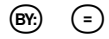
LEARNING 3-D MODELS OF OBJECT STRUCTURE FROM IMAGES

by

Joseph William Schlecht



A Dissertation Submitted to the Faculty of the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2010

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Joseph William Schlecht
entitled Learning 3-D Models of Object Structure from Images
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 8 April 2010
Kobus Barnard

Date: 8 April 2010
Alon Efrat

Date: 8 April 2010
Clayton Morrison

Date: 8 April 2010
Ian Fasel

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 8 April 2010
Dissertation Director: Kobus Barnard

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

SIGNED: Joseph William Schlecht

ACKNOWLEDGEMENTS

Many have helped to make this dissertation possible. I would first like to acknowledge the generous financial support from the University of Arizona BIO5 Institute, the UA College of Science Galileo Circle, and the National Science Foundation. Their support permitted me to remain focused on the research culminating in this dissertation. I would like to thank the members of my committee, Kobus Barnard, Alon Efrat, Clayton Morrison, and Ian Fasel, for their interesting discussions and helpful comments contributing towards the completion of this work. I owe a great deal of gratitude to my PhD supervisor, Kobus Barnard. He has been an exceptional mentor by sharing his experience, providing seasoned advice, and consistently challenging me to develop more critical and clearer arguments.

I would like to thank my colleagues and collaborators at the University of Arizona. In particular, I wish to thank Barry Pryor for the *Alternaria* data and images shown in Figure 2.1. I would also like to thank Kate Spriggs for developing a grammar for *Alternaria* growth and generating the images in Figure 3.1. Thanks to Kevin Lin for discussions and enlightening comments on MCMC sampling, and to John Kececioglue for inspiring ideas relating to image likelihood weight estimation. I am deeply indebted to Nirav Merchant and Arizona Research Labs for their support and abundant opportunities to forge exciting collaborations. Thanks to my friends and colleagues, Prasad Gabbur and Luca del Perro, for the many stimulating discussions regarding most aspects of this work.

I am thankful to the McNair Scholars program at North Dakota State University for giving me the experience to successfully follow a trajectory leading to this dissertation. I would like to thank Kay Modin, the program director, for her impossibly upbeat attitude and consistent motivation. I am especially thankful to Ken Nygard, my McNair faculty mentor, who provided me with my first research opportunity, thoughtful guidance, and an ideal environment to discover a direction. Thanks to Karl Altenburg for his guidance as well, and to John Martin and Bruce Erickson for their rigorous instruction.

I wish to thank my family and friends who supported me personally along this endeavor. I am grateful to Tom Tollefson for showing me how enjoyable life can be if you pursue your interests. Thanks to Albert and David Tingley for demonstrating how to aim for a goal and put in the hard work to achieve it. A special thanks to Mary Schlecht, Heidi Dunek, and Leah Klocke, who believed in me and encouraged me to follow this path. Thanks to Gene Schlecht for his lasting inspiration.

Finally, I would like to thank my wonderful wife, Kellie Erickson. This work would not have been possible without her as a source of fresh perspective and balance. I am forever grateful for her unwavering optimism and steadfast support along this journey.

DEDICATION

For Gene.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

Recognizing objects in images is an effortless task for most people. Automating this task with computers, however, presents a difficult challenge attributable to large variations in object appearance, shape, and pose. The problem is further compounded by ambiguity from projecting 3-D objects into a 2-D image. In this thesis we present an approach to resolve these issues by modeling object structure with a collection of connected 3-D geometric primitives and a separate model for the camera. From sets of images we simultaneously learn a generative, statistical model for the object representation and parameters of the imaging system. By learning 3-D structure models we are going beyond recognition towards quantifying object shape and understanding its variation.

We explore our approach in the context of microscopic images of biological structure and single view images of man-made objects composed of block-like parts, such as furniture. We express detected features from both domains as statistically generated by an image likelihood conditioned on models for the object structure and imaging system. Our representation of biological structure focuses on *Alternaria*, a genus of fungus comprising ellipsoid and cylinder shaped substructures. In the case of man-made furniture objects, we represent structure with spatially contiguous assemblages of blocks arbitrarily constructed according to a small set of design constraints.

We learn the models with Bayesian statistical inference over structure and camera parameters per image, and for man-made objects, across categories, such as chairs. We develop a reversible-jump MCMC sampling algorithm to explore topology hypotheses, and a hybrid of Metropolis-Hastings and stochastic dynamics to search within topologies. Our results demonstrate that we can infer both 3-D object and camera parameters simultaneously from images, and that doing so improves understanding of structure in images. We further show how 3-D structure models can be inferred from single view images, and that learned category parameters capture structure variation that is useful for recognition.

CHAPTER 1

Introduction

A central challenge in computer vision is automatically recognizing and understanding information captured in images. Humans have an amazing capacity to effortlessly accomplish this everyday. We open our eyes and immediately recognize people, cars, trees, landmarks, and situations, like danger or safety. Even when confronted with large variations in object appearance, we immediately recognize examples of diverse classes, such as birds and plants. Moreover, when the view of an object is severely obstructed, we readily imagine what has been concealed. For example, describing the parts of a chair hidden from view by a table and estimating their location is an easy task for most people. This suggests that our representation for object classes abstracts away distinctive structure information from specific instances and assists us in recognition under unseen views and appearance.

Recognition for computers, unfortunately, is not as easy. Simply providing images from a camera to a computer yields no more than a collection of numbers representing pixel intensities. We must instead devise algorithms that interpret the information encoded in these numbers. History suggests this is a difficult problem. Indeed, researchers have been trying to solve problems in computer vision for over four decades. As a result, the path of development through vision related algorithms is lengthy and punctuated with many successes (Section 1.4), but never achieving a fully functioning vision system.

In the early days of vision, much work in recognition was done with three-dimensional representations. It was well understood that three-dimensional representations provide significantly more information about 3-D objects in the world than just what is projected into a 2-D image. These model-based approaches focused on matching a known three-dimensional model to single view images (Clowes, 1971; Binford, 1971; Brooks, 1981; Pentland, 1987; Huttenlocher and Ullman, 1990) and described how an object category could be represented with a base set of three-dimensional parts or components (Winston,

1975; Biederman, 1987). A limitation these approaches shared, however, was the inability to automatically learn structure models for object classes; the models were assumed to be provided prior to analysis. Further, they did not generalize well to the variation within object classes. Today, modern approaches have focused on learning statistical models over appearance patterns and their 2-D spatial arrangement in images (Fergus et al., 2003; Leibe et al., 2004; Sivic et al., 2005; Shotton et al., 2005; Ferrari et al., 2009). These view-based representations go much further towards capturing class variation due to their statistical nature. But because the models exist only in two-dimensions, the structure variation is confounded with object view and pose. In our approach, we propose building on key ideas from both of these directions: learning 3-D representations with a statistical model over structural variation.

