

\[
q_{m,0} = (1 + e^\gamma)^{-1}
\]  

(3.17)

\[
q_{s,0}(x) = (1 + e^{-L(x)})^{-1}
\]  

(3.18)

\[
L(x) = \log \frac{q_0|Y(x)}{q_1|Y(x)} + \beta \kappa \left( \log \frac{q_0|Y(x)}{q_1|Y(x)} \right)
\]  

(3.19)

Naturally, \( q_{m,1} = 1 - q_{m,0} \) and \( q_{s,1} = 1 - q_{s,0} \). The factorization process in equation (3.16) represents two distinct aspects of the minimizing prior. The term \( q_{m,Z} \) is a function of \( \gamma \) which is constant with respect to pixel \( x \); this mixture prior represents the prior belief concerning the relative likelihood of the two classes and acts as a constant weight on classification. As noted in the previous paragraph, the quantity
L is a first-order approximation to the log-likelihood of classification smoothed via flow-by-curvature for time $\beta$; it naturally varies with respect to $x$ and biases coordinates towards the classification of their neighbors. The smoothing prior thus creates stochastic dependence in a spatially localized neighborhood to encourage local classification similarity. The combined prior $q_Z$ is derived from the geometry of the soft decision $q_{Z|Y}$; hence the name geometric prior.

**Solve for Bayes' rule**

Apply the prior $q_Z$ in (3.15) to the minimizing conditions in (3.14):

$$\log \frac{q_{Z|Y}}{q_{1|Y}} = \log \frac{q_{Y|0}}{q_{Y|1}} + \log \frac{q_0}{q_1}$$

(3.20)

The log-likelihood ratio of the a posteriori is equal to the sum of log-likelihood ratios of the likelihood of the data and the prior, and Bayes' rule follows immediately:

$$q_{Z|Y}(x) = \frac{q_{Y|Z}(x)q_Z(x)}{\sum_{c \in [0,1]} q_{Y|c}(x)q_c(x)} \quad Z \in [0,1]$$

(3.21)

The identity is thus derived. (3.21) is equivalent to (3.13) under the geometric prior defined in (3.16)-(3.19).

### 3.3 A Bayesian level set algorithm for multiclass image segmentation

A segmentation algorithm may be developed from four insights into the identity derived in the previous section.

*First*, the Bayesian minimizing conditions in (3.21) are equivalent to the level set minimizing conditions under the geometric prior defined in (3.16)-(3.19) and the log-odds relationship in (3.9).

*Second*, the Bayesian minimizing conditions in (3.21) are similar in form to the FMM-EM E-step update of (2.29). This suggests that the minimizing conditions in
Table 3.1: Pseudocode for the Bayesian Level Set (BLS) algorithm.

Pseudocode:

\[
\begin{array}{l}
\text{converged} = \text{false} \\
\Omega = \{\text{domain of image}\} \\
C = \{\text{class list for the BLS}\} \\
\epsilon = \text{tolerance} \\

\text{while} \neg \text{converged} \\
\quad \text{% update soft decision} \\
\quad \text{for} \{c \in C\} \\
\quad \quad \text{calculate } q_{c|y}^k \text{ according to (3.22)} \\
\quad \text{% update parameter estimates} \\
\quad \text{for} \{c \in C\} \\
\quad \quad \text{calculate } \Theta_{c}^{k+1} \text{ according to equation (3.23)} \\
\quad \quad \text{calculate } q_{z}^{k+1} \text{ according to equation (3.24)} \\
\quad \text{% Check for convergence} \\
\quad \text{if } \|q_{c|y}^k - q_{c|y}^{k-1}\| < \epsilon \text{ for all } x \in \Omega, c \in C \\
\quad \text{converged} = \text{true}
\end{array}
\]

(3.21) might be found in the same iterative manner. Beginning from appropriate initial conditions, fix the right hand side of (3.21) and update \( q_{Z|Y} \). Then, fix \( q_{Z|Y} \) and update the data modeling parameters according to maximum likelihood and the geometric prior in (3.16)-(3.19). Combined with the first insight, this results in an iterative Bayesian algorithm with the same minimizing conditions as level sets.

Third, Bayes' rule in (3.21) holds for any number of classes and not just the set \( C = \{0, 1\} \). A multiclass version of the iterative Bayesian algorithm may be developed simply by extending the data model and geometric prior to support an arbitrary number of regions.

Fourth, the geometric prior is a function of the a posteriori \( q_{Z|Y} \), and there exist other desirable geometric functions which might be substituted for (3.16)-(3.19). \( L(x) \) is a first-order approximation to flow-by-curvature applied for time \( \beta \) to the log-likelihood ratio of \( q_{Z|Y} \). However, the underlying intuition is that the previous soft decision \( q_{Z|Y} \) may be manipulated to form a prior which encourages desirable proper-
ties in the segmentation, such as local stochastic dependence in classification. Other smoothing functions might have served this purpose, and other desirable application-specific properties might have been calculated from the soft decision. The iterative Bayesian algorithm may be further extended by broadening the scope of admissible geometric priors to include any function such that:

- \( q_Z \) are valid probabilities; i.e. \( q_Z(x) \in [0, 1] \) and \( 1 = \sum_Z q_Z(x) \).
- \( q_Z \) are functions of the geometry of the previous iteration's soft decision \( q_{Z|Y} \) and have no direct dependence on the data \( Y \).

By combining these insights, a Bayesian level set (BLS) algorithm for multiclass image segmentation may be implemented in the following iterative two-stage process, for which pseudocode is provided in Table 3.1.

**Stage 1: update soft decision**

First, update the set of soft decisions \( q_{Z|Y}(x) \), the probability that each pixel \( x \) belongs to each class \( Z \) conditioned on the data \( Y \), according to Bayes' rule:

\[
q^k_{Z|Y}(x) = \frac{q^k_{Y|Z}(x) q^k_{Z}(x)}{\sum_{c \in C} q^k_{Y|c}(x) q^k_{Z}(x)}
\]  

(3.22)

Here, \( q_{Y|Z}(x) \) is the probability density of observation \( Y \) at pixel \( x \) given density modeling parameters \( \Theta^k_Z \) at iteration \( k \) for class \( Z \) and \( q^k_Z(x) \) is the smoothing prior at iteration \( k \).

**Stage 2: update parameters**

Second, update modeling parameters \( \theta_Z \) and the smoothing prior \( q_Z \) according to:

\[
\theta^{k+1}_Z = \arg\max_{\theta} \sum_{x \in \Omega} q^k_{Z|Y}(x) \log P(Y(x); \theta)
\]  

(3.23)

\[
q^{k+1}_Z(x) = S(q^k_{Z|Y}(x))
\]  

(3.24)
Here, $S(\cdot)$ is a set of geometric or smoothing functions chosen according to the conditions described in section 3.4. For a normal data model where $P(Y(x); \theta_Z) \sim N(Y(x); \mu_Z, J_Z)$, (3.23) becomes:

$$
\begin{align*}
\mu_{k+1}^Z &= \frac{\sum_{x \in \Omega} q_{Z|Y}^k(x) Y(x)}{\sum_{x \in \Omega} q_{Z|Y}^k(x)} \\
J_{k+1}^Z &= \frac{\sum_{x \in \Omega} q_{Z|Y}^k(x) (Y(x) - \mu_{k+1}^Z) (Y(x) - \mu_{k+1}^Z)^T}{\sum_{x \in \Omega} q_{Z|Y}^{k+1}(x)}
\end{align*}
$$

(3.25)  

(3.26)

See appendix C.1 for the derivation.

FMM-EM and Level Sets are a special case of the BLS

With proper choice of the geometric prior in (3.24), both the FMM-EM and level sets are special cases of a BLS with an identical data model $P(Y; \theta)$.

With the mixture prior of (3.36), the BLS simply is the FMM-EM described in section 2.3.2. However, the lack of a smoothing prior which induces local similarity in classification renders the algorithm susceptible to misclassification and topological fracturing due to noise.\footnote{Refer to Figure 2-3 for an example.}

With the first-order curvature prior in (3.30), a two-class BLS has the same minimizing conditions as a level set approach\footnote{As (3.30)-(3.31) is a first-order approximation to the curvature flow prior (3.32)-(3.34), it also follows that a BLS segmentation obtained using this prior is first-order approximation to a level set segmentation.} as shown in section 3.2. Per the log-odds interpretation (3.9), an optimal level set function may be recovered from the final \textit{a posteriori} $q_{0|Y}$:

$$
\phi(x) = \log \frac{q_{0|Y}(x)}{1 - q_{0|Y}(x)}
$$

(3.27)

While equivalent under the log-odds relationship, the solution may generally be attained more easily and efficiently via the BLS.
3.4 Geometric and smoothing priors

The key insight of the BLS is that local similarity in classification and smoothness in the segmenting contour may be promoted in an iterative algorithm via the use of a Bayesian prior $q_z$ determined by smoothing the previous iteration's soft decision $q_{z|y}$. This insight may be extended in order to develop additional priors which promote desirable properties in the segmentation output so long as two conditions are obeyed:

- The new $q_z$ must be valid probabilities; i.e. $q_z(x) \in [0, 1]$ and $1 = \sum_z q_z(x)$.
- The new $q_z$ must be functions of the geometry of the previous iteration’s soft decision $q_{z|y}$ and have no direct dependence on the data $Y$.

Multiple geometric or smoothing priors may be combined within a single algorithm using either a Naive Bayesian or normalized Bayesian approach. That is, given some set of priors $q_{i,z}$ indexed by $i$, the combined prior may be implemented as either:

$$q_z(x) = \prod_i q_{i,z}(x)$$

or

$$q_z(x) = \frac{\prod_i q_{i,z}(x)}{\sum_{c \in C} \prod_i q_{i,c}(x)}$$

(3.28)

(3.29)

The decision between these two approaches is determined by whether each prior is independently considered to be as important as the data for successful classification (3.28) or whether the priors as a set are held to be on par with the data (3.29).

The following non-exhaustive list details some potential geometric priors, including several smoothing functions, and notes some of the circumstances in which they are most appropriate for use within the BLS.

**First-order curvature smoothing**

First-order curvature smoothing is a multivariate version of the level set-inspired smoothing prior derived in (3.16)-(3.19). In this prior, the previous iteration’s soft decision is transformed via the logit function and added to its own curvature, a first-order approximation to curvature-based smoothing for $\beta$ time. The smoothed
quantity is then transformed back to a probability via the logistic function:

\[ q^k_Z(x) = \left(1 + e^{-L^k_Z(x)}\right)^{-1} \quad (3.30) \]

\[ L^k_Z(x) = \log \frac{q^k_{Z|Y}(x)}{1 - q^k_{Z|Y}(x)} + \beta \kappa \left(q^k_{Z|Y}(x)\right) \quad (3.31) \]

In a multivariate setting it is desirable to further normalize so that \( 1 = \sum_Z q^k_Z(x) \).

Curvature-based smoothing is known to be the solution to total variation-based denoising\(^4\) and is preferable to smoothing via Gaussian diffusion in numerous applications since it preferentially smooths perpendicularly to, and therefore does not blur, transitions in a surface. While this particular prior functions well in many applications, as a first-order approximation it may exhibit oscillatory behavior or even numerical instability when \( q_{e|Y} \) evolves rapidly.

**Curvature flow smoothing**

Should the user desire the benefits of curvature-based smoothing even in situations where numerical stability may prove difficult, it is possible to simply directly solve for the smoothed logit of \( q_{Z|Y} \):

\[ q^k_Z(x) = \left(1 + e^{-L^k_Z(x)}\right)^{-1} \quad (3.32) \]

\[ L^k_Z(0, x) = \log \frac{q^k_{Z|Y}(x)}{1 - q^k_{Z|Y}(x)} \quad (3.33) \]

\[ \dot{L}^k_Z(t, x) = \beta \kappa \left(L^k_Z(t, x)\right) \quad (3.34) \]

This prior must likewise be normalized so that \( 1 = \sum_Z q^k_Z(x) \). While this solution is more accurate and more stable than the first-order approximation above, it involves solving a partial differential equation as an auxiliary step to iteration. While this system is simple enough to not pose undue burden for reasonably sized images, it is perforce more computationally intensive than the first-order approximation.

\(^4\)See appendix B.2 and [100].
Gaussian smoothing

Penalizing the energy of a function leads to Laplacian diffusion as shown in appendix B.1. The Green’s function for diffusion for time \( \sigma \) is convolution with a Gaussian kernel of radius \( \sigma \):

\[
q^k(x) = G_{\sigma} \circ q^k_{Z|Y}(x)
\]  

(3.35)

As noted previously, curvature-based smoothing is preferable for many applications, yet Gaussian filtering is typically far simpler and more efficient to compute thanks to the wide availability of filtering toolboxes in most coding environments. As Gaussian and curvature-based smoothing converge as \( q_{Z|Y} \to 0, 1 \) [5], a condition which often occurs as the BLS reaches steady state, this simplification is often acceptable in practice.

Mixture ratio

Note that the mixture ratio developed for the FMM-EM (2.30) is equivalent to the limit of Gaussian diffusion as the smoothing radius tends towards infinity (\( \sigma \to \infty \)):

\[
q^k_Z(x) = \frac{1}{|\Omega|} \sum_{x \in \Omega} q^k_{Z|Y}(x)
\]  

(3.36)

This prior cannot be said to smooth in any meaningful sense; rather it weights all pixels according to the proportion of the image assigned to that class. As this has the effect of enhancing the likelihood of common classes and penalizing the likelihood of more infrequent ones, it may be useful for suppressing classes which might predominate within some local area but are nonetheless outliers with respect to the image as a whole. In this case, this prior may be combined with one of the proper smoothing priors listed above.

Fixed mixture ratio

Given prior knowledge of the mixture ratio, whether in the form of an \( a \) priori \( \pi_Z \) in the FMM-EM context (2.30) or \( \gamma \) in the level set equation (2.17), this mixture ratio
**Table 3.2:** Comparative advantages of Level Sets, the FMM-EM, and the BLS. Illustration of these examples is shown in the section indicated.

<table>
<thead>
<tr>
<th></th>
<th>Level Sets</th>
<th>FMM-EM</th>
<th>BLS</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth class transitions</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.2</td>
</tr>
<tr>
<td>Robust to noise</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.3</td>
</tr>
<tr>
<td>Soft classifier</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>3.7.3</td>
</tr>
<tr>
<td>Easy implementation</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.1</td>
</tr>
<tr>
<td>Natively multiclass</td>
<td>X</td>
<td>X</td>
<td></td>
<td>3.7.1, 3.7.4</td>
</tr>
<tr>
<td>Lenient initial conditions</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.2, 3.7.3</td>
</tr>
</tbody>
</table>

Best application: evolving the clustering wave equation unordered data segmentation

may be imposed directly as a prior. In the FMM-EM and level set contexts, this is equivalent to:

$$q_0^k(x) = \pi_0 = (1 + e^v)^{-1} \tag{3.37}$$

**Shape atlas and object recognition priors**

Shape atlas and object recognition priors are common in both level set algorithms [99] and EM segmentation [60]. The latter may be used by the BLS without modification, while the former may be converted into the BLS framework via the logistic function.

### 3.5 Advantages of the BLS

This section discusses the manner in which the BLS combines the advantages of level sets and the FMM-EM as illustrated in Table 3.2 and demonstrated in section 3.7.

*A soft classifier.* First, the BLS is a soft classifier, unlike level sets, a hard classifier. Rather than simply assigning hard labels, the BLS assigns a probability of class inclusion to each coordinate which may then be thresholded to generate a hard label should the user so desire. Furthermore, when paired with a numerically “well-behaved” smoothing prior, the BLS has never been observed to exhibit the limit-cycle behavior caused by the interplay between label assignment and parameter estimation in many hard classifiers.
The geometric prior. Second, unlike the FMM-EM, the BLS utilizes a geometric prior which may be used to impose desirable properties upon the final segmentation including local similarity in classification. While this may be helpful simply in rejecting noise and preventing misclassification due to outliers (section 3.7.3), in other applications it may be necessary simply to determine an adequate solution (section 3.7.2). Like level sets, smoothing variants of the geometric prior may be used to return pleasing, regularized segmenting contours while simultaneously gracefully allowing for regions to split and merge (topological flexibility) and suppressing the unnecessary formation of multiple small regions (topological fracturing). Unlike level sets, the geometric prior of the BLS naturally generalizes to variety of smoothing functions including, but not limited to, flow-by-curvature.

Multiclass segmentation. Third, the BLS is easily extensible to multi-region segmentation (sections 3.7.1, 3.7.3, and 3.7.4). While multiclass level set algorithms exist [12, 123], the Bayesian framework of the BLS allows it to gracefully segment an arbitrary number of image regions without maintaining the complicated computational machinery necessary to connect multiple level set functions.

Simple implementation. Fourth, even for two-region segmentation, the BLS is far simpler to implement than level sets, dispensing with the need to solve a partial differential equation, regularize Heaviside step and Dirac delta functions, or call auxiliary algorithms to smooth and maintain the level set function. The BLS thereby eliminates many of the parameters which must be independently tuned to balance these various components against one another.

Improved rate of convergence. Fifth, the BLS sometimes exhibits a dramatically improved rate of convergence with respect to level sets (section 3.7.1). The need to obey the Courant-Friedrichs-Lewy (CFL) condition and the artificially reduced horizon imposed by the delta function limit the maximum rate at which the level set function propagates. The BLS does not propagate a front; rather, the classification is allowed to update over the entire image at each iteration, and the geometric prior smoothes to preserve stability. In consequence, the BLS often finds an equivalent solution far more quickly.
Lenient initial conditions. Sixth and finally, much like the FMM-EM, the BLS typically finds an adequate solution even when provided with fairly inexact initial conditions (sections 3.7.2, 3.7.3, 3.7.4). By contrast, the relatively short "capture radius" of level sets renders them difficult to initialize in topologically complex cases, as the zero contour of the initial level set function must be located reasonably close to the true segmenting contour. For this reason, approaches such as [8] first segment images using a lower fidelity algorithm and then apply level set segmentation as a post-processing step in order to smooth and regularize class transitions.

3.6 The BLS and the EM

While the BLS is a heuristic extension to the EM, it is not, in general, a type of Expectation-Maximization algorithm. While the BLS may be expressed in the EM form for certain choices of the geometric prior,\(^5\) formulation of an EM-type inverse function for many desirable geometric priors within the mixture-model framework is a nontrivial and perhaps intractable problem. In particular, after extensive attempts to develop an antecedent for the diffusion or curvature-based smoothing priors within the EM framework, the author believes them not to exist. Therefore, in the absence of a log-likelihood function expressing the antecedent of the prior, the BLS cannot be shown to be a type of EM.

While the BLS is not an EM algorithm, the two-class cost function in (3.7) is very similar to a two-mixture version of the EM log-likelihood in (2.28). Indeed, the only change is the substitution of the geometric smoothing prior into the BLS in place of the EM's mixture prior \(\pi_c\) in order to impose local similarity in classification. This observation suggests the following means for developing extensions and variants of the BLS presented in this chapter:

1. Pose the expected log-likelihood function as a FMM-EM with a mixture prior \((\pi_Z\ defined\ in\ (2.30))\).
2. Derive the minimizing conditions for the FMM-EM as normal.

\(^5\)Indeed, using only a mixture prior, the BLS is identical to the FMM-EM.
3. Replace the mixture prior $\pi_z$ with a desired geometric prior $q_z$ subject to the conditions described in section 3.4.

Examination of the derivation in appendix C.1 shows that the BLS may be obtained simply by substituting (3.24) for (C.8). This process was also used to develop the texture-based BLS introduced in chapter 5.

Finally, while the BLS does not inherit the EM’s guarantee of convergence, in practice, it has never been observed to become unstable or fail to converge when using a numerically “well-behaved” smoothing prior such as curvature flow or Gaussian smoothing.\(^6\) The author conjectures but is unable to prove that the BLS might be shown to always converge when paired with such a prior.

3.7 Example applications of the BLS

In this section, a variety of examples are presented which demonstrate the advantages of the Bayesian Level Set algorithm.