In this thesis we develop the idea of learning three-dimensional, part-based representations of object classes from images. The work presented here can be grouped into three primary aspects of our approach: the idea of using a three-dimensional representation for object classes and their structure (Section 1.1), basing an object representation on an assemblage of geometric primitives which loosely correspond to human-identifiable parts (Section 1.2), and modeling images as statistically generated by three-dimensional object representations (Section 1.3).

We explore this approach in the context of two related problems: inferring representations of biological structure and man-made objects composed of easily identifiable block-like parts, such as furniture. For biological structure, we infer a connected set of geometric primitives that follow a specimen-specific pattern of growth described by a grammar. In the case of block based objects, we do not use an explicit grammar to describe the structural organization, but still learn models of connected structure from a vocabulary of primitives. In both types of structure we present algorithms to infer instances of our 3-D geometric models from images. While fitting our block model to images, however, we go beyond inferring individual instances and illustrate the process of learning whole categories of structure for object classes. We could extend the biological model in the same way to learn category descriptions for biological structure, thus enabling, for example, a quantitative description of species.

Although biological and man-made structures come from two completely different domains of object categories, our approach to learning them successfully shares a similar representation and inference process. Both types of structure utilize three-dimensional representations and a vocabulary of geometric primitives. Further, the structure models are built upon similar statistically generative frameworks. When combined with a model of the imaging system, we can use these to generate instances of observed data. For learning both biological and man-made structure models, we design similar Bayesian inference algorithms and implement trans-dimensional Markov chain Monte Carlo sampling. Finally, while the imaging systems capturing data in each case are potentially quite different, e.g., a microscope with compound optics capturing stacks of images versus a simple camera imaging single views, we abstract them away from the structure model, enabling similar inference of both. This is a unique characteristic to our approach, removing confounding effects that the imaging system can have when capturing data.

The following sections of this introduction give a brief overview of the main points in our approach, followed by a related work section. In Chapter 2, we present our ideas and methods for modeling independent, three-dimensional parts of biological structure in microscopic images. Chapter 3 extends that work to inferring complete structure models guided by a biologically-inspired grammar that describes patterns of growth. Chapter 4 shifts the focus to fitting models of man-made objects composed of block-like parts to single view images. Lastly, Chapter 5 explores learning the assemblage of blocks that can represent an object category and how to infer its instances from sets of 2-D images.

1.1 Three-dimensional representation

We develop an approach to object recognition that focuses on learning three-dimensional geometric representations for object structure. Captured images of an object give two-dimensional views of what is actually three-dimensional. If we understand the three-dimensional representation of an object, we can explain its projected 2-D image observations, even under challenging conditions. This includes situations where the object is heavily occluded or viewed from an angle that has not been seen before. A 3-D geometric

model for object structure is a strong representation able to disambiguate many projected 2-D configurations. The utility of such models goes beyond just recognition; they can be used to quantify, localize, and interact with objects in three-dimensions, enabling their application to many other vision tasks.

By using a three-dimensional representation, we can separate the object model and its pose from the imaging system capturing it. This simplifies the process of understanding observed images. For example, if we model multiple objects in a scene, separating camera from object reduces the difficulty of inference considerably because all objects with similar orientations in a scene agree on a single camera configuration. This shared view can further reinforce weak detections of other objects in the scene that may not have been correctly identified under a different view.

An alternative to separating the view from an object model is building a spatial representation into the image plane. These view-based approaches learn representations whose appearance includes information about the camera (Fergus et al., 2003; Fei-Fei et al., 2004; Leibe et al., 2004; Sivic et al., 2005; Shotton et al., 2005; Opelt et al., 2008; Ferrari et al., 2009). This leads to an interesting confusion that can occur with 2-D spatial representations—changes in the appearance of model geometry encode both structural and view information. Rather than worry about resolving this ambiguity, our geometry model is in three-dimensions and separately represents the camera capturing views of it. Thus there is a clear separation between changes in geometry and view. Two-dimensional representations have tried to avoid the problem of not knowing the camera by using features that are somewhat view invariant (Berg and Malik, 2001; Belongie and Malik, 2001; Lowe, 2004; Kadir et al., 2004; Ferrari et al., 2008). But they are still limited by feature representations that exist only in the image space of two dimensions.

There are other reasons for pursing three-dimensional models over view-based representations. The three-dimensional structure of an object is often closely linked to what it is used for or what it is doing. For example, a structure with four supporting blocks under a larger flat block, such as a table, is good for setting things on. Understanding the semantics of this structure can lead to learning about other structures by recognizing similarities. For example, if a vision system recognizes the shared structure between a table

and chair, it should also learn that chairs are useful to support things. In the biological domain form is also frequently tied to function. A certain pattern of growth of one species of fungus can indicate that its objective is feeding and reproduction, while another spatial arrangement can imply conservation of energy. Understanding the structure of an object in the three-dimensional world that it exists can enable us to see how it interacts with the world, or how it can be used to interact with the world.

Another advantage of 3-D models is the capacity for quantification. Understanding the size and volume of an object in the world, requires a representation that captures quantitative information beyond simply counting pixels in the image plane. A 3-D representation enables computing exactly how large an object is in the world. We can also extract relative information about object parts, like angles between part attachments or ratios of sizes. This has benefits that extend to fields of biological research, where images are often quantitatively analyzed for experimental purposes. Biologists are frequently interested in how a change in experimental protocol affects the physiological or structural properties of a specimen. With automated structural analysis in three-dimensions, we can count, label, and quantify specimens independent of their view. We can also learn models that quantitatively identify features or characteristics of a specimen that place it in a hierarchy of species. In effect, we could build a tool to do automatic species classification. With view-based approaches, the amount of ambiguity would be too great to discriminate species-specific branching angles, since the branches could be away from the imaging plane for a particular view.

Three-dimensional representations have clear benefits in interactive environments and robotics applications. Inferred 3-D object models visualized in virtual environments facilitate interactive shape and function exploration. In robotics, 3-D models can be utilized to understand where in the real world an object is located. Although the exact function of an object may not be known, just understanding where the object is positioned and its 3-D shape enables manipulating or avoiding it. Further, if a robot has the ability to recognize parts of the object, it could then manipulate those parts as well.