3.7.1 A simple example

Consider the simple problem shown in Figure 3-1. This is a fairly basic problem comprised of four distinct regions whose intensities vary by unit intervals, corrupted by noise so that SNIR is 4.52 dB. Despite the fairly inexact initial conditions shown in Figure 3-1b, an accurate segmentation can be achieved in only 34 iterations using the BLS algorithm with a normal data model and Gaussian smoothing prior with radius $\sigma = 6$. Implemented in Matlab R2012a and run on an Intel i3-2120 with 4 GB of memory, this requires 9.56 seconds. See Figure 3-1c for results. Likewise, a 99.9% accurate solution may also be attained using a Brox-Weickert type multi-region level set algorithm [12] with normal data model and curvature-based regularization with $\beta = 0.05$; results are shown in Figure 3-1d. Unlike the BLS, this latter algorithm accurately models the front propagation of each level set function. As a result,\(^6\) This does not include first-order curvature smoothing, which has been observed to cause instability.
Figure 3-1: A simple example demonstrating the rapid convergence of the multiclass BLS. (a) a simple test image with SNIR 4.52 dB; (b) initial conditions, within a given contour, a region was initialized to $p_{Z,Y} = 1$ (BLS) or $\phi > 0$ (level set); (c) perfect segmentation results from the BLS, requires 34 iterations and 9.56 seconds; (d) 99.9% accurate segmentation results from a Brox-Weickert multi-region level set algorithm, requires approximately 26,200 iterations and 2.68 hours. Note that image noise leaves single pixel errors for which the level set algorithm does not correct.
convergence is much less rapid, requiring approximately 26,200 iterations and 2.68
hours on the same computer. The Courant-Friedrichs-Lewy condition and the need
to smooth the level set functions places tight upper limits on the rate at which the
wave fronts may be accurately propagated, and while this image is a simple one, it is
nonetheless reasonably large at 512 × 512 pixels.

3.7.2 Wavelet texture-based segmentation of induced pluripo-
tent stem cells

To illustrate the advantages of the BLS over both level sets and FMM-EM segmen-
tation and the potential for variation in BLS segmentation results due to choice of
smoothing prior, the BLS is applied to automate the segmentation of induced pluripo-
tent stem cell colonies (iPSC) from their growth medium. Note that the comparatively
simple texture feature (3.38) used for this task is insufficiently discriminative to dis-
tinguish pluripotent cells from differentiated ones and fails in the presence of media
containing significant extracellular material or other contaminants; such segmentation
and classification will be the burden of chapters 5 and 6.

Specifically, the texture feature will be based on wavelet energy with conditional
probabilities at iteration $k$ given by:

$$\log q^k_{Y|Z}(x) \propto - \sum_{b \in \{h,v,d\}} (r_b(x) - \mu^k_Z(b))^2$$

(3.38)

from the second horizontal ($h$), vertical ($v$), and diagonal ($d$) subbands of the station-
ary wavelet transform [88]. $r_b(x)$ is the coefficient at pixel $x$ in subband $b$, squared
and with scaling tuned to the image. $\mu^k_Z(b)$ is the mean at $b$ for class $Z$ at $k$.

Comparative segmentation results using automatically generated initial conditions
are given in Figure 3-2. Figure 3-2a shows the colony with ground truth established
under the direction of a microscopist outlined in red. Initial conditions, illustrated
in Figure 3-2b were generated automatically by independently thresholding the three
feature subbands using Otsu's method [84] and combining the results. Using a Gaus-
Figure 3-2: BLS segmentation of iPSC colonies using automated initial conditions. (a) iPSC colony, red contour shows ground truth established under the direction of a microscopist; (b) initial conditions $q_0^0 \mid Y$ generated by automatic thresholding, black = 0, white = 1; (c) segmentation results for BLS; (d) segmentation results for FMM-EM; without a smoothing prior to encourage local classification similarity, the segmentation finds only the feature peaks; (e) segmentation results for level sets; with a limited capture radius, the level becomes trapped in an undesirable local minimum; (f) Dice overlap coefficients $d$ between ground truth and automated segmentation for the set of nine iPSC colonies, $\mu_d = 0.90, \sigma_d = 0.05$. 
sian prior (3.35) \((\sigma = 4)\), the BLS successfully segments the colony as shown in Figure 3-2c. The segmenting contour in the figure demarcates a hard decision recovered by thresholding at \(q_{0|Y} = 1/2\); agreement with the ground truth may be assessed via the Dice coefficient:

\[
d = 2 |A \cap B| / (|A| + |B|)
\]  

(3.39)

\(A\) and \(B\) are sets identifying the image foreground and correspond to the area in which \(q_{0|Y} \geq 1/2\). At \(d = 0.90\), the BLS returns a very accurate segmentation, with error being due largely to oversmoothing near the colony borders.

Neither the FMM-EM nor level sets equal this performance. Due to the high variability of the feature set, successful segmentation requires enforcing local similarity, and the FMM-EM consequently fails \((d = 0.41)\), finding only the feature peaks as in Figure 3-2d. Likewise, the limited capture radius of the level set algorithm causes it to become trapped in an undesirable local minimum \((d = 0.51)\), as in Figure 3-2e.

Using initial conditions generated by the same procedure, the BLS was used to automate the segmentation of a set nine iPSC images, with Dice coefficients provided in Figure 3-2f \((\text{mean } d = 0.90, \text{ standard deviation of } 0.05)\).

**Insight on smoothing priors**

For insight concerning the choice of smoothing prior, see Figure 3-3, which compares BLS results derived using first-order curvature (3.30)-(3.31) and Gaussian (3.35) smoothing priors. Initial conditions were manually set by defining \(q_{0|Y} = 1\) in a region near the colony center, \(= 0\) in a region near the image border, and \(= 1/2\) between. A hard classification via thresholding is shown in Figures 3-3b and d. Note that the two solutions are nearly identical; the Dice coefficient between them is \(d = 0.99\). This is not surprising, as flow-by-curvature and diffusion converge as \(q_{Z|Y} \to 0, 1 [5]\), but examination of the actual soft decision \(q_{0|Y}\) in Figures 3-3a and c show that the curvature prior is far more aggressive at driving the \(a \text{ posteriori}\) classification to 0 or 1. While the Gaussian prior is simpler to compute and thus often preferable for automation, the additional rigor of the curvature prior may be desirable for some
applications, such as classification in the presence of noise.

**Retinal lesion segmentation**

Note that nothing in the above procedure was specifically designed for segmentation of stem cells; the same procedure and texture feature was used to isolate the retinal lesion in Figure 3-4. As the lesion is a pattern of black speckles within varying tissue, accurate segmentation requires the same imposition of local stochastic dependence which allowed the BLS to succeed where the level set and FMM-EM algorithms failed in classifying iPSC colonies.

**Data Collection**

Induced pluripotent stem cells (iPSCs), line ESIMR90-3 (a gift from J. Thomson, U. Wisconsin) were cultured in feeder-free mTeSR media (Stem Cell Technologies) on Matrigel coated plates (BD Sciences). Colonies were imaged 1-3 days after passaging with collagenase (Invitrogen) via phase-contrast microscopy using a 4x Nikon 0.13 NA objective on a TMS cell culture microscope with a 10M Pixel Nikon D40x SLR camera. Light levels and exposure were set to minimize noise while avoiding detector
Figure 3-4: Retinal lesion segmentation. Blue contour indicates retinal lesion. Initial conditions were automatically generated via thresholding based on the textural feature.

saturation. Images were reduced to grayscale from the green channel as phase contrast optics are optimized for green light and the camera’s Bayer color filter produces slight color registration errors.

3.7.3 Noisy brain MRI

To further demonstrate the extensibility of the BLS to multi-region segmentation and its robustness to noise, it was applied to brain MRI segmentation, a popular application of the Gaussian Mixture Model EM (GMM-EM). Since phantoms provide answer keys which allow for accurate performance evaluation, the Montreal Neurological Institute (MNI) brain phantom [23] was selected and corrupted by noise so that SNIR is 2.8340 dB. See Figure 3-5a.

The various subfigures in Figure 3-5 compare the BLS to a GMM-EM in segmenting the phantom into white matter, gray matter, and cerebrospinal fluid (CSF). As both algorithms utilize a normal data model, the key difference between the two is the Gaussian smoothing prior \((\sigma = 3)\) used by the BLS as opposed to the mixture prior used by the GMM-EM. Initial data model parameters \(\theta_Z\) were the same for both methods; representative patches were manually identified for each of the three regions as illustrated in Figure 3-5b. No bias or gain correction was performed, either
Figure 3-5: Comparison of BLS and FMM-EM for MRI phantom segmentation. (a) MRI phantom with SNIR 2.8340 dB; (b) manual reference patches: white matter (blue), gray matter (red), cerebrospinal fluid (CSF, green); (c) Dice coefficient between the true segmentation and the EM or BLS results, blue is GMM-EM while red is BLS, note that the BLS performance greatly exceeds that of the GMM-EM, especially with respect to gray matter; (d) white matter soft decision from BLS; (e) white matter soft decision from GMM-EM; (f) white matter comparison, results from (d) minus results from (e), positive (light) results indicate the BLS is more likely to classify the voxel as white matter; (g) gray matter soft decision from BLS; (h) gray matter soft decision from GMM-EM; (i) gray matter comparison as with (f); (j) cerebrospinal (CSF) fluid soft decision from BLS; (k) cerebrospinal fluid soft decision from GMM-EM; (l) cerebrospinal comparison as with (f); note the improved noise rejection of the BLS as demonstrated in (f, i, l) by suppressing noise-related speckling.
in pre-processing or in the segmentation algorithms themselves.

Figures 3-5c-1 show and compare the results from the BLS and GMM-EM in segmenting white matter, gray matter, and cerebrospinal fluid, respectively. Figure 3-5c shows the Dice coefficient between the segmentations and the phantom ground truth. Note that the BLS is far more accurate in segmenting the gray matter than the GMM-EM, which at this level of noise has difficulty distinguishing it from white matter. Figures 3-5d-e show the GMM-EM and BLS segmentations for white matter. From visual inspection, the smoothing prior in the BLS helps to suppress the salt-and-pepper speckling induced by noise seen in the GMM-EM segmentation. This observation is confirmed in Figure 3-5f, which compares the two segmentations by plotting Figure 3-5d minus Figure 3-5e; lighter voxels indicate coordinates classified as more likely to be white matter by the BLS and indicate the its superior confidence in locating white matter in the correct region and rejecting it elsewhere. Figures 3-5g-1 show a similar analysis for gray matter and cerebrospinal fluid.

Noise analysis

The noise analysis in this subsection was repeated for varying levels of noise as shown in Figure 3-6. As SNIR degrades, the FMM-EM becomes unreliable as it largely fails to recognize gray matter as a distinct class and assigns its voxels to white matter. The performance of the BLS is far more robust to noise due to the smoothing prior, which encourages local similarity in classification.

3.7.4 Massively multiclass segmentation of stem cell nuclei via hard classification

While the basic implementation of the multiclass BLS algorithm natively handles an arbitrary number of classes without requiring special mathematical techniques to link these classes together, storage requirements grow linearly with the number of classes insofar as each label requires its own array for storing its soft decision. To circumvent this limitation, massively multiclass segmentation may be performed
Figure 3-6: Noise analysis for BLS and FMM-EM for MRI phantom segmentation. (a) white matter dice coefficients as a function of SNIR; at low SNIR, FMM-EM exceeds BLS performance by classifying most of the phantom as white matter; (b) gray matter dice coefficients; at low SNIR, FMM-EM misclassifies almost all gray matter as white matter; (c) CSF dice coefficients; (d)-(g) gray matter segmentation via BLS, SNIR is noted in the caption; while performance degrades at low SNIR, the BLS always recognizes gray matter as a distinct class; (h)-(j) gray matter segmentation via FMM-EM, SNIR is noted in the caption; at low SNIR, the FMM-EM largely fails to recognize gray matter as a distinct class and assigns its voxels to white matter.
Figure 3-7: Massively multiclass BLS segmentation of fixed, early differentiating cells. (a) fixed, early differentiating cells, with approximately thirty-five cells plus the background, this presents a difficult challenge to many segmentation algorithms purely due to storage requirements; (b) initial conditions derived via thresholding by Otsu’s method returns 252 distinct labels including the background; (c) segmentation results via hard-classifier BLS, 35 distinct labels are returned, including the background; (d) segmentation results via CellProfiler, 38 distinct labels are returned, including the background.
within the BLS framework by replacing the soft Bayesian update (3.22) with the maximum a posteriori label at each iteration according to:

\[ l^k(x) = \arg \max_{c \in C} q_{ey}(x) q_c(x) \]  

(3.40)

That is, rather than store the soft probability of each class for each pixel, store only the label of the most likely pixel at each iteration. In between label updates, priors may be calculated for a given class by smoothing an array which takes on the value one at pixels assigned to that class and zero otherwise. Implemented in this manner, storage requirements remain constant no matter the number of classes. If necessary, the number of numerical operations required each iteration may be further reduced at the expense of slowing convergence by only updating the pixels in the active zone immediately adjacent to the interface between two classes.

It is worth noting that this algorithm is remarkably similar in execution to the fast-level set method developed in 2008 by Shi and Karl [107]. The chief distinction is that the hard-classifier BLS combines the smoothing prior and data term into a single update rather than the two-stage sequential update in the Shi-Karl algorithm. As hard classifiers, both algorithms are prone to limit cycle effects at steady state, in which pixels oscillate between two classifications based on interplay between parameter, label, and smoothing updates. By combining the data model and smoothing prior into a single update, however, the BLS eliminates one of these sources of limit cycle behavior and so proves slightly more stable in practice.

As an example of the type of massively multiclass problem suitably solved via application of a hard-classifier BLS, refer to the fixed stem cells in the early stage of differentiation shown in Figure 3-7a. While it is simple to formulate an initialization for this problem by thresholding according to Otsu’s method [84] and then assigning a unique label to each connected component, this results in 252 distinct regions as shown in Figure 3-7b – far too many to feasibly solve using a multiclass level set algorithm. The hard BLS solves this problem in 53 iterations. The smoothing prior (Gaussian, \( \sigma = 3 \)) serves to suppress small outlier regions; the result in Figure 3-7c...
includes only 35 distinct labels, including the background. It is instructive to compare this result to those obtained in Figure 3-7d using CellProfiler [15], a software package designed specially analyze nuclear images which segments via watershed algorithms. While the BLS does not possess the necessary auxiliary algorithms to render it a real competitor to CellProfiler, in particular those for cell splitting and merging, this example clearly shows the benefits of incorporating the smoothing prior into the BLS; segmenting contours obtained via BLS are far smoother and more regular and less prone to “leak” in jagged spurs into dim, out-of-focus portions of the image.

3.8 Summary

In this chapter, a Bayesian Level Set algorithm for image segmentation was presented which combines the advantages of level sets and the FMM-EM. Insight from an identity which expressed the level set minimizing conditions using Bayes’ rule was utilized to formulate a multiclass BLS algorithm which admits a wide variety of potential geometric priors in order to promote smooth, regular segmentations and other desirable geometric properties. After discussing the advantages of the BLS in comparison to level sets and the FMM-EM, the relation of the BLS to Expectation-Maximization was explored. The chapter concluded with a variety examples demonstrating the utility of the BLS. For ease of reference, the table comparing the advantages of the level sets, the FMM-EM, and the BLS has been reproduced at the end of this chapter.
<table>
<thead>
<tr>
<th></th>
<th>Level Sets</th>
<th>FMM-EM</th>
<th>BLS</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth class transitions</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.2</td>
</tr>
<tr>
<td>Robust to noise</td>
<td>X</td>
<td></td>
<td></td>
<td>3.7.3</td>
</tr>
<tr>
<td>Soft classifier</td>
<td></td>
<td>X</td>
<td>X</td>
<td>3.7.3</td>
</tr>
<tr>
<td>Easy implementation</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.1</td>
</tr>
<tr>
<td>Natively multiclass</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.1, 3.7.4</td>
</tr>
<tr>
<td>Lenient initial conditions</td>
<td>X</td>
<td></td>
<td>X</td>
<td>3.7.2, 3.7.3</td>
</tr>
</tbody>
</table>

Table 3.3: Comparative advantages of Level Sets, the FMM-EM, and the BLS. Illustration of these examples is shown in the section indicated.
Chapter 4

The Power of Texture for Image Classification

In computer vision, the texture of an image region is modeled or classified by characterizing the degree of randomness in image gray level intensity at a variety of scales. This is only an imperfect model of texture as it is visually apparent; human observers also judge texture according to factors which are not reducible to localized intensity fluctuation, such as the recognition of primitive objects and the identification of deterministic patterns. Nonetheless, this simplification is often quite useful in classifying image regions, including those from stem cell colony images.

Consider Figure 4-1. The pluripotent colony in Figure 4-1a exhibits a fine-grained homogeneous texture within crisp, clearly-defined borders while the differentiated colony in Figure 4-1b consists of a series of large, dark, blob-like cells in a region whose borders are somewhat more vague, a texture often described as a "fried egg" or "swampland" appearance. A horizontal line scan of the two images' gray scale intensities in Figures 4-1c-d confirms that their texture may be distinguished according to statistics of intensity fluctuation at differing scales. While both images exhibit considerable variation in intensity as a function of distance, the pluripotent intensity fluctuates at a noticeable higher spatial frequency. Wavelet analysis in Figures 4-1e-g helps to quantify this observation. Figures 4-1e-f show the empirical probability density function of the coefficients in the second and third detail subbands of the two
line scans' wavelet decompositions. The higher frequency content in the pluripotent scan (solid line) manifests in a much wider distribution at these scales than the differentiated one (dashed line). The situation is reversed at the sixth decomposition level in Figure 4-1g, where the larger cells in the differentiated colony manifest in a comparatively wider distribution at low-frequency.

Nor are these results atypical. Extensive research has shown that stem cells may be reliably classified via texture-based analysis [37, 49, 63, 64, 65, 69, 70, 72, 101, 102, 103].

In this chapter, a brief review will first be provided concerning the history and various methods of modeling image texture, categorizing these methods according to whether they handle scale in the spatial or frequency domain. Subsequently, wavelet texture methods, a frequency-domain approach, will be validated for use in classifying stem cells. The chapter will conclude by formulating an approach to estimating texture features for image segmentation and reviewing a set of frequency-domain features appropriate to this method.

4.1 Historical review of image texture modeling

With regard to computer vision, the most significant result of research on human textural perception performed during the 1960s and 1970s was the Julesz conjecture: Two textures are not preattentively distinguishable if their second-order statistics are identical. First-order statistics concern the intensity of any given point in the perceptual field when considered in isolation, while second-order ones concern the relationship between the intensities of two points at a particular distance and orientation. Psychophysics research indicated that third- and higher-order statistics based on point triples, quadruples, etc. could not be distinguished by humans without direct attention. While Julesz himself later disproved the conjecture by constructing preattentively distinguishable texture patterns comprised of glyphs with identical second-order statistics but varying numbers of terminators, corners, etc. [50], the insight behind the conjecture serves as the basic inspiration for subsequent computer
Figure 4-1: Image texture as statistical fluctuation in image gray scale intensity. (a, b) unstained, live cell pluripotent (a) and differentiated (b) stem cell colonies; bar is 1,128 μm; (c,d) line scans of the intensity of the bars in (a,b), note the pluripotent intensity varies at greater amplitude and finer scale than the differentiated; (e-g) empirical probability density functions for three wavelet detail subbands in (a,b), pluripotent is the solid line while differentiated is dashed; in the second (e) and third (d) detail subbands, the thicker distribution and wider tails of the pluripotent (solid) density indicate comparatively more edges at finer scales; the situation is reversed in the sixth detail subband (g), which is on the order of the size of the differentiated cells.
vision texture modeling. The two basic postulates are:

1. Texture is the relationship between image intensity values at varying distance (or scale) and orientation.

2. Probabilistic or statistical models are the appropriate means of quantifying this relationship.

Insofar as computer vision is a field of engineering, there is perforce a third consideration:

3. Quantification of this relationship should be as simple and efficient to implement as possible.

In practice, the interplay between the first and third concerns is typically determinative; the choice of method for modeling the distance, whether by scale or neighborhood, constrains the set of available statistical models. Algorithms for textural classification thus generally align into one of two categories based on the manner in which they model distance. One approach models distance according to neighborhoods formulated in the spatial domain, while the second characterizes it according to scale expressed in the frequency domain. In practice, the distinction lies in whether the neighborhood is best represented via a graph (spatial domain) or filter (frequency domain).

The remainder of this section will provide a brief précis of several of the major algorithms in this field, classified into the two categories above as illustrated in Table 4.1. For a more extensive review, refer to [89].

4.1.1 Spatial domain neighborhood models

*Spatial domain neighborhood texture models* characterize the distance aspect of texture by directly specifying pixels or neighborhoods of pixels in the lattice of image coordinates. The original model in this category is the so-called Gray Level Co-Occurrence Matrices (GLCMs) [42, 43]. Originally introduced in 1973, they are one of the first modern image texture models and still enjoy extensive use in medical imaging, e.g.
GLCMs define a set of pairwise neighborhoods in terms of their horizontal and vertical offset; for each particular neighborhood, features such as energy or correlation are calculated from the empirical histogram of energy intensity pairs. The next iteration in the spatial domain category modeled space by imposing a Markov Random Field (MRF) structure on the image lattice so that the probabilistic texture features became the MRF clique potentials [21, 26, 41].

Theoretically, these models are incredibly flexible. GLCMs offer a framework for characterizing any pairwise pattern in an image, and MRFs extend this by allowing for arbitrary neighborhoods with user-defined structure. In practice, however, this flexibility is greatly mitigated by implementation concerns. Deriving an accurate histogram for GLCMs typically requires that intensity be binned into a few codes, and finite computational resources require choosing some subset from the nearly infinite set of possible pairwise neighborhoods. Likewise, the complexity of algorithms used to estimate clique potentials (cf. [41]) means that in practice MRFs are generally limited to pairwise Ising models. In practice, therefore, both GLCMs and MRFs often only consider the four-connected neighborhood around a given pixel.

For this reason, research in image texture modeling shifted towards frequency domain models, particularly wavelets, in the late 1990s, as these methods could examine wider scales via a comparatively simple and efficient implementation. A notable exception to this trend is Local Binary Pattern (LBP) analysis, which defined a series of

<table>
<thead>
<tr>
<th>Spatial domain neighborhood</th>
<th>Frequency domain scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>early 1970s</td>
<td>GLCMs</td>
</tr>
<tr>
<td>early 1980s</td>
<td>Markov random fields</td>
</tr>
<tr>
<td>late 1990s</td>
<td>Wavelets</td>
</tr>
<tr>
<td>early 2000s</td>
<td>Local binary patterns</td>
</tr>
<tr>
<td>late 2000s</td>
<td>Patch exemplars</td>
</tr>
<tr>
<td>early 2010s</td>
<td>Compressive sensing</td>
</tr>
</tbody>
</table>

Table 4.1: The history of image texture modeling, with approximate dates of the introduction of key algorithms.

[98]
neighborhoods in terms of the Euclidean distance to a central pixel [81, 90]. Thresholding based on the intensity of the central pixel allows the pixels in a neighborhood to be represented as a binary code, and statistics may be calculated from the set of binary codes produced by evaluating a given neighborhood across an entire image.

In recent years, advances in computational technology have seen the reintroduction of spatial domain texture models in the form of patch exemplar libraries [121]. Here, the neighborhood is defined as the entire rectangular patch of a certain width about a central pixel. The entire set of patches from the set of training images is clustered to identify a library of characteristic patch patterns. Each patch in a test image is assigned to the nearest library item, and the histogram of this assignment becomes a statistical feature for the image texture.¹

4.1.2 Frequency domain scale models

Frequency domain scale models characterize texture distance in terms of scale in the frequency domain without reference to specific locations in the lattice of image coordinates. Instead, the neighborhoods in these models are defined in terms of filters. While the seminal paper introducing the field [118] utilized Gabor functions due to their similarity to elements in human biological vision, this approach was chiefly energized by the flowering of wavelet analysis in the 1990s [68].

Wavelets decompose information into multiple subbands in terms of both orientation and scale in a manner far more computationally efficient than the spatial domain methods previously in use. They were thus a natural means of characterizing texture, and it was soon recognized that the statistical distribution of the coefficients in the wavelet detail subbands could be used to accurately model texture in several applications [29, 35]. Due to their demonstrated utility in stem cell classification [49, 69, 72], this methodology will be described in more detail in section 4.2.1.

In order to improve tuning with respect to both orientation and scale, a similar yet competing approach replaced wavelet decomposition with filter banks containing a

¹The computational requirements for this can be very large. Consider that a patch size of $m \times m$ on an image of size $N \times N$ results in $N^2$ patches, each with size $m^2$. 
variety of features in multiple orientations and scales [120]. For instance, a particular feature might be an edge or bar filter based on derivatives of Gaussian functions; the orientation of this feature may be altered via simple rotation of the filter, and its scale tuned by adjusting the parameter of the Gaussian.

One of the key contributions of the filter bank method was demonstrating that feature dimensionality might be reduced without adversely affecting classification accuracy by retaining only the maximal response over all orientations of a given feature at a given scale [120]. For example, consider feature $F(x, \sigma, \omega)$ which is the response of an edge filter with scale $\sigma$ and orientation $\omega$ at pixel $x$. Dimensionality may often be reduced without significant loss of accuracy by retaining:

$$F(x, \sigma) = \max_{\omega} F(x, \sigma, \omega)$$  \hspace{1cm} (4.1)

Though this removes orientation selectivity, many textures of interest (stem cells included) are globally isotropic, meaning that the texture should "look the same" as one traverses the image regardless of direction. The intuition is that while the response of a feature at a given scale matters, the orientation of that response is insignificant.

Recently, inspired by advances in compressive sensing, randomized filter banks have also been used to generate texture features [62].

### 4.2 Wavelet-based texture classification of stem cell patches

It was previously asserted that wavelet-based texture methods could be used to classify stem cell image patches with a high degree of accuracy. Before utilizing similar techniques for stem cell colony image segmentation, it is helpful to demonstrate that this is indeed the case and review the various techniques that might be used to accomplish it. In the process, new techniques will be introduced for adaptive and hierarchical windowing which may be used to classify both very small and very large images accurately and efficiently.
4.2.1 Wavelet-based texture analysis

As noted above, image texture is largely characterized via the random fluctuation of gray level intensity at various scales. Wavelet analysis offers an obvious means to address this insofar as it decomposes a signal locally according to both scale and orientation. Characterization of random intensity fluctuation may be accounted for by modeling the coefficients in the wavelet detail subbands as observations of some random process. In implementation, this proceeds in three steps.

First, apply the wavelet decomposition to a texturally homogeneous image region. An $n$-level decomposition results in three subbands per level, with one apiece being oriented horizontally, vertically, and diagonally. These $3n$ detail subbands are assumed to contain the texture. The approximation subband is discarded as the low frequency information it contains is assumed to be texturally irrelevant, e.g. fluctuations in illumination, shape changes in the underlying surface, etc.

Second, each detail subband in the region is modeled according to the empirical distribution of its coefficients via some user-selected probability density function estimator. For ease in modeling, coefficients within a subband are assumed to be statistically independent of their neighbors and subbands are assumed to be independent of one another. Distributions are typically assumed to be zero-mean; as the detail subbands are the output of high-pass filters, this is true in the absence of a gradient in the image's intensity. This set of empirical probability density functions constitutes the texture model of the region.

Third, the textural dissimilarity between two image regions is quantified according to the information divergence between their respective distribution estimates. Let $f_i$ be the distribution of subband $i$ in one region and $g_i$ be the distribution of the same subband for some other region. The information divergence (i.e. Kullback-Leibler divergence or KLD) between these two distributions is:

$$D_{KL}(f_i \| g_i) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} dx$$

Other divergence measure exist, such as the L1 ($\int |f - g|$), Bhattacharyya, and other
members in the Renyi family, but the Kullback-Leibler divergence is especially convenient since it admits tractable closed-form solutions for many common density models. For \( n \) levels of decomposition, the total dissimilarity between two textures is the sum over all their subbands:

\[
D(f, g) = 
\sum_{i=1}^{3n} D_{KL}(f_i \| g_i) + D_{KL}(g_i \| f_i)
\]  

(4.3)

Since in general \( D_{KL}(f_i \| g_i) \neq D_{KL}(g_i \| f_i) \), the above sum contains both quantities in order to symmetrize the distance between textures.

There are several techniques which may be used to model the density of wavelet coefficients, and in practice the selection of a particular method is heavily influenced by the tradeoff between accuracy and computational simplicity. The two parametric and non-parametric models below are characteristic of this tradeoff. As the former rely on the estimation of a comparatively small set of parameters, they are generally quicker to compute and require much less storage than the latter.

**Generalized Gaussian distribution**

Mallat noted in his original paper on the wavelet decomposition [68] that the distribution of coefficients in a detail subband often resembled a symmetric, unimodal
generalized Gaussian distribution (GGD):

\[
f(x; \alpha, \beta) = \frac{\beta}{2\alpha \Gamma(1/\beta)} e^{-(|x|/\alpha)^\beta}
\]  

(4.4)

\(\alpha\) and \(\beta\) are respectively the width factor and shape parameter, while \(\Gamma(\cdot)\) indicates the gamma function. As noted above, the process mean is assumed to be zero. As shown in Figure 4-2, the GGD is a flexible family of distributions which admits the Gaussian (\(\alpha = \sqrt{2}\sigma\)), Laplacian (\(\alpha = \sigma/\sqrt{2}\)), and uniform (\(\beta \to \infty\)) distributions as special cases with standard deviation \(\sigma = \alpha \sqrt{\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}}\).

The utility of this distribution for texture characterization was shown in [29, 35], which respectively developed moment-matching and maximum likelihood procedures for calculating \(\alpha\) and \(\beta\) and provided a closed-form solution for the divergence between two GGD processes:

\[
D_{GGD}(f_1 \parallel f_2) = \log \left( \frac{\alpha_2 \beta_1 \Gamma(1/\beta_2)}{\alpha_1 \beta_2 \Gamma(1/\beta_1)} \right) + \left( \frac{\alpha_1}{\alpha_2} \right)^{\beta_2} \frac{\Gamma((\beta_2 + 1)/\beta_1)}{\Gamma(1/\beta_1)} - \frac{1}{\beta_1}
\]  

(4.5)

Symmetric alpha-stable distribution

Another density family used for texture classification is the symmetric \(\alpha\)-stable distribution family (SaS) which models densities with heavier tail probabilities than those admitted by the GGD. Multiple formulations exist for its characteristic function; type 2 from [119] is:

\[
\phi(\omega; \alpha, \gamma) = \exp(-\gamma^{\alpha} \left| \omega \right|^\alpha)
\]  

(4.6)

\(\alpha\) and \(\gamma\) are respectively the characteristic exponent (0 < \(\alpha\) ≤ 2) and dispersion (\(\gamma > 0\)). Like the GGD, the location parameter is assumed to be zero. Unlike the GGD, the density function of the SaS only has closed form expression in a few special cases, including the Cauchy (\(\alpha = 1\), scale \(\gamma\)) and Gaussian (\(\alpha = 2\), \(\gamma \leq \sigma/\sqrt{2}\)) distributions. Therefore, [119] noted that the normalized SaS characteristic function forms a valid density function and the closed-form KLD between two such distributions is equivalent to (4.5) with \(\alpha_{SaS} = \beta\) and \(\gamma = 1/\alpha_{GGD}\). Maximum
likelihood methods exist for finding these parameters [80, 122].

Ahmad-Lin estimation using Parzen kernels

The two probability density function families above assume distributions which are symmetric, unimodal, and zero-mean. For image texture analysis, these assumptions are usually warranted. When they are not, as when a noticeable gradient exists in the image intensity, it is still possible to estimate the coefficient distribution using nonparametric methods which make no assumption regarding the underlying shape of the distribution, albeit at greatly increased computational cost and the necessity of storing the entire distribution.

The distribution $f_X$ of some data set $X$ is modeled using a non-parametric kernel density estimator thus:

$$f_X(x) = \frac{1}{|X|} \sum_{\zeta \in X} K_\sigma(x - \zeta)$$  \hspace{1cm} (4.7)

For the sake of concreteness, kernel $K_\sigma(\cdot)$ is assumed to be Parzen (Gaussian) with radius $\sigma$. A general rule for selecting the kernel bandwidth is [109]:

$$\sigma = 0.9 \min(\hat{\sigma}, \hat{\rho}/1.34) |X|^{-1/5}$$  \hspace{1cm} (4.8)

where $\hat{\sigma}$ and $\hat{\rho}$ are the empirical standard deviation and interquartile distance.

The Ahmad-Lin entropy estimate [2] of this distribution is:

$$\hat{H}(X) = -\frac{1}{|X|} \sum_{x \in X} \log f(x)$$  \hspace{1cm} (4.9)

$$= \log |X| - \frac{1}{|X|} \sum_{x \in X} \log \sum_{\zeta \in X} K_\sigma(x - \zeta)$$  \hspace{1cm} (4.10)

Note that methods used to calculate the distribution parameters are quite different; only the equation for calculating the divergence is similar.
Analogously, an estimate for the KLD between two sets of observations $X$ and $Z$ is:

$$D_{PZN}(X||Z) = \frac{1}{|X|} \sum_{\chi \in X} \log f_X(\chi) - \frac{1}{|Z|} \sum_{\chi \in X} \log f_Z(\chi)$$  \hspace{1cm} (4.11)

$$= \log \frac{|Z|}{|X|} + \frac{1}{|X|} \sum_{\zeta \in Z} \log \sum_{\chi \in X} K_\sigma(\zeta - \chi) - \log \sum_{\xi \in Z} K_\sigma(\zeta - \xi)$$  \hspace{1cm} (4.12)

**Ahmad-Lin estimation using the Loftsgaarden-Quesenberry estimator**

Another non-parametric density estimate is that of Loftsgaarden-Quesenberry, which models the density according to the distance to its $k^{th}$ nearest neighbor:

$$f(\chi) = \frac{k}{\rho_k^d(X, \chi) v_d |X|}$$  \hspace{1cm} (4.13)

Here $\rho_k(X, \chi)$ is the distance to the $k^{th}$ nearest neighbor of $\chi$ in $X$, excluding any precisely at $\chi$. Likewise, $v_d$ is the volume of a unit ball in $\mathbb{R}^d$. As a rule of thumb, $k$ is set to $\sqrt{|X|}$. Under the Ahmad-Lin estimator, the KLD between two sets of observations $X$ and $Z$ is approximately [7]:

$$D_{LQ}(X||Z) = \log \frac{|Z|}{|X| - 1} + \frac{d}{|X|} \sum_{\chi \in X} \log \frac{\rho_k(Z, \chi)}{\rho_k(X, \chi)}$$  \hspace{1cm} (4.14)

**4.2.2 Application to nuclei classification**

To demonstrate the various methods for estimating the distributions of wavelet detail coefficients and the divergence between them, the four methods were applied to a set of images of fixed nuclei. As nuclei appear as bright patches against a dark background, they are fairly easy to segment using conventional algorithms. Subsequently, adaptive windowing was applied as described below in order to extract wavelet coefficients from such small, irregular areas. The statistical, multiresolution texture model was then applied using a variety of wavelet types and each of the four KLD estimation methods.

A $k$-Nearest Neighbor (kNN) classifier was applied to fifty-three cells, samples of which are shown in Figure 4-3a. This subfigure shows a matrix where each entry represents a thresholded KLD comparison between each pair of nuclei. A white
10-Nearest Neighbor Classification: Decomposition Levels: 2

(a) kNN wavelet classification

nearest neighbor classification; class 1 (early differentiated nuclei): images 1-25, class 2 (pluripotent nuclei): 26-53; for each nuclear image on the y-axis, the ten nearest neighbors (lowest summed KLD) have been indicated by a white square along the x-axis; (b)-(c) classification accuracy with two and three decomposition levels for varying wavelet and pdf/KLD estimators; classification accuracy is highly robust to the number of decomposition levels and choice of wavelet, and the computationally more convenient parametric (GGD, SoS) methods exhibit comparable accuracy to non-parametric (A-L, L-Q) ones; A-L denotes the Ahmad-Lin estimator with Parzen kernels, and L-Q the Ahmad-Lin estimator paired with the Loftsgaarden-Quenbey estimator.

Figure 4-3: Nuclei classification via wavelet texture modeling. (a) Kullback-Leibler divergence (KLD) nearest neighbor classification; class 1 (early differentiated nuclei): images 1-25, class 2 (pluripotent nuclei): 26-53; for each nuclear image on the y-axis, the ten nearest neighbors (lowest summed KLD) have been indicated by a white square along the x-axis; (b)-(c) classification accuracy with two and three decomposition levels for varying wavelet and pdf/KLD estimators; classification accuracy is highly robust to the number of decomposition levels and choice of wavelet, and the computationally more convenient parametric (GGD, SoS) methods exhibit comparable accuracy to non-parametric (A-L, L-Q) ones; A-L denotes the Ahmad-Lin estimator with Parzen kernels, and L-Q the Ahmad-Lin estimator paired with the Loftsgaarden-Quenbey estimator.
Figure 4-4: Comparing GLCM and wavelets for cell nucleus classification. (a) GLCM nearest neighbor classification using the same procedure as Figure 4-3a; the correlation of the GLCM proved the most accurate feature, while other features performed worse; (b) comparison of the classification accuracy of wavelet-based and GLCM texture analysis; angular second moment, contrast, and correlation are three GLCM features; each performed noticeably worse than wavelet-based textural analysis, represented here by the Daubechies-6 (db6) wavelet with two decomposition levels as given in figure 4-3b.

Subfigures 4-3b-c show classification accuracy as a function of the wavelet and number of decomposition levels and indicate that, for these images, accuracy is highly robust to variations in either. This result extends previous studies of the decomposition wavelet [96], which were limited to the Daubechies and Gabor families. In cases in which cell images exhibit noticeable fine-scale variation, however, it might be critical to extract as many decomposition levels (and thus variation in scale) as possible. The GGD is the preferred density function model insofar as the accuracy of the four methods is comparable, and the GGD is easiest to compute.

To assess these results, it is helpful to compare them to another common tex-
Figure 4-5: *Adaptive windowing for classification of small, irregularly shaped regions.* (a) a pluripotent nucleus with blue segmenting contour; (b) modeling coefficients selected at each decomposition level, red are the coefficients for level 3, orange at level 2, and green at level 1; light blue are interior to the cell but exterior to level 1; dark blue are exterior to the cell.

Texture measure: GLCMs [42]. Two GLCMs were calculated, with offsets of a single horizontal and vertical pixel, respectively. The GLCM histogram was given eight bins, and kNN classification was applied using the angular second moment, contrast, and correlation features. Figure 4-4a shows classification results for correlation, the best performing feature at 85% success; by contrast, wavelet-based texture analysis generally exceeded 95% success, with several wavelet / density function model combinations achieving perfect classification. GLCM results are compared to wavelet texture (two decomposition levels, Daubechies-6 wavelet, GGD model) in Figure 4-4b. The superior performance of the wavelet-based texture model over the GLCM may be attributed to its more efficient and precise representation of texture in terms of intensity variation at multiple scales. Many GLCM offset combinations would be required to track the scale of variation recorded in a wavelet subband. Likewise, the necessary binning of the GLCM degrades the precision in modeling that variation in comparison to a probability density function model of the wavelet subband.

**Adaptive windowing**

Wavelet pyramid analysis decomposes two-dimensional signals in square or rectangular regions. In cases involving small or irregularly shaped objects, such as cell nuclei, isolating a rectangular region of sufficient size to accurately estimate density parameters may prove impossible. Nor is it possible merely to select all coefficients within
the region of interest; as wavelet filtering occurs over a window, this would result in coefficients comprised of information from within and without the region of interest being admitted to the estimator.

Adjustable windowing [65] may be used to extract all and only those coefficients in a given subband which were derived from information contained in the region of interest. Begin with some segmentation mask $M_1$ indicating the region, and consider any particular pixel $x$ within it. To determine whether a $2 \times 2$ window whose bottom right pixel is $x$ is entirely within the mask, it is sufficient to check whether the pixels to the left, above, and diagonally up and to the left of $x$ are also within the mask. This check may be performed for the entire image simply by taking the intersection of $M_1$ with itself translated one space to the right, itself translated one space down, and itself translated one space down and to the right. Iterating on this procedure will determine the mask $M_k$, indicating all pixels such that a $k \times k$ window extending up and to the left is also within the region of interest. Formally:

$$M_{k+1} = M_{k}^{0,0} \cap M_{k}^{0,1} \cap M_{k}^{1,0} \cap M_{k}^{1,1}$$

(4.15)

where $M_{k}^{i,j}$ denotes $M_k$ right-shifted $i$ pixels and down-shifted $j$ pixels.