Modeling objects in three-dimensions with our approach is not without challenges, however. A major problem is how to extract three-dimensional information from single

two-dimensional images. Consider the common case that the orientation and position of the camera creating the images is unknown. This leads to the under-constrained problem of trying to find the correct configuration for a camera and pose of an object producing the observed image. Starting from a single view image of an object, this can be difficult; however, cues from perspective projection of parallel lines and simultaneously fitting 3-D geometric models of structure aid in the recovery process by constraining the possible views of the camera.

1.2 Objects as assemblages of parts

We propose a strong representation for objects that comprises three-dimensional geometric primitives assembled into a connected structure. We believe that objects can be represented by a set of 3-D geometric parts, and that we can learn assemblages of these parts from images. We have chosen geometric primitives such as blocks, cylinders, and ellipsoids to represent object parts. An advantage of representing objects with a set of geometric primitives is that they often correspond to what humans might identify as parts. Another reason a part-based representation might be helpful is in identifying certain substructures of the object category; our man-made objects and biological specimens have substructures that closely resemble our set of chosen primitives. Representing basic parts of an object also allows part grouping at higher levels, so we can learn configurations of these primitives and their shared substructures across categories. Finally, having a part-based level of detail for structure representation enables quantification of a subset or all of an object.

Three-dimensional alternative representations that do not specifically model parts include organization of 2-D feature patches and their relative transformations approximating foreshortening (Savarese and Fei-Fei, 2007, 2008), space carved 3-D object volumes (Hoiem et al., 2007), or even extracted three-dimensional surfaces (Amenta and Bern, 1999; Levin, 2003). Although the representation in these approaches is three-dimensional, they lack the ability to identify quantitative information about the detected parts. Furthermore, Savarese and Fei-Fei (2007, 2008) are unable to localize parts of the

object because their representation is only a collection of surface patches with relative orientations in 3-D. Learning about shared structure in volume-based approaches is not possible due to the individual parts not being identified. In contrast, our approach represents objects with 3-D parts that can be identified across object categories and localized in three dimensions.

The idea of object representation as a set of primitive parts is similar to recognition-by-components proposed by Biederman (1987). In our case, we assume that the available set of geometric primitives comprise easily identifiable parts of an object, e.g., blocks for legs of tables and chairs. While the pairings of blocks, ellipsoids, and cylinders to parts may not always be the same labeling a human might identify, it is a starting point to reasoning about substructure within an object class. Another characteristic we share with recognition-by-components is that the number of objects we recognize is limited by our vocabulary of primitives. An idea for extending this work is to learn the vocabulary of parts from images based on criteria such as which parts offer the most discriminative power.

Parameterizing our structure model on the part-level enables capturing information about structure variability with respect to parts. For example, we could learn from a category model of chairs that the structural variation is primarily divided among those with armrests and those without. Or for tables, the number of legs under the table-top or the shape of the table-top—that it is rounded or square. Having a part-based representation enables an understanding of this variability. In the biological context, we learn about frequencies of particular substructures. This could be particularly useful if one of the substructures is identified as being important for reproduction of a specimen.

A part representation in three-dimensions is much more helpful in an interactive or robotics setting than a simple volume or surface reconstruction. The robot or machine has a good starting point for knowing where on an object to grasp to manipulate it. Knowing which pieces of an object can be independently controlled is useful in many cases, for instance opening the door of a car or pressing buttons on a phone. With a volumetric approach we would not be able to identify these individual parts for manipulation. We can further ascribe specific knowledge of part function to an object, for example the support

offered by armrests to a chair, or which substructures are reproductive to a biological specimen.

The primary difficulty with this approach is learning a meaningful assemblage of primitives from our vocabulary that represents an object class. An easier approach would be to hand craft assemblages or rely on external object models. Constructing a 3-D model for each object in the world is not feasible, however, so we aim to learn them from images. The challenge is that our parameter space over object parts is large, of unknown dimension, and has many local minima in the energy function used for learning due to ambiguity. To quantify uncertainty about which object model best fits a set of images, we develop a Bayesian statistical framework over parameters and a trans-dimensional sampling algorithm for inference.

1.3 Stochastic generative model

We consider the observed image data of an object as generated by a statistical model for an object category and the camera viewing it. The model has multiple generative levels spanning a general category representation to detectable image features. In the most general case (Chapter 5) we first generate an object category. The model for an object category contains statistical information for an object that captures its variance across images. This includes both structural and appearance information. In this way we represent the variation that an object category has in its topological structure and visual appearance.

Conditioned on a sampled category, we sample an instance of the object from the category statistics. The generated instance includes the specific size, shape, and pose of an object. We also sample a camera capturing the specific view of the object in an image. The process for generating an instance of the biological and man-made structure models is similar, but each has different shape primitives. For example, in Chapters 2 and 3 where we represent biological structure, the generated shapes include ellipsoids and cylinders; in Chapters 4 and 5 where we learn man-made structures like furniture, the generated shapes are blocks. In both cases the size, position, and attachment points are generated

from the category model.

Once we have generated instances of the object model and camera capturing the image, we can generate projected detectable image features. There are many different types of image features one could model. In our case, we investigated using projected object contours to generate edge points in the image and the surface of the object to generate foreground. For the edge points, we say that, conditioned on the object and camera models, each edge point is independently generated by a point on the object contour with Gaussian error in its contour distance orientation. In this way we generate features as they might be detected by a gradient-based edge detector. The generative process for other features are modeled in the same fashion.

Given a set of images containing objects from a category, we would like to infer, or learn, the best fitting set of parameters under our model for the category. To do this we will reverse the generative process through a Bayesian posterior distribution over the parameters conditioned on images. By combining the distributions at each level of our statistical generative model and some relatively uninformative prior information. we can create a process for statistical inference of our model under observed images. We can then use Bayesian inference to find the object category parameters and particular instances that fit the data well.

A statistical model for 3-D objects is powerful, but learning its parameters from data is challenging. We present Markov chain Monte Carlo sampling algorithms for maximum posterior estimates. The basic idea of MCMC sampling is to construct a chain of memoryless state transitions that iteratively generate random samples from a distribution, and that over time converges to the target distribution. In our application, once the convergence takes place, the subsequent samples are likely close to the maximum of the posterior. This is because we construct a density function with most of its mass on a small region of parameter space—where our object model visually matches well with the image data. We then use the generated samples with highest probability as our best fitting model.