To apply this to small, irregular objects it is helpful to apply the stationary wavelet transform (SWT) [88], which upsamples the filter rather than downsampling the image, in order to maximize the coefficients available. For base filter length $n_f$, the SWT decomposition window at level $d$ has length $k = 2^{d-1} (n_f - 1) + 1$. For this $k$, $M_k$ as determined in (4.15) indicates the set of coefficients at level $d$ which were derived solely from information within the region of interest. This procedure is demonstrated in Figure 4-5.

### 4.2.3 Application to colony classification via windowing

A stem cell colony image represents a very different class of problem from stem cell nuclei. While it would be fairly easy to use textural methods to classify the regions in a segmented image, the differing image regions are too texturally complex to be
(a) texture library for 256 × 256 pixel windows

(b) window by window classification of a pluripotent colony

(c) differentiated colony

(d) algorithm robustness to classifier parameters

(e) comparison between GGD and SaS model

Figure 4-6: Window-based stem cell colony classification using wavelet texture models. (a) the statistical dissimilarity measure (KLD) between two windows is visualized via color in each entry of the matrix, from blue (similar) to red (very dissimilar); thus, the diagonal elements are each dark blue as each window is texturally identical to itself, and intra-class KLDs appear as bluish blocks along the diagonal; (b) window by window classification of a pluripotent hESC colony; three passes are used to classify the original image, with intermediate results shown after the classification and border detection phases: blue is pluripotent, green is differentiated, red is exterior, and gold is unknown; (c) differentiated colony, clearly distinguished from pluripotent; (d) algorithm robustness to changes in classifier parameters; (d-i) is constant wavelet (Daubechies-4), classifier parameters \((k, k_u)\) vary; blue is pluripotent, green is differentiated, red is exterior; bar indicates extent of 90% confidence interval, diamond is maximum a posteriori (MAP) estimate of accuracy; (d-ii) is pluripotent, (d-iii) is differentiated, and (d-iv) is exterior; these with constant classifier parameters \((k, k_u) = (7, 5)\) and varying wavelet; (e) comparison between GGD and SaS models, Daubechies-4 wavelet, \((k, k_u) = (7, 5)\); (d) (e) indicate performance is robust to choice of wavelet, classifier parameters; GGD performance equals or exceeds SaS performance.
isolated using conventional segmentation algorithms. In lieu of a segmentation, it is possible to tile the image into windows which may be classified separately using a library-based kNN algorithm.

**Classification procedure**

Building an image texture library is a matter of identifying the textures which span a set of images, manually selecting a number of reference samples for each texture, and computing and storing the textural parameters for each sample. The results of this procedure are illustrated in Figure 4-6a. A GGD model was selected for this procedure, and each square entry in the matrix represents the KLD between a pair of sample windows from the learning set, with the color indicating the degree of dissimilarity. The intra-library KLD shows the grouping of library textures into six classes, one differentiated, three exterior, and two pluripotent (from top left to bottom right), shown as bluish (low dissimilarity) blocks clustered along the diagonal of the matrix. Note the resemblance, though the two remain distinguishable, between the first (differentiated) and last (pluripotent) classes, shown in the greenish bar off-diagonal in the top left corner, and the textural heterogeneity of the differentiated class, with its lighter shade of blue.

Classifying the windows of test images against the library is performed using a kNN classifier. To reject ambiguous cases, models are only considered as neighbors if their dissimilarity is below some ceiling value, and the classification is only accepted if at least $k_n \leq k$ neighbors concur. In practice, classification is shown to be robust to the choice of $k$, $k_n$, and the dissimilarity ceiling.

For ease and speed of computation, it is desirable to classify relatively large windows, but this initial result may be refined by identifying suspicious windows which are likely to have been misclassified, dyadically partitioning them (e.g. so that a single $256 \times 256$ window becomes four $128 \times 128$ windows), and re-classifying the individual partitions. In this application, two spatial reasoning heuristics govern this *hierarchical image windowing* procedure. First, partition any window failing the kNN test, i.e. fewer than $k_n$ neighbors exist and concur. Second, partition any differentiated win-
window that borders upon a window of another class; since differentiated hESC regions are characterized by a type of textural inhomogeneity, border windows are prone to being misidentified as differentiated. This process may be reiterated until characteristic features (e.g. differentiated cellular clumps, pluripotent white lines, etc.) are on the order of the window size, $64 \times 64$ pixels in this application.

This procedure is illustrated for a pluripotent colony in Figure 4-6b, and results are provided for a differentiated colony in Figure 4-6c. Blue windows are classified as pluripotent, green as differentiated, and red as exterior. Gold windows are unknown or could not be classified. These are very few in number.

Performance results

Classification of inhomogeneous hESC colony images is quite successful. Typically, identification of pluripotent windows is near perfect (99%, with 90% confidence interval (CI) [0.9812, 0.9986]). All but one window among the very small percentage of misclassified pluripotent windows are assigned to the colony exterior, so pluripotent windows are essentially never classified as differentiated. Differentiated and exterior accuracy is 89% (90% CI [0.8547, 0.9196]) and 96% (90% CI [0.9432, 0.9812]), respectively, with no differentiated region was misclassified as pluripotent.

Results in Figure 4-6d give accuracy results for a variety of classification ($k, k_n$) and modeling (wavelet basis) parameters. Blue indicates pluripotent, green differentiated, and red exterior. The bar indicates the 90% CI and the diamond the mean of the distribution. Note that pluripotent accuracy is highly robust to changes in these values. Differentiated accuracy varies but is acceptable with wavelets of eight taps or less (sym4 and left). This for the GGD model. By way of comparison, results with SaS models are shown in Figure 4-6e for the “typical” set of model parameters. While pluripotent accuracy is relatively unchanged, differentiated accuracy decreases noticeably (82%, down from 89%), validating the use of GGD density models in this application.
4.3 Texture features for stem cell colony segmentation

As noted, many popular image texture models are ill-suited for segmentation since they rely on statistics calculated for rectangular, texturally homogeneous windows. For texture-based segmentation, texture must be modeled for every coordinate in an image, including those near the interface between two distinct textures. This can be problematic; the feature distribution near a textural interface represents a superposition of the distributions of the distinct, individual textures. The result is that statistics calculated in near the interface may more closely resemble another class entirely.

Accordingly, the first part of this section describes a framework for localizing texture models for use in image segmentation. The remainder of this section consists of brief descriptions of the textural features which will be considered for stem cell segmentation and introduces the set of codes used to abbreviate them. \(d\) denotes a wavelet decomposition level, radius is \(\sigma\), and \(n\) is window size.

4.3.1 Localizing texture models for image segmentation

Efficient segmentation requires that each pixel be assigned to a class based on the smallest possible set of local features. Localization is necessary insofar as neighboring pixels might be from another class and hence have different characteristics. Minimizing the set of features is necessary due to storage limitations. As each feature must be stored for each pixel, memory requirements increase in proportion to the number of features; it is infeasible to store an entire nonparametric density estimate for each pixel in an image. These two objectives are realized via the following three-stage procedure:

1. Calculate a desired frequency domain texture feature \(F(x)\) at each pixel \(x\) in the image.

2. If \(F\) is an oriented feature, reduce dimensionality by retaining only the maximal
response at any orientation.

3. Estimate the local feature expectation via convolution with a Gaussian kernel.

**Calculate a frequency domain texture feature**

Contemporary spatial domain texture models such as local binary patterns [81, 90] and patch exemplars [121] generally rely on histograms of discrete codes. As texture feature localization will be performed by estimating the local feature expectation, only frequency domain features are appropriate for this procedure.

**Retain maximal response over all orientations**

Inspired by [120], the set of oriented features is minimized by retaining only the maximal response over any orientation at a particular scale as in (4.1).

For instance, wavelet-based multiresolution methods (e.g. [35]), one of the most powerful oriented features, compute multiple features at a particular scale or decomposition level, each corresponding to a different orientation within the subband. As each subband is effectively the output of an oriented band-pass filter, the result is an oriented edge detector tuned to a particular scale. Consequently, the band with the greatest response is the one most closely perpendicular to the edges present in the image and thus the the best measure of the image’s local structure. This observation holds for other oriented features; the maximal response of a bar feature lies at the orientation closest to any bar structures in the image, the maximal entropy difference is at the orientation perpendicular to textural transitions, etc.

**Estimate expectation via Gaussian convolution**

Localization proper is next performed by estimating the local expectation of features via filtering with radially symmetric Gaussian kernels. Many statistics may be calculated as the expectation of various quantities, and a Gaussian kernel serves as the coefficients of a weighted average in a neighborhood centered about a particular pixel. While there are other methods for calculating a weighted local average, Gaussian fil-
ters are especially attractive since they prioritize the pixels closest to \( x \) while allowing for a principled way to control neighborhood size. In summary, for \( G_\sigma \), a Gaussian kernel with radius \( \sigma \):

\[
\hat{E}_\sigma [F] = G_\sigma \circ F
\]  

In practice, choosing the radius \( \sigma \) requires a tradeoff between precision (small \( \sigma \)) and estimation accuracy (large \( \sigma \)).

### 4.3.2 Wavelet-based features (dcmag, dcmag, and wave)

As discussed previously, wavelets perform excellently when used to classify texturally homogeneous stem cell image patches. This section customizes wavelet methods for use in segmentation.

**Wavelet standard deviation (wave)**

The methods listed in section 4.2.1 for modeling the density of coefficients in a patch are infeasible for texture-based segmentation. While nonparametric methods would admit analysis under (4.16), storing an empirical density estimate at each pixel location requires far too much computational memory. Conversely, while parametric methods are tractable with respect to memory, the iterative solvers used to estimate these problems are not tractable under (4.16).

Instead, a local estimate of the standard deviation serves as a single statistic for characterizing the wavelet coefficients in a given subband. Though less accurate than an actual density estimate, standard deviation is easily estimated via simple filtering operations. For \( D_{bd} \), the coefficients at decomposition level \( d \) and orientation band (vertical, horizontal, or diagonal) \( b \), the estimated variance is:

\[
\text{Var}_\sigma D_{bd} = \hat{E}_\sigma [D_{bd}^2] + \hat{E}_\sigma [D_{bd}]^2 = G_\sigma \circ D_{bd}^2 + [G_\sigma \circ D_{bd}]^2
\]

Standard deviation, of course, is the square root of this quantity. Over the three
Figure 4-7: Impulse response of (a) DWT and (b) DT-CWT filters; note that the top row in (b) is the filter real part with the corresponding imaginary part on the bottom row; use of the overcomplete DT-CWT basis provides improved directional tuning which is desirable for texture modeling; image from code courtesy of [13].

orientation subbands $b$ at decomposition level $d$:

$$\text{waved} - \sigma = \max_b \sqrt{\text{Var}_\sigma D_{bd}}$$

(4.19)

**DT-CWT magnitude and maximum (dcmag, dcmx)**

Among the many extensions of wavelet analysis is the dual-tree complex wavelet transform (DT-CWT), which decomposes a signal onto a complex, overcomplete basis in order to achieve superior directional tuning and near translation invariance [104]. In the context of image texture modeling, the former property is very important since
maximal response is attained when the filter is perpendicular to the local image structure. In contrast to the standard discrete wavelet transform (DWT), which extracts information at $0^\circ$, $45^\circ$, and $90^\circ$ orientations, the DT-CWT extracts information in six orientations: $\pm 15^\circ$, $\pm 75^\circ$, and $\pm 135^\circ$; compare the filter responses in Figure 4-7.

Previous applications of the DT-CWT to texture [28, 45, 55], have relied upon features such as the subband’s normalized L1 norm and energy. In this thesis, the information is localized in two ways, the maximum magnitude of the complex coefficient or the maximum of the real or imaginary portion of the coefficient. These quantities are both positive, and thus their mean is used as a statistic to characterize their distribution. For decomposition level $d$ and subband orientation $b$, these features are calculated as:

$$d_{\text{cmag}} d - \sigma = G_\sigma \circ \max_b |D_{bd}| \quad (4.20)$$

$$d_{\text{cmax}} d - \sigma = G_\sigma \circ \max_b \max(\text{real}(D_{bd}), \text{imag}(D_{bd})) \quad (4.21)$$

4.3.3 Other filter-based features (abar, gauss, log)

Wavelet based features are augmented with three of the four filter-bank features from the MR8 [120]: gauss, log, and bar. The fourth feature, an edge filter, is a high-pass filter whose passband is tunable via a scaling parameter. As this is redundant to the wavelet-based features, it is consequently unused in this thesis.

Gaussian filter (gauss)

The Gaussian filter $G_\sigma$ is a directionless lowpass filter whose passband is modulated by scale $\sigma$. As a lowpass filter, it represents a local estimate of mean intensity. Coefficients displaced from the central tap by $(i, j)$ pixels are proportional to:

$$G_\sigma (i, j) = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{i^2 + j^2}{2\sigma^2}\right] \quad (4.22)$$
To ensure zero DC gain regardless of filter support size, coefficients are normalized so that $\sum_{i,j} G_\sigma (i,j) = 1$. For image $Y$, the Gaussian feature is thus:

$$\text{gau}ss_\sigma = G_\sigma \circ Y \quad \text{(4.23)}$$

**Laplacian-of-Gaussian filter (log)**

The Laplacian-of-Gaussian (LoG) filter is effectively a directionless derivative with scale $\sigma$. Coefficients are determined as:

$$H_{\text{LoG}_\sigma} (i,j) = \frac{\partial^2}{\partial i^2} G_\sigma + \frac{\partial^2}{\partial j^2} G_\sigma = \frac{i^2 + j^2 - 2\sigma^2}{2\pi\sigma^6} \exp \left[ -\frac{i^2 + j^2}{2\sigma^2} \right] \quad \text{(4.24)}$$

To ensure uniform response regardless of filter support size, LoG filters are normalized using the same factor as a Gaussian filter with identical support. For image $Y$, the LoG feature is:

$$\text{log}_\sigma = H_{\text{LoG}_\sigma} \circ Y \quad \text{(4.25)}$$

**Bar filter (abar)**

An anisotropic filter which responds to bar-shaped structures along the horizontal at scale $\sigma$ may be formulated by convolving the second derivative of a univariate Gaussian kernel $g_\sigma$ oriented along the vertical with another univariate Gaussian $g_{\sigma_h}$ oriented along the horizontal. While $\sigma_v$ may be tuned independently, $\sigma_h = 3\sigma$ generally
provides good results. The coefficients are thus:

\[ H_{\text{bara}}, \omega (i, j) = g_{3\omega} (i) \cdot \frac{\partial^2}{\partial j^2} g_{\sigma} (j) \]

\[ = \frac{j^2 - \sigma^2}{6\pi \sigma^6} \exp \left[ -\frac{j^2 + 9j^2}{18\sigma^2} \right] \]

To ensure uniform response regardless of filter support size, bar filters are normalized so that their 1-norm has unit magnitude. The filter is displayed in Figure 4-8a.

If there is a bar-structure present in the image, the maximal response to a bar filter of the same scale will occur if both the filter and the image structure have identical orientation. Unlike the Gaussian and Laplacian-of-Gaussian filters, the bar filter above is anisotropic and responds most strongly to horizontally oriented structures, though this orientation may be changed by rotating the \((i, j)\) coordinate system. In order to recover the feature response which most nearly corresponds to the image's local orientation, multiple iterations of \(H_{\text{bara}}\), rotated in even increments, are applied to the image and the largest absolute magnitude responds is retained:

\[ \text{abaro}(x) = \arg \max_{\omega} |H_{\text{bara}}, \omega \circ Y(x)| \]

4.3.4 Entropy-based features (hdiff, hloc)

Wavelet-based texture methods assess change in image intensity by measuring the occurrence of edges at various scales. Another measure of variation is the entropy in gray level intensity in a given neighborhood. The key insight behind localized entropy measurement is that the probability of an image taking on a certain intensity level in a neighborhood of scale \(\sigma\) may be estimated by convolving a Gaussian kernel \(G_{\sigma}\) with an appropriate indicator function.

For image \(Y\), define an indicator function \(\delta(Y(x) = i)\), which is one if pixel \(x\) takes on intensity level \(i\):

\[ \delta(Y(x) = i) = \begin{cases} 1 & Y(x) = i \\ 0 & Y(x) \neq i \end{cases} \]

Hence the name, abaro for absolute value of the bar filter response with scale \(\sigma\).
By (4.16):
\[
\hat{E}_\sigma [\delta(Y(x) = i)] = G_\sigma \circ \delta(Y(x) = i)
\]  
(4.30)

Since the indicator function is either zero or one, the localized expectation is equivalent to the localized probability that \(Y(x) = i\), and thus:
\[
\hat{P}_\sigma (Y(x) = i) = G_\sigma \circ \delta(Y(x) = i)
\]  
(4.31)

**Local entropy estimate (hloc)**

Given (4.31), it follows immediately that the entropy in a neighborhood may be estimated by summing over the 256 possible intensity levels in an 8-bit image:
\[
hloc \sigma = - \sum_{i=0}^{255} (G_\sigma \circ \delta(Y(x) = i)) \cdot \log [G_\sigma \circ \delta(Y(x) = i)]
\]  
(4.32)

While evaluating this term consists of 256 separate filtering operations, texture features need only be calculated once for an image. For reasonably sized neighborhoods, the computational burden is not excessive, and binning may be used to simplify computation at larger scales.

**Difference in local entropy estimates (hdiff)**

Each of the methods above computes image statistics according to a weighted average within an isotropic window. In areas bordering more than one texture region, this leads to averaging over multiple sets of texture properties with anomalous results. This thesis introduces a feature which detects border regions by explicitly detecting oriented changes in local entropy.

This is accomplished by constructing an oriented window by convolving the positive components of the first derivative of a univariate Gaussian kernel \(g_{\sigma_i}\), oriented along the horizontal with another univariate Gaussian \(g_\sigma\) oriented along the vertical. As with the bar filter, in general \(\sigma_i = 3\sigma\).
window are:

\[
H_{\text{directed} \sigma}(i, j) = g_\sigma(j) \cdot \frac{\partial}{\partial i} g_\sigma(i) \quad \text{for } i \leq 0
\quad \text{(4.33)}
\]
\[
= \begin{cases} 
  -\frac{i}{6\pi\sigma^2} \exp \left[ -\frac{i^2 + j^2}{18\sigma^2} \right] & i \leq 0 \\
  0 & i > 0
\end{cases}
\quad \text{(4.34)}
\]

To ensure unity DC gain, the windowing filter is normalized so that its L1 norm has unit magnitude. Like (4.16), this window estimates a localized mean; unlike the previous window, however, this result is anisotropic and estimates the mean of a region to the left of the central pixel. See Figure 4-8b for a visual representation of this filter. As with the bar filter, the directionality may be changed by rotating the \((i, j)\) coordinate frame.

Given this shaped measurement window, directed difference in entropy at orientation \(\omega\) is simply the difference between the entropy estimate under kernels \(H_{\text{directed} \sigma, \omega}\) and \(H_{\text{directed} \sigma, \omega + 180^\circ}\):

\[
\hat{H}_{\text{directed} \sigma, \omega}(x) = \sum_{i=0}^{255} (H_{\text{directed} \sigma, \omega} \circ \delta(Y(x) = i)) \cdot \log [H_{\text{directed} \sigma, \omega} \circ \delta(Y(x) = i)]
\]

\[
\text{hdiff}_\sigma = \max_{\omega} \left| \hat{H}_{\text{directed} \sigma, \omega} - \hat{H}_{\text{directed} \sigma, \omega + 180^\circ} \right|
\quad \text{(4.35)}
\]

Note that hdiff\(_\sigma\) is a partial exception to the localization schema formulated at the beginning of section 4.3.1, as the order of dimensionality reduction (step 2) and expectation estimation (step 3) is reversed.

### 4.3.5 Compressive sensing features (cs)

Recent efforts to apply compressive sensing theory to texture classification [62] may be similarly localized via this framework. Compressive sensing texture modeling begins by re-ordering the \(n \times n\) patch around a pixel \(x\) into a single \(n^2 \times 1\) vector \(f(x)\) which is then projected into an \(m\) dimensional observation \(w(x)\) using a \(m \times n^2\) sensing kernel.
\( \Phi \) as \( w(x) = \Phi f(x) \). In accordance with compressive sensing theory, this sensing kernel \( \Phi \) is comprised of independent Gaussian white noise with unit variance. This methodology may be formulated as a frequency domain texture method by column re-ordering the rows of \( \Phi \) into a bank of \( m \) filters of size \( n \times n \).