Using a generative representation has a number of advantages. We have a probabilistic way to detect an object in an image and determine how good that detection is. From an instance that is fit well to the data, we can further say that some part of the object is

missing from the detection. So if part of the structure an image is not observed, we can hallucinate its presence. This provides a simple and effective way to deal with occlusion. We can also utilize the statistics over object structure in our model, enabling formal statements about estimates of appearance and structure variation within an object category. Finally, since we have a probability distribution over model parameters, we could include higher-level inference about objects. We could, for example, apply a risk function, which would be particularly useful in the biological imaging context.

In Chapters 2 and 3 we introduce a 3-D biological structure model for microscopic fungus of the genus *Alternaria* and statistical inference algorithms that fit instances of it to microscope image data. The details of these chapters are based on our previous work published in Schlecht et al. (2006) and Schlecht et al. (2007). In Chapters 4 and 5 we develop similar ideas but for more general object structure, which we base on an expansion of our previously published work in Schlecht and Barnard (2009a) and Schlecht and Barnard (2009b). Additionally, in Chapter 5 we show how to infer not only instances, but the object category as well. What is developed in these last chapters for general structure could be applied to our earlier biological structure model as well. Together, these chapters present our approach for using a statistical generative model and three-dimensional object representations.

1.4 Related work

The following works in the literature have served as inspiration for our ideas and approach presented in this thesis. We describe their relevance in groups based on how related they are to model-based vision (1.4.1), view-based vision (1.4.2), grammars and topologies (1.4.3), biological structure (1.4.4), and statistical inference (1.4.5).

1.4.1 Model-based vision

Researchers have represented object structure with three-dimensional models in vision for a long time. For much of that time, however, the model was assumed to be known. These approaches are often referred to as model-based vision. One of the major challenges

encountered in model-based vision is deciding the type of 3-D model to use for representing objects and how to match it to projections in an image. Clowes (1971) was among the first to explain two-dimensional line drawings of projected object contours with a three-dimensional representation. Their model comprises planar, right-angled constructs and matches corners and junctions of drawn lines with their contour correspondences in three-dimensions. Sugihara (1984) proved that it is generally possible to represent 2-D line drawings as projections of 3-D polyhedral scenes. Winston (1975) developed an idea for connecting three-dimensional wire-frame blocks into a representation for simple objects projected as line drawings.

Given a three-dimensional representation for an object, much work has been done in model-based vision to detect its pose in an image. For example, Binford (1971) and Brooks (1981) propose using a known model comprising generalized cylinders to capture object structure in 3-D and match its views to 2-D images. Models comprising more sophisticated parts, such as superquadrics, have been investigated as well (Pentland, 1987, 1990). In the latter case, however, fitting is done to range data, significantly constraining the pose estimation. Lowe (1987, 1991) shows how to linearize the projection of a known 3-D model into the image plane and apply Newton's method for matching contours in the projected model to edges in a single image. Lowe (1991) also describes how to fit a more complicated, but known, 3-D model with parametrized parts to single 2-D images using the same edge matching and gradient descent algorithm. If a few correspondence points are known between a given 3-D model and its projected image, Huttenlocher and Ullman (1990) give a closed form solution for the transformation that maps one set of points to another.

Biederman (1987) was among the first to propose a formal theory that the human vision system represents objects as compositions of 3-D geometric primitives. In this view, a vision system recognizes objects by correctly assembling parts from a vocabulary of basic 3-D geometric icons, referred to as geons. For a sufficiently sized vocabulary of geons, they argue that most objects can be represented through composition, scaling, rotation, and other transformation of the primitives. The idea of recognition by component, challenged the assumption in model-based vision that a complete representation must be

known beforehand; that the model can be learned through grouping and fitting of smaller, simpler structural parts. Indeed, for our approach we are heavily influenced by this concept; that we can learn the composition of 3-D parts comprising the object structure.

In Pope and Lowe (1996) we see the transition of model-based approaches to learning structure representations of 3-D shape under different views. Their representation is not completely 3-D, however, and somewhat related to view-based approaches. They learn a codebook of sequential 3-D views constructed from a training set; each view contains statistics over appearance information for the object. They also construct a probability density for matching detected features to points in their model. This probabilistic formulation enables an understanding of variation within object classes.

1.4.2 View-based vision

More recently, vision research has moved away from explicit three-dimensional representations and focused on learning two-dimensional view-based models. The dependency of relying on a provided set of 3-D models for each object category in the world was viewed as unsustainable. Further, the complexity increase caused by three-dimensions was perceived as excessive and unnecessary for simple detection and recognition problems in the image plane. Given recent developments in pattern recognition and machine learning, the focus has instead turned to learning 2-D statistical models over view-dependent image patches and boundary fragments.

Fergus et al. (2003); Fei-Fei et al. (2003, 2004) presented an approach to learn two-dimensional object category models based on a constellation of descriptive image patches corresponding to parts. The model encodes patch appearance statistics and relative 2-D spatial statistics over part relationships. Sivic et al. (2005) uses a bag of words topic model (Hofmann, 2001; Blei et al., 2003) to learn object categories in unlabeled images. They represent categories as document topics and appearance-based features as words within a topic, creating a bag of features approach. Sudderth et al. (2005) describe a hierarchical generative model for a statistical representation of 2-D image features, parts, objects, and images. The learned parts are clusters of features that are shared across objects. Torralba et al. (2004) learn discriminative classifiers jointly for object detection that

also share features, or parts, across subsets of object categories. When multiple 2-D views of an object category are available, e.g., front and side views of a car, they further show that parts appearing in different views can be learned as shared. Leordeanu et al. (2007) and Kushal et al. (2007) learn statistics over 2-D part models, but with more emphasis on geometric constraints over spatial arrangement of parts. Leordeanu et al. (2007) focuses on pairwise geometric relationships among parts, while Kushal et al. (2007) models affine transformations relating 2-D views of image patches across an object category. Crandall and Huttenlocher (2006) present an algorithm for learning part-based models over both image appearance patches and their geometric relations. Their approach differs from others by simultaneously learning statistics over part appearance and spatial arrangement in a weakly supervised manner.

Other approaches have successfully created discriminative view-based algorithms that rely on detecting two-dimensional parts in images. Leibe et al. (2004) presents a generalized Hough voting approach to construct implicit 2-D shape models for object categories based on a codebook of distinctive image patches. Detection proceeds by matching patches in a test image to the codebook and accumulating votes for the object center. Shotton et al. (2005) uses projected object contours for detection rather than appearance information from image patches. They present an algorithm to learn a set of 2-D boundary fragments for an object class that is organized into a star pattern. When matched with detected edges in an image, the fragments identify the object center. Opelt et al. (2008) combine patch appearance and boundary fragment models into a voting framework for object detection. Local fragments and patches are matched against a learned codebook, which then vote for the object location. Dalal and Triggs (2005) shows that collecting responses of oriented gradient filters in simple histograms of overlapping image windows provide an effective means for capturing discriminative 2-D object shape.