Specifically, consider \( H_{CSn-1} \), an \( n \times n \) matrix of independent, unit variance Gaussian white noise. Compressive sensing texture features may be generated as:

\[
\text{c}sn-1 = H_{CSn-1} \circ Y
\]  

(4.37)

This single compressive sensing feature is equivalent to the first entry in \( w(x) \). By generating \( m - 1 \) additional filters \( H_{CSn-i} \) (with \( i \) being an index), the entire vector \( w(x) \) may be populated.

4.3.6 Comparative summary

For a visual comparison, please see Figure 4-9, which computes each of the features described in this section for the same pluripotent stem cell colony image. Note the similarity among the wavelet family of features (wave3-16, dcmag3-16, and dcmax3-16); choosing the best member of this family for stem cell image segmentation will be performed in section 5.2.1. Further insights which will be confirmed in this section include the apparent lack of power of the Gaussian and cs features for stem cell segmentation, and the utility of the hdiff feature in locating colony borders.

While some features, such as the wavelet-based family, are powerful for a variety of applications, it should be noted that the utility of a texture feature is often highly application-specific. A feature useful for segmenting stem cell colonies (section 5.2.1) might be of little use in segmenting an image of a zebra (section 5.4) and vice versa. Experimentation and engineering judgment are necessary to find a good feature set for a particular problem.
Figure 4-9: *Texture feature comparison.* The texture features discussed in this section are each evaluated for the pluripotent colony image in (a).
4.4 Summary

This chapter began by motivating image texture as a means of classification and then briefly recounted the history of image texture modeling, grouping algorithms according to whether they quantify distance in the spatial or frequency. Next, wavelet-based texture models were examined and validated for use in classifying stem cells. In the process, both adaptive and hierarchical windowing methods were formulated for textural analysis of very small and very large image regions. The chapter concluded by providing a procedure for localizing texture features for use in image segmentation and demonstrating several candidate features for this purpose.
Chapter 5

Texture-based Segmentation using Bayesian Level Sets

In chapter 3, a Bayesian Level Set algorithm was developed for image segmentation, and chapter 4 demonstrated that image texture could reliably classify stem cell colonies. Motivated by this application, chapter 5 fuses these methods into a texture-based Bayesian Level Set algorithm for simultaneous segmentation and classification. This chapter also presents a methodology for training the texture-based BLS which is demonstrated on stem cell colony images.

While the texture-based BLS algorithm and the training procedure were developed for use in segmenting stem cell colony images, they are not specific to this application. Accordingly, the chapter concludes by demonstrating the algorithm on two images of zebras.

5.1 The texture-based Bayesian Level Set algorithm

To adapt the BLS developed in chapter 3 to texture-based segmentation, the mixture model is augmented by adding a third texton stage between the class assignment and the observation. Other modifications to support texture segmentation include the addition of a mixture prior at the class level in order to suppress outliers and conjugate data priors in order to prevent overfitting.
5.1.1 Three-stage Markov texture model

The version of the BLS developed in chapter 3 explicitly assumes that all observations $Y$ are generated via a finite mixture model as described in section 2.3.2 and illustrated in Figure 2-2. For each pixel $x$, a class $Z(x)$ is first selected from some set $C$. Conditioned on that class assignment, an observation $Y(x)$ is then generated using a class-specific parameter set. Intuitively, it is inherently implausible to apply this model to texture, which human vision recognizes as a global pattern consisting of elements which might individually appear quite different. By way of example, consider the coat of the zebra in Figure 5-2, a simple texture consisting of a series of alternating black and white stripes. While recognizably a single texture, the local appearance of the zebra’s coat varies dramatically.

The texture-based BLS accounts for this by introducing an additional texton stage between the class assignment and the observation in order to better represent the texture’s varying local characteristics. This process is compared to the two-stage finite mixture model in Figure 5-1. At pixel $x$, the first event assigns to the unknown class $Z(x)$ a label from from some finite set $C$. A texton $T(x)$ is next selected from some finite set $T$ according to a multinomial distribution with parameters specific to class $Z(x)$. Lastly, the observation $Y(x)$ is generated according to parameters specific to the texton $T(x)$. As observation parameters $\theta_T$ are determined by the texton, the
Figure 5-2: A zebra’s stripes are an example of texture. The zebra’s stripes form a distinct texture pattern consisting of two noticeably distinct elements: the black and white stripes.

texton thus characterizes what the texture locally “looks like.” Furthermore, the three stage model is Markovian insofar as data $Y$ is independent of class $Z$ when conditioned on texton $T$.

For a demonstration of this texture model, see Figure 5-3. The blue dots in Figure 5-3b show subsampled data points from the pluripotent texture shown in Figure 5-3a. Texture data is modeled as Gaussian, so that $P(Y; \theta_T) = N(Y; \mu_T, J_T)$, and the resulting textons are shown via the red ellipses which indicate the half-peak contour of the Gaussian. Mixture ratios $\pi_{T|Z}$ are 0.4148 for the texton with solid contour and 0.5852 for the dashed texton. Figure 5-3c shows the conditional probability density $P(Y, Z)$ corresponding to this texton data model obtained by marginalizing with respect to the textons.

The crucial advantage of the three-stage procedure is that it allows different density models to be imposed on the selection of class $Z$ and texton $T$. As discussed in section 3.5, the geometric prior promotes local classification similarity in order to reduce misclassification, reject noise, discourage topological fracture, and encourage the formation of smooth, pleasing segmenting contours. These advantage may be realized in texture-based segmentation by imposing the same types of geometric priors at the class level ($Z$) when solving the three-stage Markov model. As the zebra example suggests, however, it may be harmful to impose geometric and smoothing priors at

---

1The four-dimensional texture data and the associated textons are derived as discussed in section 5.2. Only two dimensions are shown here in the interests of legibility.
Figure 5-3: A demonstration of the Markov texture model. (a) pluripotent stem cells are a texture of interest; (b) blue dots represent subsampled data from the texture in (a), and the two red contours indicate the half-peak contour of the two textons used to model this data; texton mixture ratios are in the legend; (c) the conditional probability density corresponding to this texton model.
Figure 5-4: The most likely texton varies within the class. (a) A pluripotent test image; the texture-based BLS algorithm classified blue regions as pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony; smoothing on the class prior results in the rounded segmenting contour and smooth class transition shown here; (b) The most likely texton according to the texture-based BLS, textons 1 and 2 are associated with the pluripotent class, 3-5 with differentiated, and 6-8 with media; smoothing the texton prior might have interfered with classification, such as by suppressing the striations and protrusions of textons 7 and 8 (dark red) within texton 6 (orange).

the texton level \( T \), since textures are defined by a combination of local appearances which vary in close proximity; black stripes are immediately adjacent to the white stripes. Within a given class or region, it is therefore best to spatially dissociate the selection of textons by imposing only a mixture ratio prior such as (3.36). This is illustrated in Figure 5-4.

**Mathematical interpretation**

This three-stage Markov process is equivalent to modeling the texture of a class \( Z \) according to a set of weighted basis functions. Individual bases are the distributions by which \( Y \) is generated from \( T \), and weights are the multinomial parameters describing the probability of \( T \) conditioned \( Z \).

If significant training data exists, as is the case with many laboratory and clinical applications, including the segmentation of stem cell colony images, the texton approach is superior to nonparametric density estimators such as the Parzen approach.
used in [53]. Prior training may be leveraged to identify a more sparse set of basis vectors than might be allowed with a Parzen estimator while still allowing shape and weight (i.e. the data modeling parameters and texton mixture ratios) of the basis functions to be refined to better model the test data. (See section 5.1.4.) While this has obvious performance benefits in the form of reduced computational cost, the smaller number of basis vectors also serves to help discourage the texture model of a given class from overtraining or diverging too greatly from the textures identified in the training data insofar as a fixed number of basis vectors shapes the family of distributions which may be modeled.

5.1.2 Adapting the three-stage Markov model in the BLS framework

To segment images consisting of data generated by this three-stage model in the BLS framework, begin by posing the complete log-likelihood of the data in terms of the indicator function defined in (2.26):

$$L(Y, \Theta) = \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} \delta(Z(x) = c \cap T(x) = t) \log \pi_c(x) \pi_{tlc}(x) P(Y(x); \theta_t)$$  \hspace{1cm} \text{(5.1)}$$

Here, \(\pi_c\) is the prior probability of class \(c\), \(\pi_{tlc}\) is the prior probability of texton \(t\) conditioned on class \(c\), and \(P(Y(x); \theta_t)\) is the probability density of the observation \(Y(x)\) assuming texton-specific parameters \(\theta_t\). The model parameters \(\Theta\) are the union of all \(\pi_c\), all \(\pi_{tlc}\), and all \(\theta_t\). For some candidate distribution \(Q\) which assigns \(P(Z(x) = c \mid Y(x)) = q_{c|Y}(x)\) and \(P(T(x) = t \mid Y(x), Z(x) = c) = q_{tl|c}(x)\), the expected log-likelihood becomes:

$$E_Q[L(Y, \Theta)] = \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} q_{c|Y}(x) \log \pi_c(x)$$

$$+ \sum_{x \in \Omega} \sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} q_{c|Y}(x) q_{t|Y,c}(x) \left[ \log \pi_{tl|c}(x) + \log P(Y(x); \theta_t) \right]$$  \hspace{1cm} \text{(5.2)}$$

A BLS for this expected log-likelihood function may be derived by implementing
the procedure described in section 3.6:

1. Pose the expected log-likelihood function as a FMM-EM with a mixture prior ($\pi_z$ defined in (2.30)).
2. Derive the minimizing conditions for the FMM-EM.
3. Replace the mixture prior $\pi_z$ with the desired geometric prior $q_z$ subject to the conditions described in section 3.4.

For a corresponding FMM-EM derivation using a Normal data model, please see appendix C.3. The resulting soft decision updates are thus:

\begin{equation}
q_{z|y}^{k+1}(x) = \frac{\pi_z^k(x) \sum_{t \in T} \pi_{i|z}^k N(Y(x); \mu_t^k, J_t^k)}{\sum_{c \in C} \pi_c^k(x) \sum_{t \in T} \pi_{i|c}^k N(Y(x); \mu_t^k, J_t^k)} \tag{5.3}
\end{equation}

\begin{equation}
q_{T|y,z}^{k+1}(x) = \frac{\pi_{T|z}^k N(Y(x); \mu_T^k, J_T^k)}{\sum_{i \in T} \pi_{i|z}^k N(Y(x); \mu_i^k, J_i^k)} \tag{5.4}
\end{equation}

Note that (5.3) is simply a marginalization with respect to the choice of texton $T$.

Parameter updates are:

\begin{equation}
\theta_T^k = \arg \max_{\theta} \sum_{x \in \Omega} q_{z|y}^k(x) q_{T|y,z}^k(x) \log P(Y(x); \theta) \tag{5.5}
\end{equation}

\begin{equation}
\pi_z^k(x) = S(q_{z|y}^k(x)) \tag{5.6}
\end{equation}

\begin{equation}
\pi_{T|z}^k = \frac{\sum_{x \in \Omega} q_{z|y}^k(x) q_{T|y,z}^k(x)}{\sum_{x \in \Omega} q_{z|y}^k(x)} \tag{5.7}
\end{equation}

For a Normal data model (implying that $\theta_t = \{\mu_t, J_t\}$), (5.5) becomes:

\begin{equation}
\mu_T^k = \frac{\sum_{x \in \Omega} \sum_{c \in C} q_{c|y}^k(x) q_{T|y,c}^k(x) Y(x)}{\sum_{x \in \Omega} \sum_{c \in C} q_{c|y}^k(x) q_{T|c,y}^k(x)} \tag{5.8}
\end{equation}

\begin{equation}
J_T^k = \frac{\sum_{x \in \Omega} \sum_{c \in C} q_{c|y}^k(x) q_{T|y,c}^k(x) (Y(x) - \mu_T^k)(Y(x) - \mu_T^k)^T}{\sum_{x \in \Omega} \sum_{c \in C} q_{c|y}^k(x) q_{T|c,y}^k(x)} \tag{5.9}
\end{equation}

**5.1.3 Geometric prior: combine smoothing and mixture**

While any of the geometric and smoothing priors in section 3.4 is potentially a valid choice for texture-based segmentation, the results in chapter 6 were obtained using
a hybrid geometric prior consisting of the curvature flow smoothing prior \( q_{s,z} \) from (3.32)-(3.34) and the mixture prior \( q_{m,z} \) from (3.36) combined via the Naive Bayesian approach in (3.28).

The smoothing prior is, of course, one of the chief advantages of the BLS, and the curvature flow prior was chosen as it better preserves segmentation structure than Gaussian smoothing and is more stable than first-order curvature smoothing. This latter quality is critical in handling the initial rapid changes in segmentation due to the many small disconnected regions sometimes formed by the automated initial conditions described in section 5.3.

The mixture prior serves to suppress misclassification of outlier regions by increasing the probability of large classes and penalizing the likelihood of small ones, which is crucial for correct segmentation of stem cell colonies. As noted in section 4.3, the distribution of features near a textural interface is a superposition of the distributions characterizing the individual textures, and statistics calculated in this region may more closely resemble those of some other class. In the present case, textural statistics near the interface between pluripotent cells and the growth media notably resemble those of differentiated cells. Likewise, dead cells and debris within differentiated colonies often have textural characteristics similar to those of pluripotent regions. The mixture prior corrects for potential confusion as shown in Figure 5-5.

The combined geometric prior for segmenting and classifying stem cell colony images is:

\[
q^k_{Z}(x) = q^k_{s,z}(x) \cdot q^k_{m,z}(x) \\
q^k_{s,z}(x) = \left(1 + e^{-L^k_{Z}(1,x)} \right)^{-1} \\
L^k_{Z}(0, x) = \log \frac{q^k_{Z}(x)}{1 - q^k_{Z}(x)} \\
\dot{L}^k_{Z}(t, x) = \beta \kappa (L^k_{Z}(t, x)) \\
q^k_{m,z}(x) = \frac{1}{|\Omega|} \sum_{x \in \Omega} q^k_{Z}(x)
\]
Figure 5-5: *The mixture prior suppresses outlier classes to prevent misclassification.* (a) a pluripotent test image, segmented via the texture-based BLS algorithm described in this chapter; (b) the same image when the mixture prior has been omitted; the mixture prior is critical at suppressing misclassification of outliers, including areas near the pluripotent-media interface which have similar statistics to the differentiated texture; blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.

5.1.4 Conjugate data priors: Normal-inverse-Wishart and Dirichlet

Overtraining is a common problem in machine learning and is especially acute in texture-based simultaneous segmentation and classification, where textons may be defined for classes that do not even exist within a particular test image. In these cases, textons from absent classes run the risk of being refined to match outliers within a test data set which more properly belong to another class, resulting in misclassification. To prevent this phenomenon, conjugate priors are imposed upon the distributions of the texton modeling parameters.

Figure 5-6 illustrates how a data prior is necessary for proper segmentation of the pluripotent colony. In the absence of data priors, the differentiated textons train to outliers along the border of the pluripotent colony; since there are no differentiated regions present in this image, this is a misclassification. Observe how the green and
Figure 5-6: Data priors prevent misclassification due to overtraining. (a) a pluripotent test image, segmented via the texture-based BLS algorithm described in this chapter with a data prior; the area colored blue is classified pluripotent, green differentiated, and red media; (b) the same image, segmented without data priors; note the extensive region along the colony border misclassified as differentiated; (c) final locations of the differentiated textons in (a); dots are sample data from the first two of the four transformed features, calculated as in section 5.2.2, with red for media and blue for pluripotent – there is no differentiated data present in this image; black ellipses correspond to the final half-peak contours of the distributions corresponding to the three differentiated textons; texton mixture ratios are given in the figure legend; (d) final locations of the differentiated textons in (b); the green and purple arrows in (c) and (d) pick out the same textons in both images and illustrate how the lack of data priors leads to overtraining; while in (c) the texton indicated in green is completely unaligned to the data, the lack of data prior in (d) allows the texton to move and conform to a space populated by pluripotent data; likewise, the texton indicated in purple overtrains by assuming increased importance, with its mixture ratio increasing by 106% when the data prior is omitted.
purple arrows pick out the same texton in Figure 5-6c-d. With the data prior, the
texton indicated in green is noticeably offset from the image data; omitting the data
prior allows it to move and conform to the pluripotent data. Likewise, omitting the
data prior allows the texton indicated in purple to overtrain by assuming increased
importance, and its mixture ratio increases by 106% from 0.1896 to 0.3906.

A Dirichlet conjugate prior $D(\cdot)$ with parameters $\eta_Z$ is imposed on the multinomial
distribution whose parameters are the conditional mixture ratios $\pi_{T|Z}$. The parameter
set $\eta_Z$ consists of one $\eta_{t|Z}$ for each $t \in T$, which in turn may be estimated from the
unnormalized empirical mixture ratios observed in the training set. Specifically, if
the training set consists of $n$ instances of $T$ in class $Z$, then $\eta_{T|Z} - 1 = n$.

$$
\pi_{T|Z} \sim D(\eta_Z)
$$

$$
= P(\pi_{T|Z}; \eta_Z) = \frac{1}{B(\eta)} \prod_{t \in T} \pi_{t|Z}^{\eta_{t|Z}-1}
$$

Here $B(\cdot)$ is the multinomial Beta function.

Likewise, a Normal-Inverse-Wishart prior $NIW(\cdot)$ with parameters $\nu_T$, $\lambda_T$, $\psi_T$,
and $\tau_T$ is imposed on Normal data model whose parameters are the texton centers
$\mu_T$ and covariances $J_T$. $\nu_T$ is the empirical mean of a training set with $\lambda_T$ samples,
while $\psi_T$ is the empirical covariance from a training set with $\tau_T$ entries. As the
same training set is generally used to estimate both $\nu_T$ and $\psi_T$, generally $\lambda_T = \tau_T$.
The $NIW$ is itself a product of two distributions, a normal distribution on $\mu_T$ with
mean $\nu_T$ and covariance $\frac{1}{\lambda_T} J_T$, and an inverse Wishart distribution on $J_T$ with inverse
scaling matrix $\tau_T\psi_T$ and $\tau_T$ degrees of freedom. Thus:

\[
(\mu_T, J_T) \sim NIW(\nu_T, \lambda_T, \psi_T, \tau_T) \\
\mu_T \sim N\left(\nu_T, \frac{1}{\lambda_T} J_T\right) \\
\Rightarrow P(\mu_T | \nu_T, \lambda_T, J_T) = \frac{\lambda_T^{1/2}}{(2\pi)^{-d/2} |J_T|^{1/2}} e^{-\frac{\lambda_T}{2} (\mu_T - \nu_T)^T J_T^{-1} (\mu_T - \nu_T)}
\]

\[
J_T \sim W^{-1}(\tau_T \psi_T, \tau_T) \\
\Rightarrow P(J_T | \psi_T, \tau_t) = \frac{\tau_T^{\tau_T/2} |\psi_T|^{\tau_T/2}}{2^{\tau_T d/2} \Gamma_d\left(\frac{\tau_T}{2}\right)} |J_T|^{-(\tau_t + d + 1)/2} e^{-\frac{\tau_T}{2} \psi_T (\psi_T)^{-1}}
\]

Here, $d$ is the degree of the data $Y$.