Many of these view-based approaches rely on groups of detected 2-D features that are somewhat invariant to object scale, pose, and view (Belongie and Malik, 2001; Berg and Malik, 2001; Lowe, 2004; Kadir et al., 2004; Ferrari et al., 2008). Although some approaches have attempted to learn transformations that relate appearance of these features under different views (Kushal et al., 2007), large changes in pose or viewpoint are

typically not tolerated. This is particularly true if the change in viewpoint reveals a side of the object not encountered during training. Thus these models must be learned for each canonical view of an object category.

A key component of our approach is modeling detected image features as 2-D projections generated by our 3-D representation for object structure. For recognition we aim to use a statistical inference and recover 3-D information from 2-D image data. Recent view-based methods have shown how to recover and utilize depth information in single view images for multi-view object detection. Saxena et al. (2005) applies supervised learning to recover depth maps from difficult outdoor scenes in single view images. Hoiem et al. (2005, 2006) recovers depth information for single view images by assuming that most pixels align with one of three primary planes in the world. Based on perspective cues in image regions with similar texture, they learn which of the planes each pixel is positioned on. The estimated depth map is then used it to put object detectors into the correct perspective. Hoiem et al. (2007) proposes improving detection and recognition with a volumetric 3-D model estimated from training data for each object category. Statistics over appearance and shape are then registered onto the 3-D model for use by an object category detector under multiple views. Liebelt et al. (2008) follows a similar approach to Pope and Lowe (1996) by showing how a 3-D model can improve view-based object detection. They train a codebook for feature matching on discretized 2-D views of the model.

Another idea put forward recently for recognition under multiple views combines a statistical model over patches with their estimated pairwise three-dimensional relationships. Savarese and Fei-Fei (2007, 2008) uses a statistical representation over part appearances similar to other view-based methods, but the shape information captures 3-D viewpoint relationships with affine transformations between pairs of patches. For each patch, or part, they learn transformations to describe how it looks when viewed from an adjacent part, e.g., whether it is scaled (foreshortened), rotated, or translated. While this approach captures implicit 3-D information through patch relationships, it does not define a complete structure model that could be utilized in areas outside detection, such as robotics applications.

1.4.3 Grammars and topologies

In both of our biological and man-made object models, we represent object structure as a topology generated by as set of rules comprising a grammar. The geometric primitives in our structure models, such as ellipsoids, cylinders, and blocks, belong to the grammar vocabulary; rules for attachment define how elements of the vocabulary are used to construct topologies. Some of the inspiration for this work comes from Han and Zhu (2005), where they develop a grammar for parsing groups of object parts in an image. Their grammar vocabulary consists of 2-D rectangle groups, with rules for alignment and symmetry used to parse windows, kitchen scenes, and other 2-D block objects. Tu et al. (2005) extends the concept of image parsing to segmentation and identifying meaningful regions in an image. They develop a generative model for image composition based on a parse graph and bottom-up, discriminative detections. Zhu and Mumford (2006) pushes the idea of parsing images further into developing a grammar for images. Zhu et al. (2006) learns a grammar of Markov random fields for grouping appearance-based 2-D image patches. We have also been influenced by Tenenbaum et al. (2006); Kemp and Tenenbaum (2008), where they seek to learn topological structure automatically from data.

1.4.4 Biological structure

Algorithms that extract 3-D object structure from images have been applied in the biological domain for some time. The potential impact of structure quantification and analytical analysis has driven much exploration in this area. Pawley (1995) and Cheng et al. (1994) contain large collections of works on processing 3-D biological images. Unlike most of these, however, our approach is built upon Bayesian statistical framework. Grenander and Miller (1994) was among the first to apply a Bayesian approach to fitting 2-D structure in medical images. They demonstrate how deformable templates can be fit to mitochondria cells using an informal trans-dimensional sampler. Al-Awadhi (2001) and Al-Awadhi et al. (2004) fit geometric shapes to cartilage tissue but formalizes the trans-dimensional inference using reversible jump Markov chain Monte Carlo (Green, 1995). Song et al. (2002) applies Bayesian inference to estimate 3-D models of heart ventricles

from echocardiograms. The general form and dimensionality of their ventricle model is fixed, and they find the best fitting parameter values through a maximum a posterior estimate with the EM algorithm. Other modern attempts at extracting tubular filament structure from biological images utilize median-based filters. For example, Can et al. (1999) traces retinal veins and Can et al. (1999); Al-Kofahi et al. (2002, 2003) follow the paths of neurons in three-dimensional stacks of confocal microscope images. While our approach is similar to many of these in our choice of statistical inference, we model biological structure, specifically microscopic plants and cells, in a much different way. To extract 3-D structure information from microscopic images, we represent biological specimens as a collection of parts whose assemblage is generated by a grammar for global specimen structure (e.g., L-systems Lindenmayer, 1968, 1975).

1.4.5 Statistical inference

Our model of three-dimensional object structure has potentially many parameters for the geometric primitives and their attachments. Inference of our Bayesian posterior over the parameters is challenging analytically, so we pursue a Markov chain Monte Carlo sampling strategy. Tierney (1994) first described an MCMC sampling algorithm for exploration of Bayesian posterior distributions. To switch between parameter subspaces of differing dimensionality, Green (1995, 2003) proposed a reversible-jump MCMC sampling algorithm. The algorithm guarantees converges to the target density function and draws samples of varying dimensions, which is shown useful for model selection. Forsyth et al. (2001) and Andrieu et al. (2001) review MCMC approaches for Bayesian posterior inference with applications to vision problems. Kaess et al. (2004) details a reversible-jump MCMC application to the vision problem of fitting piece-wise curves to 2-D shape models. Unfortunately, MCMC sampling convergence to the target distribution can sometimes require long runs of the sampler. To speed this process up, Zhu et al. (2000); Tu et al. (2002) propose a data-driven MCMC algorithm that preprocesses the data with respect to parameter state-space. This enables generating acceptable proposals in the sampling algorithm with much higher frequency and accelerates convergence. In our approach to modeling 3-D structure, we create a reversible-jump MCMC sampling algorithm that uti-

lizes a data-driven process for increased sampler convergence rate and inference that is more accurate after a reasonable amount of time than standard proposal distributions.

CHAPTER 2

Inferring 3-D Biological Structure and Microscope Models

2.1 Introduction

In this chapter we detail a new method for automatically detecting and quantifying three-dimensional structure elements of biological specimen that counteracts the blurring effects of a microscopic imaging system. Quantifying the structure of cells and organisms is important for many biological experiments, but this process can be expensive and time consuming when done manually. A method to automatically detect, quantify, and classify the three-dimensional structure of specimen in microscopic images would enable high-throughput data analysis, improved experimental efficiency, and possibly lead to increased frequency of scientific discoveries.

The challenges in creating such an algorithm for analyzing microscopic data lie not only in the detection of structure, but in understanding the image formation process of the microscope. Depending on the type of microscope used, images of a specimen under view may contain a significant amount of blur from out-of-focus regions. In a standard compound microscope with high magnification, this is a result of a shallow depth of field. Thus, the optical system of a microscope can make accurate localization of detected structure in images more difficult.

To detect and quantify the structure of biological specimen in microscopic images, we propose a model that stochastically generates the observed data. A set of 3-D geometrical objects model the structure of the specimen under study, and a theoretical impulse response of the microscope models the optical system. Using Bayesian statistical inference and Markov chain Monte Carlo sampling, we fit both of these models simultaneously to microscopic image data with mutual benefit; information learned from inferred specimen structure is used to learn model parameters of the imaging system and vice-versa.

The impulse response, or point spread function, of the microscope's optical system

blurs the observed image data. Learning a model of the point spread function (PSF) enables an understanding of the image formation process in the microscope. This permits us to hypothesize unblurred images of the specimen and obtain a more accurate fit to its structure. Moreover, using a model to learn the PSF from image data facilitates inferring structure that has been imaged under a range of optical systems.

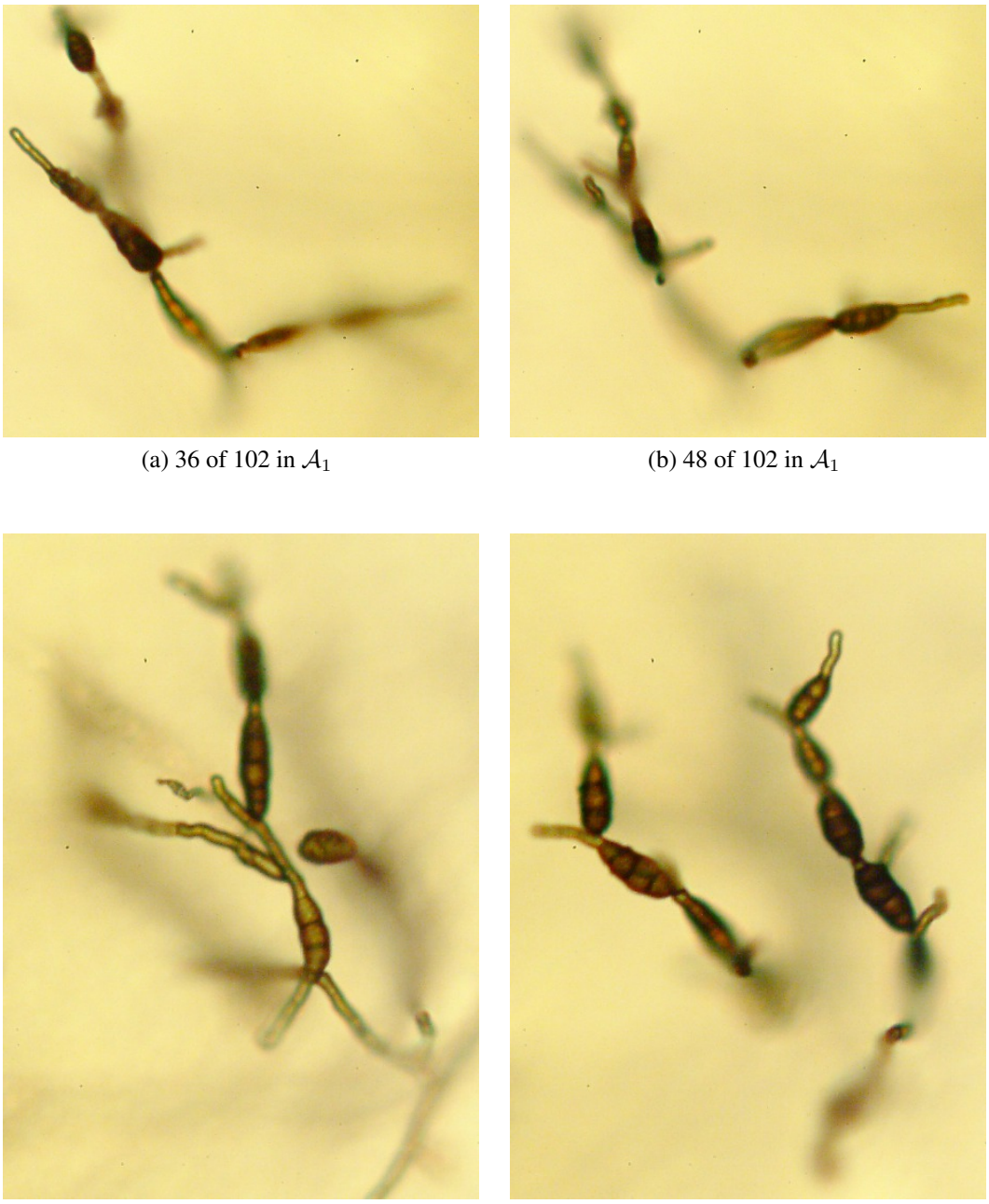
The effects of a PSF are shared by all images a microscope captures. Although in this work we infer structure and PSF models from image data sets independently, we could learn the PSF in conjunction with fitting structure in multiple data sets at once. When sufficiently fit, our learned model of the PSF could be used to detect structure in future data sets more robustly and with less computation.

2.1.1 Scientific motivation

Understanding the morphological structure of an object by modeling it and automatically fitting it to data yields valuable quantitative information that creates further insight into function. For a biologist interested in analyzing microscopic specimen, automatically inferred structure enables a high-throughput data analysis system to improve experimental efficiency and increase the frequency of scientific discoveries. Moreover, the function of a specimen is often captured in other modes of data, such as gene expression data, providing opportunities to learn a coupling with structural information learned in the fitting process. Multi-modal data linking can reveal new functional information that was not previously available due to limitations in manual structure quantification. Finally, our model is of the complete structure, and once fit to data, can be used for visualization in virtual environments and three-dimensional printing for tactile exploration.