Derivations for a two stage FMM-EM with these priors are given in appendix C.2; derivations for the three stage algorithm are exactly analogous. The result of imposing these priors is that parameter updates become:

\[
\eta_{T|Z}^k = \frac{\sum_{x \in \Omega} q_{Z|Y}(x) q_{T|Y,Z}^k(x) - 1 + \eta_{T|Z}}{\sum_{x \in \Omega} q_{Z|Y}(x) + \sum_{t \in T} \eta_{t|Z} - |T|} \\
\mu_T^k = \frac{\sum_{x \in \Omega} \sum_{c \in C} q_{Z|Y}^k(x) q_{T|Y,c}^k(x) Y(x) + \lambda_T \nu_T}{\sum_{x \in \Omega} \sum_{c \in C} q_{Z|Y}^k(x) q_{T|c,Y}^k(x) + \lambda_T}
\]

\[
J_T^k = \left[\frac{\sum_{x \in \Omega} \sum_{c \in C} q_{Z|Y}^k(x) q_{T|c,Y}^k(x) (Y(x) - \mu_T^k) (Y(x) - \mu_T^k)^T + \tau_T \psi_T + \lambda_T (\mu_T^k - \nu_T) (\mu_T^k - \nu_T)^T}{\sum_{x \in \Omega} \sum_{c \in C} q_{Z|Y}^k(x) q_{T|c,Y}^k(x) + \tau_T + d + 2}\right]
\]

To interpret these equations, note that the new updates for mixture priors $\pi_{T|Z}$ and texton centers $\mu_T$ are simply weighted averages of the empirical estimate under the current $Q$ and the prior, which is itself an estimate taken from the training data. (The quantity $\sum \eta_{t|Z} - |T|$ in (5.19) is simply the total number of samples used to obtain the $\eta_{T|Z}$.) The new estimate of the covariance matrices $J_T$ are a weighted combination of the empirical estimate (first term), the prior (second term and likewise an estimate based on the training data), and a term based on the variance between the empirical and prior estimates of the texton center.
5.1.5 The texture-based Bayesian Level Set algorithm

The texture-based Bayesian level set (BLS) algorithm used to simultaneously segment and classify stem cell colony images in chapter 6 is implemented via the two-stage process for which pseudocode is provided in Table 5.1.

Pseudocode:

```
converged = false
Ω = {domain of image}
C = {class list for the BLS}
T = {texton list for the BLS}
e = tolerance

while ~converged
    % update soft decisions
    for {c ∈ C}
        calculate q_c^k|y according to (5.3)
        for {t ∈ T}
            calculate q_t^k|y,c according to (5.4)
    % update parameter estimates
    for {c ∈ C}
        calculate q_c^{k+1} according to (5.10)-(5.14)
        for {t ∈ T}
            calculate π_t|c according to (5.19)
    for {t ∈ T}
        calculate μ_t and J_t according to (5.20), (5.21)
    % Check for convergence
    if ∥q_c^k|y - q_c^{k-1}|y∥ < ε for all x ∈ Ω, c ∈ C
    converged = true
```

Table 5.1: Pseudocode for a texture-based Bayesian Level Set (BLS) algorithm.

Update soft decision

First, update the soft decisions, which consist of both q_zy, the likelihood of the class conditioned upon the data, and q_t|y,z, the probability of a texton conditioned upon both the data and a particular class assignment, as in (5.3)-(5.4).
Update parameters

Second, conditioned upon the soft decisions and the data prior parameters \( \{ \eta, \nu, \lambda, \psi, \tau \} \) identified during training, update the entire parameter set \( \Theta \). This set consists of both data modeling terms:

- Texton mixture priors \( \pi_{T|z} \), updated as (5.19),
- Texton centers \( \mu_T \), updated as (5.20), and
- Texton covariances \( J_T \), updated as (5.21).

and also:

- Hybrid geometric prior \( q_z \), consisting of both mixture and curvature flow smoothing priors and updated as (5.10)-(5.14).

5.2 Training

Training data is easily available for many of the desired laboratory and clinical applications of automated image segmentation and classification, including stem cell colonies. In these cases, automated segmentation greatly benefits from prior training in order to identify:

- the sparsest combination of features which classify acceptably,
- the optimal number of textons for each class of interest, and
- the parameters for the data priors described in section 5.1.4.

The remainder of this section describes the training process used to prepare for the automated simultaneous segmentation and classification of stem cell colony images described in chapter 6.

For the purposes of training, three pluripotent and three differentiated images were selected, and their textural regions were manually identified. In order to partly preserve these images for future testing, only half of each image was used for training. The training images are shown in Figure 5-7.
Figure 5-7: Pluripotent and differentiated training images. (a)-(c) pluripotent training images; (d)-(f) differentiated training images; the red portion of the image was reserved from training for future testing, and an asterisk next to the letter code denotes a training image rather than the test images shown in other figures.
5.2.1 Feature selection

A great many candidate image texture features were identified in section 4.3; obviously using them all in automated stem cell segmentation is infeasible. Selecting the best features from this list was performed by pruning away features according to the heuristics below. Since texture feature utility is often highly application-specific, the best set of features is likely to differ for another classification problem.

After computing each of the candidate features at varying neighborhood sizes on the training data, the first step was to eliminate two of the three wavelet feature families from consideration (i.e. dcmag, dcmax, and wave). Each of these three families is an attempt to characterize the maximal response in an oriented frequency subband, and so multiple features from within this family would be redundant. The dcmax family was thus selected as it was generally found to be the most informative in terms of mutual information $MI(Y, Z)$ from the feature to the class:

$$MI(Y, Z) = \int_{y \in Y} \sum_{z \in C} P(Y, Z) \log \frac{P(Y, Z)}{P(Y) P(Z)}$$

The second step was to identify the optimal neighborhood size for each remaining candidate feature, also according to its mutual information.

In the third step, features with $MI(Y, Z) < 0.3$ were pruned as uninformative as shown in Figure 5-8. This threshold was inspired by the compressive sensing (cs) features as described below. The single exception to this rule is the hdiff8 feature, which is highly informative at the critical interface between colonies and their growth media. As previously discussed, neighborhoods straddling textural transitions present a mixture of data drawn from the different density functions characterizing the two textures; the features of these neighborhoods are perforce inherently somewhat corrupted and prone to misclassification. The value of the difference in entropies feature (hdiff) lies in alerting the algorithm to these transitions, as illustrated in Figure 5-9.

The seven selected features are thus abar4, dcmag1-16, dcmag2-16, dcmag3-16, dcmag4-16, hdiff8, and hloc8.
Figure 5-8: Mutual information-guided feature selection. (a) mutual information between the candidate features of interest (abar4, dcmax1-16, dcmax2-16, dcmax4-16, gauss16, hdiff8, hlocal8, log2) and the manually identified classes for the training images in Figure 5-7, the red threshold at 0.3 indicates the utility threshold inspired by the CS features; (b) mutual information between the candidate features of interest, ordered as in (a); note that the relative redundancy of the selected features, which occasions feature reduction in section 5.2.2.

Figure 5-9: The hdiff texture feature is critical for successful segmentation near colony boundaries. (a) pluripotent segmentation when hdiff8 is included in the set of texture features; (b) the same test without hdiff8; neighborhoods on the interface between pluripotent and media regions contain a mixture of data from these two texture distributions with aggregate statistics prone to misclassification as differentiated; (c) hdiff8 calculated for this colony image with the blue contour in (a) redrawn in black, note how hdiff8 identifies the border region immediately exterior to the black contour, which is prone to misclassification; blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Further insight on texture feature selection

As noted, \( \text{dcm} \) is the most informative of the three wavelet-based features (\( \text{dcmag} \), \( \text{dcm} \), and wave); results such as those in Figure 5-10, which shows the feature-to-class MI for a single pluripotent and differentiated training colony, are common. Specifically, \( \text{dcm} \) is slightly more informative than \( \text{dcmag} \), and both are about as informative as the wavelet standard deviation calculated for a neighborhood of twice the size.

Examination of Figure 5-10 conveys further insight; specifically, larger neighborhoods are generally more informative than smaller ones. (e.g. \( \text{dcm} \) is more informative than \( \text{dcmag} \), which is more informative than \( \text{dcm} \).) This should not be surprising, as the larger the neighborhood, the more effective samples it contains and the better the feature estimate. Precision, however, places a soft limit on the maximum neighborhood size. As described in section 4.3, estimation of a neighborhood’s statistics are performed via Gaussian filtering, a blurring operation. Consequently, increasing the neighborhood size inherently involves a tradeoff between the accuracy of feature statistics and the precision with which those statistics are located in space. This tradeoff is illustrated in Figure 5-11, where the feature blurring in the latter subfigure was judged to have caused too much divergence from the user-
Figure 5-11: Precision limits neighborhood scale. (a) dcmax3-16 for the colony in Figure 5-7a, the black contour is the manually identified segmentation; (b) dcmax3-32 for the same colony; at this scale, feature blurring associated with the neighborhood size was judged too great to support a precise segmentation.

specified ground truth segmentation. A similar process was carried out for each family of the candidate features.

Compressive sensing as a soft threshold on feature usefulness

Compressive sensing features (cs) can serve as a soft threshold for feature utility. In [62],2 it was suggested that the number of features be one third the original patch size. Representing an \( m \times n \) patch with \( \frac{1}{3} mn \) features is only a two-thirds reduction in the feature size and suggests that the underlying texture may not be sparse in the image domain. Furthermore, as Figure 5-12 illustrates, the predictive power of individual compressive sensing features is low, with a maximum observed mutual information \( MI(Y, Z) \) between feature and class of 0.2155 for stem cell images.

The low information content of compressive sensing texture features and their method of construction suggests, however, that they might serve as a soft threshold on the usefulness of texture features. Intuitively, a good feature ought to be more informative than one resulting from a sensing kernel formed via random chance. In-

\(^{2}\)See section 4.3.5 for a review.
The predictive power of individual compressive sensing texture features is low. MI from feature to class for compressive sensing features with varying support size; data was comprised of a single pluripotent and differentiated training image; maximum observed MI is 0.2155.

spired by this consideration, uninformative features were screened out using a soft lower bound of $MI(Y, Z) = 0.3$, or approximately 40% greater than the maximum observed compressive sensing $MI(Y, Z)$.

### 5.2.2 Feature reduction via Principle Component Analysis (PCA)

In order to increase computational efficiency and decrease redundancy, Principal Component Analysis (PCA) [108] is used to reduce the seven selected features to four. To account for magnitude variation, the seven raw features were normalized to have unit standard deviation prior to applying PCA. As a good feature should be both informative (implying high feature-to-class MI) and possess enough variation to easily estimate its density (implying a high eigenvalue), the product of the mutual information $MI$ and eigenvalue $V$ was used as a score of feature efficacy, as shown in Figure 5-13a.

The first four features were selected, resulting in the training data shown in Figure 5-14. Inspection reveals the transformed data to be largely but not completely separable; the overlap in the data creates an ambiguity in classification which must be resolved by the ability of the BLS to impose local classification similarity and discourage outlier classes as discussed in section 5.1.3.
(a) $V \cdot MI$

(b) principal components

(c) MI feature-feature, original data

(d) MI feature-feature, transformed data

Figure 5-13: PCA feature reduction. (a) the product of the MI from feature to class and the eigenvalue $V$ is the utility criterion for PCA feature reduction since transformed features should be informative (high MI) and have large variance $V$ in order to allow for accurate estimation; (b) the principal components of the PCA transformation reveal the informative modes within the original data, in order: response in any feature save $hdiff8$, response in $hdiff8$, low frequency response ($dcmx4_{-16}$ with high local entropy) and bar-shaped structures, and low frequency response ($dcmx3_{-16}$ and $dcmx4_{-16}$ with high local entropy) without bar-shaped structures; (c) MI between the original features, illustrating their high degree of redundancy; (d) MI between the transformed features shows redundancy greatly reduced.
Figure 5-14: Data reduced via PCA. Blue indicates pluripotent, green differentiated, and red media. The transformed data is not entirely separable, as noticeable overlap remains.
Further insight on the reduced data

Further insight into these results may be obtained from Figures 5-13b-d. The directions of the most significant modes within the original data are illustrated by the eigenvectors shown in Figure 5-13b. The first mode is an almost even combination of all features save hdiff8; the most powerful criterion for distinguishing texture is simply the presence and magnitude of a response in any one of these features. By contrast, the second feature primarily responds to hdiff8 and is thus functionally a detector of the region immediately external to a colony. The third and fourth features both extract information at low frequencies, primarily via dmax4-16 and local entropy. The two features differ in that the former but not the latter also responds to the presence of bar-shaped structures via the abar4 response. By comparing the feature-to-feature mutual information of the original and reduced features in Figures 5-13c-d, it can be seen that the reduction process noticeably decreases the degree of redundancy in the transformed features.

5.2.3 Texton learning via GMM-EM clustering

Following reduction to four features via PCA, textons were learned for each texture class using the GMM-EM described in section 2.3. The cluster parameters identified in this process become the data priors used by section 5.1.4, and the optimum number of textons per class was discovered via direct testing. For the stem cell colony segmentation and classification problem, this was found to be two (2) pluripotent, three (3) differentiated, and three (3) media textons. The results are illustrated for the pluripotent data in Figure 5-15.

5.3 Initialization

As advertised in section 3.5, the Bayesian level set often performs acceptably even in the face of an extremely inexact automated initialization, and the texton-based BLS is no exception. The stem cell segmentation in chapter 6 was therefore initialized
Figure 5-15: Pluripotent textons derived via the GMM-EM. Blue dots are subsampled pluripotent data points, and the two red contours indicate the half-peak contour of the two textons identified via the GMM-EM; the legend indicates the two texton mixture ratios.
without any operator input. Class probabilities \( q_{Z|Y} \) were initialized in proportion to the mixture ratio of the nearest texton, while conditional texton probabilities \( q_{T|Y,Z} \) were initialized according to the data priors. For the texton centers \( \nu_T \), covariances \( \psi_T \), and mixture ratios \( \eta_{T|Z} \) identified in training, the initialization was:

\[
T = \arg \min_{t \in T} |Y(x) - \nu_t|
\]

\[
q^0_{Z|Y}(x) = \frac{\eta_{T|Z}}{\sum_{c \in C} \eta_{T|c}}
\]

\[
q^0_{T|Y,Z}(x) = \frac{\eta_{T|Z} N(Y(x) ; \nu_T, \psi_T)}{\sum_{t \in T} \eta_{T|Z} N(Y(x) ; \nu_t, \psi_t)}
\]

### 5.4 Case Study: Zebras

While the algorithm and training procedure described in this chapter were developed for use in simultaneous segmentation and classification of stem cell colony images, they are not specific to this application and are suitable for general use in texture-based segmentation. By way of example, consider the segmentation of an image containing a zebra standing on a grassy plain. The zebra texture, of course, is defined by its stripes while the grassy plain presents a grainy appearance. Training on the manually classified image in Figure 5-16a identified the abar1, abar2, abar4, dmax2-4, dmax3-4, and gauss4 features as most useful in distinguishing between the zebra...
and the plain. By application of PCA, these seven features were reduced to three (3), which were clustered into three textons apiece. Using this data prior, both the training image and a test zebra were segmented to within 98% accuracy as shown in Figure 5-16.

5.5 Summary

This chapter began by formulating a variant on the BLS that uses a three-stage Markov texture model, hybrid curvature-mixture geometric prior, and conjugate data priors in order to segment and classify according to texture. Subsequently, mutual information-guided training heuristics were provided for this algorithm and demonstrated on stem cell colony images. A procedure for automatic initialization was presented, and the algorithm was finally demonstrated on zebra images.
Chapter 6

Texture-Based BLS Applied to Non-Invasive Stem Cell Colony Segmentation and Classification

In this chapter, the texture-based Bayesian level set algorithm described in chapter 5 is used to simultaneously distinguish pluripotent from differentiated stem cell colony images and segment the colony region from the surrounding growth media. Using image texture as a non-invasive biomarker for pluripotency, this automated procedure serves as a support aid for biologists, tissue engineers, and clinicians engaged in growing, maintaining, and analyzing stem cells by enabling automated image acquisition and analysis to autonomously assess and monitor the degree of pluripotency in a non-invasive, non-destructive manner. Pluripotent classification is near perfect at 97.36% accuracy, while differentiated results are 91.38% accurate.

6.1 Motivation and impact

A scalable, verifiable method for identifying pluripotency in a non-destructive and non-invasive manner is needed for deriving rare iPSCs from somatic cells or for quality control of large numbers of hESC colonies intended for cell therapy. Unfortunately, current state-of-the-art methods in human hESC classification are limited to visual
inspection of live cells by a trained microscopist or biochemical or immunochemical staining. While visual observation using brightfield or phase contrast microscopy is non-invasive, it is time consuming, non-quantitative, and are not scalable to the large quantity of cells expected in a therapeutic or commercial setting. Likewise, while biochemical staining of hESC is consistent, quantitative, and automatable, it is destructive and renders the sample unfit for therapeutic use [101]. Live cell fluorescent markers can be used to recognize nascent iPSC colonies [17] but are invasive, requiring addition of extracellular or membrane permanent dyes that have limited application in kinetic assays because of dye loss over time and may damage photosensitive cells. Further, accurate quality control requires a measure of the homogeneity of cell morphology, which is nearly impossible to perform visually for very large numbers of cell culture plates. Morphological measurements can serve as end-point indicators of cell pluripotency or differentiation and provide real-time measurement of the experimental agents on cells.

Image texture methods have been successfully employed in numerous biological applications including fluorescent images of protein distribution in cells [20], nuclear texture characteristic of aggressive cancers in patient biopsies [95, 126], and, as demonstrated in chapter 4, classification of stem cell nuclei and colonies, where apparent texture corresponds to morphological variation indicative of pluripotency, maturity, and differentiation state [37, 49, 63, 64, 65, 69, 70, 72, 101, 102, 103].

Texture-based analysis effectively provides biologists and clinicians with a biomarker that is non-invasive, statistical, and quantitative; in this application it replaces more invasive biomarkers for use in automatically identifying and segmenting pluripotent and differentiated stem cell colony images. This methodology may additionally serve as a statistical and quantitative support aid for biologists and clinicians engaged in growing, maintaining, and analyzing hESCs or iPSCs, with the potential for automated image acquisition and analysis to autonomously assess and monitor the degree of pluripotency in a non-destructive manner.
6.2 Virtual staining

One of the key advantages of automated, image-based stem cell colony classification and segmentation is the ability to identify colony regions without resorting to destructive chemical staining. While the images shown in this chapter will be colorized to denote their class, with blue areas classified as pluripotent, green as differentiated, and red as media, image colorization is performed via computer, and no chemical staining was applied to any of the colonies imaged in this chapter. This virtual staining is non-invasive, non-destructive, and leaves the colony fit for further research or therapeutic use.

6.3 Data collection

Pluripotent human embryonic stem cells (hESCs), line WA07, were grown feeder-free on Geltrex-coated plates (Invitrogen) and maintained in StemPro (Invitrogen) a defined pluripotency media. Media was changed every other day, and colonies were passaged weekly with Collagenase Type IV. Specific and selective differentiation to an epithelial cell type was achieved by culturing for four days in StemPro without bFGF but with the addition of BMP-4 (100 ng/mL) to the media [37].

Colonies were imaged via phase-contrast microscopy using a 4x Nikon 0.13 NA objective on a TMS cell culture microscope with a 10M Pixel Nikon D40x SLR camera. RGB images were reduced to grayscale by selecting the green channel to reduce chromatic aberration and avoid color registration errors.

6.4 Colony image segmentation

The texture-based Bayesian level set algorithm described in chapter 5 was applied to automate the simultaneous segmentation and classification of a set of forty-seven (47) dual-class images, each containing growth media and a single cell type. Twenty-eight (28) of these images were pluripotent and nineteen (19) were differentiated. Since the
<table>
<thead>
<tr>
<th>Colony type</th>
<th>Overall</th>
<th>Pluripotent</th>
<th>Differentiated</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pluripotent</td>
<td>97.36%</td>
<td>97.05%</td>
<td>-</td>
<td>97.08%</td>
</tr>
<tr>
<td>Differentiated</td>
<td>91.38%</td>
<td>-</td>
<td>86.61%</td>
<td>92.62%</td>
</tr>
</tbody>
</table>

Table 6.1: Numerical results for simultaneous segmentation and classification of stem cell colonies.

The texture-based BLS algorithm was given no prior information on which classes were present within any particular image, this constitutes a true classification.

The remainder of this section first describes the experimental procedure for automated segmentation and classification of stem cell colony images and then provides the numerical results of the test. For further insight, the evolution of a typical texture-based BLS segmentation is then discussed at some length. The section concludes with a discussion

### 6.4.1 Experimental procedure

The texture-based BLS was implemented as in section 5.1 and initialized as in section 5.3. Texture priors were trained as in section 5.2; two pluripotent textons were identified, as were three each from the differentiated and media classes.

When generating results for test images, data prior weights $\eta_T$, $\lambda_T$, and $\tau_T$ were dynamically rescaled to equal half the number of pixels in the image. This was done to prevent the training data from dominating parameter estimation during testing; as the training data was drawn from a set of six half-images, the number of training samples generally exceeded the number of pixels in a test image.