The data used in this research are 3-D images of *Alternaria*, a genus of fungus, captured by a standard brightfield transmitted-light microscope. The images are 3-D in the sense that the mycologist who captured them continuously imaged the specimen while increasing the focal depth of the microscope, a process commonly referred to as 3-D microscopy. Figure 2.1 shows images from two of these sets, $\mathcal{A}_1$ and $\mathcal{A}_2$. Notice the significant blur in the images, a result of the optics in the transmitted-light microscope.

The general form of *Alternaria* is tree-like with species-dependent branching pat-



(a) 36 of 102 in $\mathcal{A}_1$

(b) 48 of 102 in $\mathcal{A}_1$

(c) 13 of 82 in $\mathcal{A}_2$

(d) 53 of 82 in $\mathcal{A}_2$

Figure 2.1: Images from *Alternaria* 3-D data sets $\mathcal{A}_1$ and $\mathcal{A}_2$. In each image, the point-spread function of the brightfield transmitted-light microscope generated blur from nearby focal planes. *Alternaria* comprises two primary substructures: elliptical shaped spores used for reproduction and cylindrical shaped hyphae involved in nourishment collection.

terns. It comprises tubular filaments, known as hyphae, and ellipsoid-shaped reproductive spores that are darkly pigmented. Branching typically occurs as a bifurcation of the hyphae, but it may occur in the spores as well. Species of *Alternaria* are frequently found in soil and organic debris and are estimated to contribute to 25-50% of agricultural spoilage (Wilson and Wisniewski, 1994). They are also among the most common potent airborne allergens (Wilken-Jensen and Gravesen, 1984) and one of the most prodigious producers of toxic chemicals, some of which have been linked to forms of cancer (Gui-Ting et al., 1992). For these reasons, *Alternaria* is heavily analyzed by mycologists in order to better understand its functionality and discover methods that reduce its effects.

Following a discussion of related work in Section 2.2, we present a stochastically generative approach in Section 2.3. We model the reproductive spores of *Alternaria* with independent ellipsoids and the 3-D microscopic imaging system with a parameterized point spread function. In Section 2.4 we describe the process for Bayesian statistical inference of our models. Sections 2.5 and 2.6 present our sampling algorithm for inference of both the structure and imaging models simultaneously from data. Finally, in section 2.7 we give some results of this inference process. In Chapter 3 we continue to build on the fundamentals developed in this chapter and present a more complete structure model using a stochastic grammar and methods of inference from the same images used here.

2.2 Related work

Extracting biological structure information from microscopic images is important for many scientific inquiries. Collecting three-dimensional stacks of images has been common among biologists for some time, and much effort has been made in developing methods to process, quantify, and visualize this data (Section 1.4.4; Samarabandu et al., 1994; Boxall et al., 1994).

More recent approaches have focused on Bayesian statistical inference of known biological structure models from images (Grenander and Miller, 1994; Song et al., 2002). In most of these cases, however, the model representing structure was given beforehand. In an approach perhaps most closely related to ours, Al-Awadhi et al. (2004) show that a

Bayesian model for a variable number of ellipses could be inferred from images of cartilage cells formed under a confocal microscope. This analysis enabled quantification of the cells in the image and their variation. However, because the model is in two dimensions and the structure of the cells exists in three, further quantitative analysis was not possible. To address this issue, a 3-D model for fitting ellipsoids to the cells in a stack of images was proposed by Al-Awadhi (2001), but the method of inference resulted in poor performance.

Many of the images captured by microscopes for analysis are from expensive confocal microscopes. We pursue a solution for images captured by standard brightfield microscopes. While less expensive, brightfield microscopes pose a greater challenge in analysis; they generate substantial blur in the imaging plane. The optical system of a confocal microscope attempts to minimize the aberrations, flare, and blurring potential of its PSF, thus reducing blur in its images (Pawley, 1995; Chen et al., 1995; Conn, 1999; Webb, 1999; Kong et al., 1999). Previous studies in statistical inference of structure modeled the PSF of a confocal microscope with a Gaussian function, but the parameterization was obtained by preliminary, manual analysis of the image data (Al-Awadhi et al., 2004; Al-Awadhi, 2001). Furthermore, because of the minimal blur in the confocal data, their PSF model was less critical to a good fit of the structure than it would have been under a standard transmitted-light microscope, such as the one imaging our data.

Depending on the immersion medium and microscope, a PSF can be measured and estimated by imaging a tiny bead of material, such as oil or latex. The resulting measurements can subsequently be used to deconvolve images formed by the microscope (Shaw and Rawlins, 1991). However, performing the measurements can be a very time consuming and tedious process and the results are microscope dependent.

Efforts have been made to learn the structure of a PSF without direct measurement for the sole purpose of image restoration (Holmes, 1989, 1992; Conchello et al., 1994; Conchello, 1998; Markham and Conchello, 2001; Carasso, 2001; Preza and Conchello, 2004). Results of this work have been somewhat successful. However, the images were often captured under a confocal microscope. It has not been shown that these methods can effectively deconvolve images from a brightfield transmitted-light microscope. Fur-

thermore, eliminating blur in the images prior to analysis is a loss of information. By modeling this blur and understanding it, we can get more accurate fits of our structure model to data.

2.3 Structure and imaging models

Our generative model for the 3-D microscopic image data comprises a model for *Alternaria* spores, the PSF of the imaging system, and the background light intensity of the brightfield microscope. Although the spores are linked together in a contiguous shape, we model them as independent sub-structures to, at least initially, simplify the problem and develop our general approach. What follows is a description of our structure and imaging models.

2.3.1 Spore structure

From Figure 2.1 we observe that *Alternaria* spores are fairly elliptical in shape and darkly pigmented. Since we have a three-dimensional stack of images and are interested in modeling the structure of *Alternaria* in 3-D, we represent them as ellipsoids with varying levels of opacity. Thus, the i-th spore in the structure model has parameters for position, size, rotation, and opacity

$$\boldsymbol{s}_i = (p_o, a, b, c, \varphi, \vartheta, \psi, \lambda) \ . \tag{2.1}$$

The position p_o gives the center of the spore ellipsoid in a 3-D imaging window $\mathcal{W}$. The size of the spore is specified by $a, b, c \in \mathbb{R}^{3+}$, which are the semi-axis lengths in the representative ellipsoid. The orientation parameters φ, ϑ, ψ are Euler rotation angles that all range over $[0, \pi]$, due to symmetry in the ellipsoid. Finally, $\lambda \in [0, 1]$ represents the average opacity of a spore rendered as a filled ellipsoid in the image data.