### 6.4.2 Numerical results

Accuracy statistics for automated segmentation and classification of the entire set of forty-seven images are given in Table 6.1, and results for individual pluripotent and differentiated colonies are given in Figures 6-1 and 6-2. Accuracy is determined by comparing a manual segmentation of a test image to the label map $M$ obtained from
Figure 6-1: Pluripotent colony segmentation accuracy. Accuracy of the BLS results for individual pluripotent colonies; (a) overall segmentation accuracy; (b) accuracy in segmenting regions manually identified as pluripotent; (c) accuracy in segmenting regions manually identified as media; colony codes along the x-axis are the same as those in Figures 6-3 through 6-6.
Figure 6-2: *Differentiated colony segmentation accuracy.* Accuracy of the BLS results for individual differentiated colonies; (a) overall segmentation accuracy; (b) accuracy in segmenting regions manually identified as differentiated; (c) accuracy in segmenting regions manually identified as media; colony codes along the x-axis are the same as those in Figures 6-3 and 6-7 through 6-8.
the most likely \textit{a posteriori} class as determined by the automated, texture-based BLS:

\[ M(x) = \arg \max_{c \in C} q_{c|y}(x) \]  

(6.1)

Results show that the accuracy of pluripotent segmentation is quite high, with average accuracy of 97.36\%, while differentiated accuracy is 91.38\%. Segmentation of individual colony images is shown in Figures 6-3 through 6-8.

6.4.3 Evolution of a single BLS segmentation

For additional insight, the progress of the texture-based BLS in segmenting a single stem cell colony image will be demonstrated on pluripotent D. The evolving label map \( M \) for this segmentation is shown in Figure 6-9. Note how, at the initialization in Figure 6-9a, the colony border is largely misclassified as differentiated, as are some elongated, bar-shaped protrusions and isolated regions. The geometric smoothing prior corrects for this as shown in Figure 6-10. Note how the smoothing prior causes the pluripotent and media classes to flow into the misclassified border as evidenced by the light blue halo near the interface of the pluripotent and media priors as shown in Figure 6-10a and c. Simultaneously, the mixture component of the geometric prior works to systematically suppress the likelihood of the outlier differentiated class.

The evolution of texton locations is shown in Figure 6-11, which shows textons in the first two feature dimensions. In the pluripotent series (left column), the rightmost texton evolves little, but the leftmost texton noticeably shifts up in order to better model the data. Likewise in the media series (right column), the rightmost texton also shifts up, and the leftmost texton elongates to better model the data. The differentiated series (middle column) changes little, as the data priors prevent overfitting of this class to outliers.

6.4.4 Texton training for differentiated colonies

In the test set of differentiated images, four images which were judged inappropriate for segmentation via the texture-based BLS and so removed from the numerical re-
Figure 6-3: Simultaneous segmentation and classification of the training set. (a)-(c) pluripotent training images; (d)-(f) differentiated training images; note that one side of the image was reserved from the training set as shown in Figure 5-7; blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony; the asterisk next to the image code denotes a training colony; note that the green region at the bottom right of $B^*$ is correctly labeled as this colony has begun to differentiate along its borders.
Figure 6-4: Simultaneous segmentation and classification of pluripotent stem cell colonies. Blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-5: Simultaneous segmentation and classification of pluripotent stem cell colonies. Blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-6: Simultaneous segmentation and classification of pluripotent stem cell colonies. Blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-7: Simultaneous segmentation and classification of differentiated stem cell colonies. Blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-8: *Simultaneous segmentation and classification of differentiated stem cell colonies.* Blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-9: *Evolution of a single BLS segmentation: pluripotent colony segmentation label map.* (a) initialization, note that the colony border is largely misclassified as differentiated (shaded green) as are many small protrusions and isolated regions; (b) after iteration 1, the geometric smoothing prior has already begun to group like classes, thereby thinning the misclassified border and eliminating many of the isolated misclassified regions; (c) after iteration 10, most of the regions misclassified as differentiated have disappeared; (d) after iteration 191, the segmentation has reached its final state with only a small area misclassified as differentiated.
Figure 6-10: Evolution of a single BLS segmentation: pluripotent colony geometric prior. The geometric prior for each class and iteration shown in Figure 6-9. The hybrid geometric prior combines curvature-flow smoothing and mixture priors.
Figure 6-11: Evolution of a single BLS segmentation: pluripotent textons in feature dimensions 1 and 2. The blue dots comprise a subsampling of data in this image that was manually identified as pluripotent, and red dots are media; textons are indicated by black contours drawn at the half-peak of the Gaussian.
Figure 6-12: *Differentiated colonies inappropriate for segmentation via texture-based BLS.* (a)-(c) the differentiated colony has become so diffuse, and the distance between cells so great, that the BLS has trouble recognizing the colony as a distinct texture; (d) the amount of debris and other extracellular material renders it difficult for the BLS to accurately locate colony borders; blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
Figure 6-13: Retraining differentiated texton priors to recognize diffuse colonies. (a)-(b) Texture-based BLS segmentation of two differentiated colonies; while colony U is too diffuse for proper segmentation, the extracellular material in C* is correctly classified as media; (c)-(d) Texture-based BLS segmentation of the same two differentiated colonies using alternate textons trained to recognize diffuse colonies; while performance on the diffuse U has improved, despite its inclusion in the training set, the extracellular material in C* is misclassified as differentiated; blue regions were classified pluripotent, green differentiated, and red media; image colorization was performed via computer, and no chemical staining was applied to the colony.
sults. In three of these images, shown in Figure 6-12a-c, the colony has become so diffuse that the BLS fails to recognize it as a coherent texture, with the result that differentiated cells are misclassified as media. The fourth is shown in Figure 6-12d; in this image, the quantity of debris and extracellular material present causes the BLS to misclassify media as differentiated cells. While this phenomenon is partly present in more successful colony segmentations (cf. pluripotent A* and differentiated C*, M, and O), it is particularly extreme in this case.

As demonstrated in Figure 6-13, it is possible to retrain the differentiated texton data priors to improve performance on the diffuse colonies, at the cost of increased sensitivity to extracellular material in the media. While the retrained segmentation in Figure 6-13c better recognizes the diffuse colony, it fails to correctly classify the media in Figure 6-13d despite its inclusion in the training set. Given the choice between sensitivity to diffuse colonies or extracellular material in the media, some degree of process control will be necessary to successfully apply automated stem cell colony image classification in a laboratory setting.

6.5 Multi-class image phantom segmentation

The results in the previous section were performed on colony images containing a single type of cell, either pluripotent or differentiated. To assess performance on images containing a mixture of cell types, phantoms were created as shown in Figure 6-14. In order to modify the texture-based BLS for use in segmenting mixed-type colonies, the mixture ratio prior in (5.14) was localized via the use of a disk filter.

At 92% or greater, accuracy of phantom results is quite high, though with a noticeable increase in the misclassification of small outlier regions in comparison to dual-class colony segmentation. This is the result of the decreased effectiveness of the mixture ratio prior in suppressing outliers, owing to the presence of nontrivial amounts of these classes in other locations within the image.

These results, however, may be further tuned to support the needs of researchers and clinicians. Tissue engineers might be willing to accept the misclassification of
Figure 6-14: *Segmentation of mixed-class stem cell colony phantoms.* The areas shaded blue are classified pluripotent, the areas shaded green are classified differentiated, and the red regions media; (a) phantom with pluripotent and differentiated regions separated by media, accuracy is 93%; (b) phantom with differentiated colony containing a circular pluripotent region, accuracy is 92%; (c) phantom of a pluripotent colony with a large differentiated region on its border, accuracy is 94%; (d) the same phantom, but with a bias to classification rendering the pluripotent class twice as likely *a priori* as differentiated, accuracy is 95%.
differentiated cells as pluripotent in order to ensure that no tumor-causing pluripotent cells were wrongly labeled as differentiated. Likewise, a researcher maintaining stem cells in pluripotent state might be willing to discard misclassified pluripotent cells to prevent a failure to detect differentiated cells which might corrupt a colony. Classification results may be tailored to the benefit of either of these researchers by augmenting the geometric prior with a fixed mixture ratio (3.37) that biases classification towards a particular result. An example of this is shown in Figure 6-14d, which successfully eliminates spurious differentiated regions via a fixed mixture prior which judges the pluripotent cells \textit{a priori} to be twice as likely as differentiated ones.

6.6 Recognizing anomalous texture classes

Results in this chapter were derived from a texture-based BLS that was trained to recognize pluripotent, differentiated, and media textures. In a laboratory environ-
ment, however, it is possible that an untrained texture might be presented to the algorithm by contamination of a cell culture plate or some other mishap. To guard against this possibility, textural anomalies may be flagged in post-processing for human inspection and verification by comparing the maximum likelihood of the data \( q_{Y|Z} \) against some threshold \( \zeta \):

\[
\max_{x \in \mathcal{C}} q_{Y|Z}(x) < \xi \quad (6.2)
\]

\[
q_{Y|Z}(x) = \sum_{t \in T} \pi_t Z P(Y(x); \theta_t) \quad (6.3)
\]

If (6.2) obtains, pixel \( x \) is considered potentially anomalous. If the fraction of potential anomalies in the local neighborhood of pixel \( x \) exceeds some other threshold \( \zeta \), \( x \) is then reclassified as an anomaly requiring human attention. This process is illustrated in Figure 6-15 for \( \xi = 0.001 \) and \( \zeta = 0.75 \).

### 6.7 Conclusions

The texture-based Bayesian level set algorithm enables automated, simultaneous segmentation and classification of pluripotent and differentiated stem cell colony images. At 97.36% accuracy, the segmentation of pluripotent colony images is near perfect, while differentiated segmentation is 91.38% accurate. These results demonstrate that image texture models may be used as a non-invasive, non-destructive biomarker for stem cell pluripotency, with possible application to the automated monitoring of stem cell cultures and their derivatives.
Chapter 7

Summary, Contributions, and Suggestions for Future Work

In this chapter, the principal contributions of this thesis are reviewed and suggestions are made for future work.

7.1 Contributions

Contributions to image segmentation and texture modeling

This thesis provides two major and two minor contributions to the fields of segmentation and texture-based image analysis and classification. Major contributions are:

1. *The Bayesian Level Set (BLS) algorithm for image segmentation* – a new segmentation algorithm that produces the smooth, regular segmenting contours typical of level sets within a simpler, more computationally efficient Bayesian implementation similar to that of the finite mixture model EM.

2. *Simultaneous segmentation and classification via a texture-based BLS* – accomplished by fusing image texture modeling to the BLS via a three-stage Markov texture model.

*The Bayesian Level Set (BLS) algorithm for image segmentation.* The first contribution is a novel algorithm for image segmentation whose pseudocode is given in
Table 3.1. The algorithm produces smooth, regular segmentations similar to those of level sets by imposing on the classification a geometric prior formed by smoothing the previous iteration's posterior likelihood, thereby promoting local similarity in labeling. As a result, this approach is significantly more robust to noise and better at rejecting outliers than the basic FMM-EM, while also possessing a simpler and more computationally efficient implementation than the partial differential equations and ancillary algorithms required to solve level sets. See Table 3.2 for a summary of its advantages with respect to level sets and the FMM-EM.

**Simultaneous segmentation and classification via a texture-based BLS.** The second contribution is an extension to the BLS that simultaneously combines segmentation with classification against learned texture libraries. Texture is incorporated into the BLS framework using a three-stage Markov texture model, and the resulting algorithm is used to automate the segmentation and classification of stem cell colony images. See Table 5.1 for pseudocode.

**Minor contributions.** Two minor contributions to image texture modeling were developed while investigating stem cell classification. First, a framework was developed in section 4.3.1 for localizing statistical, multiresolution texture algorithms for use in segmentation. After reducing dimensionality by retaining the maximal response at any orientation within a particular scale, statistics are estimated by calculating a local mean via a Gaussian sensing kernel. Second, an adaptive windowing method was formulated in section 4.2.2 to enable classification texture patches whose small size or irregular shape precludes identification of the rectangular, texturally homogeneous windows used by most image texture algorithms.

**Contributions to non-invasive, image-based stem cell biomarkers**

By fusing the BLS with texture-based methods to automate the simultaneous segmentation and classification of stem cell colony images, this thesis demonstrates that image texture analysis serves as a non-invasive, non-destructive biomarker for assessing the pluripotency of stem cells. This new biomarker may serve as a surrogate for destructive chemical staining or visual inspection by a microscopist and is thus a
potentially enabling technology for automated stem cell cultivation.

7.2 Suggestions for future work

From a theoretical standpoint, the derivation of a convergence proof is the primary avenue for future work on the Bayesian Level Set. As discussed in section 3.6, the BLS is not an Expectation-Maximization algorithm since in general the inverse of the geometric prior does not permit exact probabilistic interpretation. Neither is the BLS exactly a gradient descent algorithm. In over two years of extensive use, however, the BLS has never been observed to diverge or exhibit limit cycle behavior when paired with a stable geometric prior, such as Gaussian diffusion or curvature flow. The local structure imposed by smoothing geometric priors, especially filter-based ones such as Gaussian diffusion, might enable an examination of the spectral radius of the Jacobian of the update in $q_{Z|Y}(x)$ [47].

From a practical standpoint, the most fruitful avenue for development would be to increase the set of cell colony textures in the texton library so as to enable classification of additional image types. In addition to the StemPro growth media and the pluripotent and trophectoderm cells classified in chapter 6, unpublished work has shown that wavelet-based texture methods will also reliably distinguish homogeneous image patches containing mouse embryonic stem cells and fibroblasts, neural stem cells and neurectoderm, embryonic kidney cells, and oligodendrocyte precursor cells. While the wavelet-based analysis has shown the possibility of texture-based classification of these cell types, extension of these results to simultaneous classification and segmentation is likely to require the formulation of additional texture features and their incorporation into the training set. As the zebra example in section 5.4 illustrates, the best features for texture-based segmentation are generally problem-specific; a single bar feature is adequate for stem cells while several are required to segment zebras. Too, as the number of classes within the library increases it becomes computationally prohibitive to continually calculate a posteriori probability for each of them. Automated segmentation and classification over a larger pool of potential
cell types is therefore likely to require some methodology for pre-screening the potential classes present within an image. The windowing method described in section 4.2.3 may prove helpful in this regard.

Another important extension of the texture-based BLS would be to augment the three-stage Markov texture model with a "junk class" which served to detect anomalous, untrained textures on-line rather than in post-processing, as in section 6.6. While this has obvious practical application in improving algorithm robustness, incorporation of an outlier class within the texture model is likely to prove an interesting theoretical challenge, as well.

Lastly, improvements in automating the texton training in section 5.2 might mitigate the need for operator intervention in training the texture-based BLS. As mutual information has already been employed to evaluate features for classification in section 5.2.1, max-relevance / min-redundancy (MRMR) criteria such as [87] might be employed to automate feature selection. Proper scaling on the redundancy term may prove an obstacle to such; developments of MRMR (e.g. [38]) typically normalize the redundancy term by dividing by feature entropy; this is inappropriate for texture features as their differential entropy can be and often is negative. A further challenge to information theoretic procedures would be adequate evaluation of features such as hdiff, which have low feature-class mutual information but are invaluable at preventing specific types of misclassification. (See Figure 5-9.) Information criteria (Akaike, Bayesian) might also be used to choose the number of textons in section 5.2.3.
Appendix A

Notation

This appendix lists some of the more common functions and variables in this text.

- $|.|$ when the argument is a set, the number of elements in the set
- $c$ a specific class from set $C$
- $C$ the set of all image classes
- $H(\cdot)$ the Heaviside step function
- $J$ a covariance matrix
- $q_{Y|Z}^k(x)$ $P^k(Y(x) | Z(x); \theta_Z)$, the probability at coordinate $x$ and iteration $k$
  of data $Y$ conditioned on class $Z$ with parameters $\theta_Z$;
  other probabilities are abbreviated similarly
- $t$ a specific texton from set $T$
- $T$ the texton, or local texture appearance, as an unknown or latent variable
- $\mathcal{T}$ the set of all image textons
- $x$ a coordinate, typically a pixel, from domain $\Omega$
- $Y$ image data or features
- $Z$ the image class as an unknown or latent variable
- $\alpha$ weight of the data fidelity term
- $\beta$ weight of the curvature term
- $\gamma$ weight of the area term
\( \delta(\cdot) \)  the Dirac delta function
\( \eta \)  parameter of the Dirichlet prior;
   equal to the training set empirical mixture ratio
\( \theta \)  observation parameters for a specific class
\( \Theta \)  the set of all parameters in a given problem
\( \kappa(\cdot) \)  the curvature of the iso-contours of the argument;
   equal to the divergence of the normal
\( \lambda \)  parameter of the NIW data prior;
   equal to the number of samples used to train \( \nu \)
\( \mu \)  a mean
\( \nu \)  parameter of the NIW data prior;
   equal to the training set empirical feature mean
\( \pi \)  multinomial parameter;
   equal to the mixture ratio of a particular class
\( \sigma \)  a scale parameter or standard deviation
\( \tau \)  parameter of the NIW data prior;
   equal to the number of samples used to train \( \psi \)
\( \phi \)  a level set function
\( \psi \)  parameter of the NIW data prior;
   equal to the training set empirical feature covariance matrix
\( \Omega \)  the domain of all coordinates in an image
Appendix B

Derivations for smoothing conditions

In this appendix, common smoothing terms are derived from the variations of corresponding penalty terms. Each case follows a common procedure:

1. Calculate the Gateaux differential of the penalty term.
2. Recover the increment using the identity \( \mathbf{x} \cdot \nabla f = -f \nabla \cdot \mathbf{x} + \nabla \cdot (f \mathbf{x}) \).
3. Introduce Neumann boundary conditions using the divergence theorem on the plane.

B.1 Diffusion from energy

Let \( u \) be some function defined on the region \( \Omega \). Its two-norm energy is:

\[
E(u) = \frac{1}{2} \int_\Omega \nabla u \cdot \nabla u d\Omega
\]  

\text{(B.1)}

The Fréchet differential is technically correct for determining the variation; as this function is fairly well-behaved, proceed by assuming it exists. Given that assumption, the simpler Gateaux differential exists and is equivalent to the Fréchet differential\(^1\). \footnote{\text{[66]} contains a very helpful review of these concepts.}
For increment $\psi$, the Gateaux differential is:

$$
\delta(E(u) ; \psi) = \int_\Omega \delta \left( \frac{1}{2} \nabla u \cdot \nabla u ; \psi \right) d\Omega \quad (B.2)
$$

$$
\delta \left( \frac{1}{2} \nabla u \cdot \nabla u ; \psi \right) = \frac{1}{2} \lim_{\alpha \to 0} \frac{\nabla(u + \alpha \psi) \cdot \nabla(u + \alpha \psi) - \nabla u \cdot \nabla u}{\alpha} = \nabla u \cdot \nabla \psi \quad (B.3)
$$

In step two, recover the increment from its gradient using the identity $x \cdot \nabla f = -f \nabla \cdot (x) + \nabla \cdot (fx)$, with $f = \psi$ and $x = \nabla u$:

$$
\delta(E(u) ; \psi) = \int_\Omega \nabla u \cdot \nabla \psi d\Omega = -\int_\Omega \psi \Delta u d\Omega + \int_\Omega \nabla \cdot (\psi \nabla u) d\Omega \quad (B.5)
$$

In the final step, introduce Neumann boundary conditions by applying the divergence theorem on a plane:

$$
\delta(E; \psi) = -\int_\Omega \psi \Delta u d\Omega + \int_{\partial\Omega} \psi \frac{\partial u}{\partial n} d\omega \quad (B.6)
$$

Where $n$ is the outward normal to the perimeter $\partial\Omega$.

Minimizing conditions, then, are:

$$
0 = -\Delta u \quad \text{on } \Omega \quad (B.7)
$$

$$
0 = \frac{\partial u}{\partial n} \quad \text{on } \partial\Omega \quad (B.8)
$$

The first equation, of course, is simple diffusion.

### B.2 Curvature from total variation

Total variation is:

$$
TV(u) = \frac{1}{2} \int_\Omega \| \nabla u \| d\Omega \quad (B.9)
$$
As in the previous section, take the Gateux differential:

\[
\delta(TV(u); \psi) = \int_\Omega \delta(\frac{1}{2} \| \nabla u \| ; \psi) \, d\Omega \tag{B.10}
\]

\[
\delta(\| \nabla u \| ; \psi) = \frac{1}{2} \lim_{\alpha \to 0} \frac{\| \nabla (u + \alpha \psi) \| - \| \nabla u \|}{\alpha} \tag{B.11}
\]

\[
= \frac{1}{2} \lim_{\alpha \to 0} \left( \frac{\| \nabla u + \alpha \nabla \psi \| - \| \nabla u \|}{\alpha} \right) \left( \frac{\| \nabla u + \alpha \psi \| + \| \nabla u \|}{\| \nabla u + \alpha \psi \| + \| \nabla u \|} \right) \tag{B.12}
\]

\[
= \lim_{\alpha \to 0} \frac{\| \nabla u + \alpha \nabla \psi \|^2 + \| \nabla u \|^2}{2 \alpha \| \nabla u + \alpha \nabla \psi \| + \| \nabla u \|} \tag{B.13}
\]

\[
= \lim_{\alpha \to 0} \frac{\nabla u \cdot \nabla \psi + \alpha \| \nabla \psi \|^2}{\| \nabla u \|} \tag{B.14}
\]

\[
= \frac{\nabla u \cdot \nabla \psi}{\| \nabla u \|} \tag{B.15}
\]

(B.14) uses the identity that \( \| x \|^2 = x \cdot x \).

From this point, the derivation proceeds as in the previous section, with \( x = \nabla u / \| \nabla u \| \) in step two. The minimizing conditions are thus:

\[
0 = -\nabla \cdot \left( \frac{\nabla u}{\| \nabla u \|} \right) \quad \text{on } \Omega \tag{B.16}
\]

\[
0 = \frac{1}{\| \nabla u \|} \frac{\partial u}{\partial n} \quad \text{on } \partial \Omega \tag{B.17}
\]

(B.16) is simply the definition of curvature \( \kappa(u) \), the divergence of the normal:

\[
\kappa(u) = \nabla \cdot \left( \frac{\nabla u}{\| \nabla u \|} \right) \tag{B.18}
\]

### B.3 Curvature from contour length

In terms of the level set function \( \phi \), the contour arc length of the zero level set of \( \phi \) may be determined as:

\[
C(\phi) = \int_\Omega \delta(\phi) \| \nabla \phi \| \, d\Omega \tag{B.19}
\]

Begin calculating the Gateux differential by first invoking the chain rule:

\[
\delta(C; \psi) = \int_\Omega \psi \delta(\| \nabla \phi \|) \, d\Omega + \int_\Omega \delta(\| \nabla \phi \|; \psi) \, d\Omega \tag{B.20}
\]
Where $\nabla \delta(\phi) = \delta'(\phi) \nabla \phi$. By (B.15), this is equivalent to:

$$\delta(C; \psi) = \int_{\Omega} \delta' \nabla \phi \psi \, d\Omega + \int_{\Omega} \delta(\phi) \frac{\nabla \phi \cdot \nabla \psi}{\| \nabla \phi \|} \, d\Omega \quad \text{(B.21)}$$

Now, apply the identity $x \cdot \nabla f = -f \nabla \cdot (x) + \nabla \cdot (fx)$ twice. First, with $f = \psi$ and $x = \delta(\phi) \nabla \phi / \| \nabla \phi \|$, and second, with $f = \delta(\phi)$ and $x = \nabla \phi / \| \nabla \phi \|$:

$$\delta(\phi) \frac{\nabla \phi}{\| \nabla \phi \|} \cdot \nabla \psi = -\psi \nabla \cdot \left( \delta(\phi) \frac{\nabla \phi}{\| \nabla \phi \|} \right) + \nabla \cdot \left( \psi \delta(\phi) \frac{\nabla \phi}{\| \nabla \phi \|} \right)$$

$$= -\psi \delta'(\phi) \| \nabla \phi \| - \psi \delta(\phi) \kappa(\phi) + \nabla \cdot \left( \psi \delta(\phi) \frac{\nabla \phi}{\| \nabla \phi \|} \right) \quad \text{(B.22)}$$

Combined with (B.21), this results in:

$$\delta(C; \psi) = -\int_{\Omega} \psi \delta(\phi) \kappa(\phi) \, d\Omega + \int_{\Omega} \nabla \cdot \left( \psi \delta(\phi) \frac{\nabla \phi}{\| \nabla \phi \|} \right) \, d\Omega \quad \text{(B.24)}$$

Apply the divergence theorem in the plane to the latter term:

$$\delta(C; \psi) = -\int_{\Omega} \psi \delta(\phi) \kappa(\phi) \, d\Omega + \int_{\partial \Omega} \psi \delta(\phi) \frac{\phi}{\| \nabla \phi \|} \, n \, d\omega \quad \text{(B.25)}$$

The minimizing conditions are thus:

$$0 = -\delta(\phi) \nabla \cdot \left( \frac{\nabla \phi}{\| \nabla \phi \|} \right) \quad \text{on } \Omega \quad \text{(B.26)}$$

$$0 = \frac{\delta(\phi)}{\| \nabla \phi \|} \frac{\partial \phi}{\partial n} \quad \text{on } \partial \Omega \quad \text{(B.27)}$$
Appendix C

Derivation of the GMM-EM

C.1 Derivation of the GMM-EM update equations

The expected log-likelihood of the Gaussian Mixture model is:

\[ E_Q [L(Y \mid \mu, J, \pi)] = \sum_{x \in \Omega} \sum_{c \in C} q_c(x) [\log \pi_c + \log N(Y \mid \mu_c, J_c)] \quad (C.1) \]

where:

\[ \log N(Y \mid \mu_c, J_c) = -\frac{d}{2} \log 2\pi - \frac{1}{2} \log |J_c| - \frac{1}{2} (Y(x) - \mu_c)^T J_c^{-1} (Y(x) - \mu_c) \quad (C.2) \]

To form the utility function, adjoin the entropy term to obtain the Kullback-Leibler divergence (as in (2.22)) and constrain probabilities to sum to one using Lagrange multipliers \( \xi \):

\[ D(Q\|P(Y, Z \mid \Theta)) = \sum_{x \in \Omega} \sum_{c \in C} q_c(x) [\log \pi_c + \log N(Y \mid \mu_c, J_c)] - \sum_{x \in \Omega} \sum_{c \in C} q_c(x) \log q_c(x) \]

\[ + \xi_{\pi} \left( 1 - \sum_{c \in C} \pi_c \right) + \sum_{x \in \Omega} \xi_q(x) \left( 1 - \sum_{c \in C} q_c(x) \right) \quad (C.3) \]
C.1.1 Expectation step

To calculate the Expectation step, take the partial derivative with respect to $q_c(x)$:

$$0 = \frac{\partial}{\partial q_c(x)} D(Q\|P(Y, Z | \Theta))$$

$$= \log \pi_c + \log N(Y(x); \mu_c, J_c) - \log q_c(x) - (\xi_q(x) + 1)$$

Rearrange terms to recover:

$$q_c(x) = \pi_c N(Y(X); \mu_c, J_c) e^{-(\xi_q(x)+1)}$$

(C.5)

By enforcing the constraint that $1 = \sum_c q_c$, solve for and eliminate $\xi_q$:

$$q_c(x) = \frac{\pi_c N(Y(x); \mu_c, J_c)}{\sum_{c \in C} \pi_c N(Y(x); \mu_c, J_c)}$$

(C.6)

(C.6) is the update equation for the Expectation step.

C.1.2 Maximization step

The Maximization step consists of three parts, solving for $\pi$, $\mu$, and $J$. Begin by taking the partial derivative of (C.3) with respect to $\pi_c$:

$$0 = \frac{\partial}{\partial \pi_c} D(Q\|P(Y, Z | \Theta)) = \frac{\sum_{x \in \Omega} q_c(x)}{\pi_c} - \xi_\pi$$

(C.7)

Rearrange terms and enforce the constraint that $1 = \sum_c \pi_c$ to eliminate the Lagrange multiplier and recover:

$$\pi_c = \frac{1}{|\Omega|} \sum_{x \in \Omega} q_c(x)$$

(C.8)

Now, take the partial derivative with respect to $\mu_c$, using the identities $\partial_A \text{tr}(AB) = B^T$ and $\partial_A \text{tr}(ABA^T) = 2AB$:

$$0 = \frac{\partial}{\partial \mu_c} D(Q\|P(Y, Z | \Theta)) = -J_c^{-1} \mu_c \sum_{x \in \Omega} q_c(x) + J_c^{-1} \sum_{x \in \Omega} q_c(x) Y(x)$$

(C.9)
Rearrange to obtain:
\[ \mu_c = \frac{\sum_{x \in \Omega} q_c(x) Y(x)}{\sum_{x \in \Omega} q_c(x)} \]  
(C.10)

Finally, take the partial with respect to \( J_c \), using the identities that \( \partial_A |A| = |A| A^T \) and \( x^T A^{-1} y = -y A^{-T} xy^T A^{-T} \):

\[ 0 = \frac{\partial}{\partial J_c} D(Q\|P(Y, Z \mid \Theta)) \]
\[ = -\frac{1}{2} J_c^{-1} \sum_{x \in \Omega} q_c(x) - \frac{1}{2} \sum_{x \in \Omega} q_c(x) J_c^{-1} (Y(x) - \mu_c)(Y(x) - \mu_c)^T J_c^{-1} \]

Eliminate redundant terms and rearrange:
\[ J_c = \frac{\sum_{x \in \Omega} q_c(x) (Y(x) - \mu_c)(Y(x) - \mu_c)^T}{\sum_{x \in \Omega} q_c(x)} \]  
(C.12)

(C.8), (C.10), and (C.12) are the update equations for the Maximization step.

C.2 Derivation with Dirichlet and Normal-Inverse Wishart priors

To incorporate prior information into the estimates of \( \pi, \mu, \) and \( J \), impose prior distributions on these parameters:

\[ \pi \sim D(\eta) \]  
(C.13)

\[ \Rightarrow L(\pi; \eta) = -B(\eta) + \sum_{c \in \mathcal{C}} (\eta_c - 1) \log \pi_c \]

\[ \mu_c \sim N\left(\nu_c, \frac{1}{\lambda_c} J_c \right) \]  
(C.14)

\[ \Rightarrow L(\mu_c; \nu_c, \frac{1}{\lambda_c} J_c) = -\frac{1}{2} d \log 2\pi + \frac{1}{2} \log |J_c| - \frac{1}{2} \lambda_c (\mu_c - \nu_c)^T J_c^{-1} (\mu_c - \nu_c) \]

\[ J_c \sim W^{-1}(\tau_c \psi_c, \tau_c) \]  
(C.15)

\[ \Rightarrow L(J_c; \psi_c, \tau_c) = \frac{1}{2} \tau_c d \log 2 - \log \Gamma_d \left( \frac{1}{2} \tau_c \right) + \frac{1}{2} \tau_c \log \tau_c + \frac{1}{2} \tau_c \log |\psi_c| - \frac{1}{2} (\tau_c + d + 1) \log |J_c| - \frac{1}{2} \text{tr}(\tau_c \psi_c J_c^{-1}) \]
\( d \) is the dimension of the observation \( Y(x) \).

To find the effect of adjoining these priors, add the log-likelihood from the equation above to (C.3) and solve for the respective parameter. For instance:

\[
0 = \frac{\partial}{\partial \pi_c} \left( D(Q \| P(Y, Z \mid \Theta)) + L(\pi; \eta) \right) = \sum_{x \in \Omega} q_c(x) - 1 + \frac{1}{\pi_c} - \xi \tag{C.16}
\]

Solve as in (C.8) to obtain:

\[
\pi_c = \frac{\sum_{x \in \Omega} q_c(x) - 1 + \eta}{\sum_{x \in \Omega} \left( \sum_{x \in \Omega} q_c(x) - 1 + \eta \right)} \tag{C.17}
\]

Likewise, to obtain \( \mu_c \):

\[
0 = \frac{\partial}{\partial \mu_c} \left( D(Q \| P(Y, Z \mid \Theta)) + L(\mu_c; \nu_c, \frac{1}{\lambda_c} J_c) \right) \tag{C.18}
\]

\[
= -J_c^{-1} \sum_{x \in \Omega} q_c(x) + J_c^{-1} \sum_{x \in \Omega} q_c(x) Y(x) - \lambda_c J_c^{-1} \mu_c + \lambda_c J_c^{-1} \nu_c \tag{C.19}
\]

and solve for \( \mu_c \):

\[
\mu_c = \frac{\sum_{x \in \Omega} q_c(x) Y(x) + \lambda_c \nu_c}{\sum_{x \in \Omega} q_c(x) + \lambda_c} \tag{C.20}
\]

To obtain \( J_c \), it is necessary to adjoin both the \( L(J_c; \psi_c, \tau_c) \) and \( L(\mu_c; \nu_c, \frac{1}{\lambda_c} J_c) \) as \( J_c \) is a parameter to the latter term.

\[
0 = \frac{\partial}{\partial J_c} \left( D(Q \| P(Y, Z \mid \Theta)) + L(\mu_c; \nu_c, \frac{1}{\lambda_c} J_c) + L(J_c; \psi_c, \tau_c) \right) \tag{C.21}
\]

\[
= -\frac{1}{2} J_c^{-1} \sum_{x \in \Omega} q_c(x) + \frac{1}{2} \sum_{x \in \Omega} q_c(x) J_c^{-1} (Y(x) - \mu_c) (Y(x) - \mu_c)^T J_c^{-1} \tag{C.22}
\]

\[
- \frac{1}{2} (\tau_c + d + 2) J_c^{-1} + \frac{1}{2} J_c^{-1} \psi_c J_c^{-1} + \frac{1}{2} \lambda_c J_c^{-1} (\mu_c - \nu_c) (\mu_c - \nu_c)^T J_c^{-1}
\]

Rearrange to obtain:

\[
J_c = \frac{\sum_{x \in \Omega} q_c(x) (Y(x) - \mu_c) (Y(x) - \mu_c)^T + \lambda_c (\mu_c - \nu_c) (\mu_c - \nu_c)^T + \psi_c}{\sum_{x \in \Omega} q_c(x) + \tau_c + d + 2} \tag{C.23}
\]
Note that many references present this equation in the form:

\[ J_c = \sum_{x \in \Omega} q_c(x) \left( (Y(x) - \bar{Y}_c) (Y(x) - \bar{Y}_c)^T + \frac{\lambda_c n_c}{n_c + \tau_c + n_c} (\bar{Y}_c - \nu_c) (\bar{Y}_c - \nu_c)^T + \psi_c \right) \]

\[ \frac{n_c + \tau_c + n_c}{n_c + \tau_c + n_c} \]

where:

\[ \bar{Y}_c = \frac{1}{n_c} \sum_{x \in \Omega} q_c(x) Y(x) \]  
\[ n_c = \sum_{x \in \Omega} q_c(x) \]  

This form can be reached by applying (C.20) and the following identity to (C.23):

\[ \sum_{x \in \Omega} q_c(x) (Y(x) - \mu_c) (Y(x) - \mu_c)^T \]

\[ = n_c (\bar{Y}_c - \mu_c) (\bar{Y}_c - \mu_c)^T + \sum_{x \in \Omega} q_c(x) (Y(x) - \bar{Y}_c) (Y(x) - \bar{Y}_c)^T \]

C.3 Derivation of the three-stage GMM-EM

Now, pose a three stage GMM, in which class \( c \in C \) leads to subclass \( t \in T \) which leads to data \( Y \). The log-likelihood function for such a three stage GMM-EM is:

\[ E_Q [L(Y; \Theta)] = \sum_{x \in \Omega} \sum_{c \in C} q_c(x) \log \pi_c \]

\[ + \sum_{x \in \Omega} \sum_{c \in C} \sum_{t \in T} q_c(x) q_{ct} \left[ \log \pi_{tc} + \log P(Y(x); \theta_t) \right] \]

To form the utility function, adjoin the entropy term to obtain the Kullback-Leibler divergence (as in (2.22)) and constrain probabilities to sum to one using Lagrange multipliers \( \xi \). For the sake of concreteness, assume also that data model is Normal so
that \( \theta_t = \{ \mu_t, J_t \} \). The result is:

\[
D(Q \| P(Y, Z \mid \Theta)) = \sum_{x \in \Omega} \sum_{c \in C} q_c(x) \log \pi_c
\]

\[
+ \sum_{x \in \Omega} \sum_{c \in C} \sum_{t \in T} q_c(x) q_{t|c} \left[ \log \pi_{t|c} + \log N(Y(x) ; \mu_t, J_t) \right]
\]

\[
- \sum_{x \in \Omega} \sum_{c \in C} q_c(x) \log q_c(x) q_{t|c}(x)
\]

\[
+ \xi_c \left( 1 - \sum_{c \in C} \pi_c \right) + \sum_{c \in C} \xi_t(c) \left( 1 - \sum_{t \in T} \pi_{t|c} \right)
\]

\[
+ \sum_{x \in \Omega} \sum_{c \in T} \xi_{q_t|c}(x) \left( 1 - \sum_{c \in T} q_c(x) \right) + \sum_{x \in \Omega} \sum_{c \in C} \xi_{q_t|c}(x, c) \left( 1 - \sum_{t \in T} q_{t|c}(x) \right)
\]

(C.30) (C.31) (C.32) (C.33) (C.34)

\section{C.3.1 Expectation step}

To calculate the Expectation step, first take the partial derivative with respect to \( q_{t|c} \):

\[
0 = \frac{\partial}{\partial q_{t|c}(x)} D(Q \| P(Y, Z \mid \Theta))
\]

\[
= q_c(x) \left[ \log \pi_{t|c} N(Y(x) ; \mu_t, J_t) - \log q_{t|c}(x) - (\log q_c(x) + 1 + \xi_{q_t|c}(x, c)) \right]
\]

(C.35) (C.36)

Divide by \( q_c \) and enforce the constraint as with (C.6) to obtain:

\[
q_{t|c}(x) = \frac{\pi_{t|c} N(Y(x) ; \mu_t, J_t)}{\sum_{t \in T} \pi_{t|c} N(Y(x) ; \mu_t, J_t)}
\]

(C.37)

Next, take the partial derivative with \( q_c(x) \):

\[
0 = \frac{\partial}{\partial q_c(x)} D(Q \| P(Y, Z \mid \Theta))
\]

\[
= -\log q_c(x) + \log \pi_c + \sum_{t \in T} q_{t|c} \log \frac{\pi_{t|c} N(Y(x) ; \mu_t, J_t)}{q_{t|c}(x)} - (1 + \xi_{q_t|c}(x, c))
\]

\[
= -\log q_c(x) + \log \pi_c + \sum_{t \in C} \pi_{t|c} N(Y(x) ; \mu_t, J_t) - (1 + \xi_{q_t|c}(x))
\]

180
To get from the second to the third line, apply (C.37). Enforce the constraint:

\[ q_c(x) = \frac{\pi_c \sum_{t \in T} \pi_{tlc} N(Y(x); \mu_t, J_t)}{\sum_{c \in C} \pi_c \sum_{t \in T} \pi_{tlc} N(Y(x); \mu_t, J_t)} \]  \hspace{1cm} (C.39)

(C.37) and (C.39) are the update equations for the Expectation step.

### C.3.2 Maximization step

The Maximization step is fairly straightforward. The derivation for \( \pi_c \) is identical to (C.8). Likewise, the derivations for \( \pi_{tlc} \), \( \mu_t \), and \( J_t \) are the same as those in (C.8), (C.10), and (C.12) for \( q_c(x) \rightarrow q_c(x) q_{tlc}(x) \). The resulting Maximization update equations are:

\[
\pi_c = \frac{1}{|\Omega|} \sum_{x \in \Omega} q_c(x) \]  \hspace{1cm} (C.40)

\[
\pi_{tlc} = \frac{\sum_{x \in \Omega} q_c(x) q_{tlc}(x)}{\sum_{x \in \Omega} q_c(x)} \]  \hspace{1cm} (C.41)

\[
\mu_t = \frac{\sum_{x \in \Omega} \sum_{c \in C} q_c(x) q_{tlc}(x) Y(x)}{\sum_{x \in \Omega} \sum_{c \in C} q_c(x) q_{tlc}(x)} \]  \hspace{1cm} (C.42)

\[
J_t = \frac{\sum_{x \in \Omega} \sum_{c \in C} q_c(x) q_{tlc}(x) (Y(x) - \mu_t)(Y(x) - \mu_t)^T}{\sum_{x \in \Omega} \sum_{c \in C} q_c(x) q_{tlc}(x)} \]  \hspace{1cm} (C.43)
Bibliography