As we describe our model for *Alternaria* spores and microscope imaging system, we define a parameter space over which we will infer the model from data. We let uppercase symbols represent parameter spaces and lowercase symbols represent elements of that space. Denote the first part of the space containing all parameterizations of the i-th spore

as $\mathbf{S}_i$, and let the space for n spores be $\mathbf{\Psi}^{(n)} = \mathbf{S}_1 \times \cdots \times \mathbf{S}_n$. Then an ordered set of n spores, $\boldsymbol{\psi}^{(n)} \in \mathbf{\Psi}^{(n)}$, is given by

$$\boldsymbol{\psi}^{(n)} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \ . \tag{2.2}$$

2.3.2 Imaging system

The images formed under a brightfield microscope exhibit a substantial amount of blur, as can be seen in Figure 2.1. This is due to the shallow depth of field caused by the high magnification of the optical system. In order to perform a good fit of a model and localize the structure of *Alternaria* in the microscopic image data, it is important for the PSF model to accommodate this blurring effect.

The image formation process in a microscope is a convolution of the clear, unobserved 3-D image with the point spread function, or impulse response, of the imaging system. The PSF is the 3-D response $h(x, y, z)$ of a point source of light in the system. Using constraints from previous empirical observations (Shaw and Rawlins, 1991), we introduce a model for the PSF of a transmitted light microscope.

Let $\tilde{h}(\cdot)$ be a model that approximates the actual PSF in the imaging system. The x, y-plane in the space containing the model is defined to be parallel to the focal plane and the z-axis aligned with the optical axis of the microscope. The function is defined as a sequence of weighted 2-D Gaussians, each parallel to the x, y-plane and centered on the z-axis. Thus it is symmetric about the x, y-plane and around the z-axis.

Formally, we define $\tilde{h}(\cdot)$ as a mixed function of stacked Gaussians ranging over $x, y \in \mathbb{R}^2$ and weighted along $z \in \mathbb{Z}$,

$$\tilde{h}(x, y, z) = \frac{\alpha^{|z|}}{\sqrt{2\,\pi\,(\beta\,|z| + \gamma^2)}} \ \exp\left[-\frac{(x^2 + y^2)}{2\,(\beta\,|z| + \gamma^2)}\right] \ . \tag{2.3}$$

The parameter γ gives the base variance for all the Gaussians, and β scales their distance from the x, y-plane. Thus, each Gaussian in $\tilde{h}(\cdot)$ has a variance that is linear with respect to its distance from the x, y-plane. The base α is for the geometric distribution used to weight the Gaussians.

An informal and approximate visual description of the PSF geometry in (2.3) is that of two blurry, conical volumes placed apex-to-apex at the origin of the x, y-plane. The values within the cones are weighted 2-D Gaussians parallel to the x, y-plane. Figure 2.2 shows an illustration of this description.

It is possible to define $\tilde{h}(\cdot)$ completely over $\mathbb{R}^3$ instead of letting z be an integer. In this case we could use the continuous exponential distribution over z for the weighting function. But as a practical matter, the point spread function in our model is convolved with a discrete set of images, so ranging z over $\mathbb{Z}$ in our representation is justified. Moreover, since the images themselves are composed of discrete pixels, x and y must be quantized as well.

Alternaria in the 3-D image data occupy a relatively small region of the imaging window. Hence, many pixels in the data are saturated with the intensity of light used by the brightfield microscope. We define the background intensity of the imaging system over the range $[0, 1]$ and denote it as υ.

We continue building the space over our model parameters by denoting the space of PSF parameters and background intensities with $\mathbf{\Phi}$. We further let a parameterization of an imaging model in this space be given by

$$\boldsymbol{\phi} = (\alpha, \beta, \gamma, \upsilon)\,. \tag{2.4}$$

2.3.3 Generative image model

Let $\boldsymbol{\theta}^{(n)} = (\boldsymbol{\phi},\ \boldsymbol{\psi}^{(n)})$ be an instance of the parameter space $\mathbf{\Phi} \times \mathbf{\Psi}^{(n)}$ defined over multispore and imaging system models. Then the space of potential solutions spanning all model configurations is

$$\mathbf{\Omega} = \bigcup_{n>0} n \times \mathbf{\Phi} \times \mathbf{\Psi}^{(n)}\,. \tag{2.5}$$

For any $(n, \boldsymbol{\theta}^{(n)}) \in \mathbf{\Omega}$, we generate a model-scene image stack $\mathcal{I}_\theta(i, j, k)$ by intersecting the ellipsoids in the structure model with a set of equally spaced and parallel ($z = 0$)

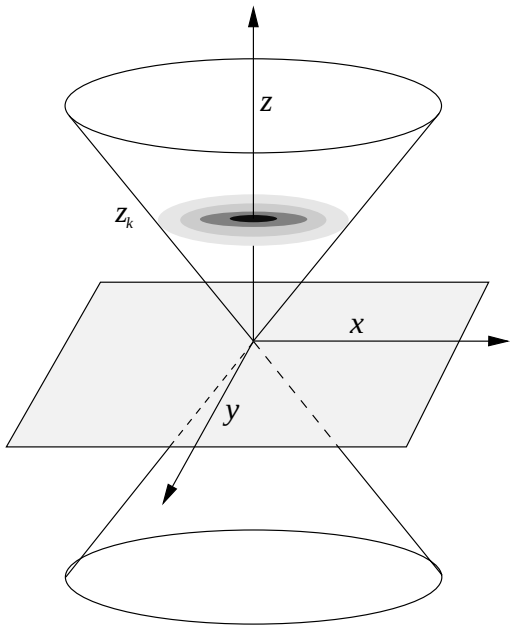


Figure 2.2: Diagram of the PSF model $\tilde{h}(x, y, z)$. The x, y-plane represents the focal plane of the microscope, and the z-axis is aligned with the optical axis. The 2-D Gaussians are stacked along the z-axis away from the focal plane in both directions, with linearly increasing variance and geometrically decreasing weight. A Gaussian at distance z_k from the focal plane is illustrated.

planes. This stack of images approximates the optical sectioning of the image data captured at stepped focal lengths under the microscope. The distance between image planes along the z-axis in $\mathcal{I}_\theta(\cdot)$ is fixed and provided by the microscope operator. The model-scene is then a hypothesis of the unobserved 3-D image data; it is what might have been observed through the microscope if the blurring effects were not apparent. The pixels comprising the model-scene are defined by the background and spore intensity parameters. Background pixels of $\mathcal{I}_\theta(\cdot)$ have the highest saturation with value υ. Pixels inside a plane intersected ellipsoid belonging to a spore with opacity λ have the value $\upsilon\,(1-\lambda)$. Figures 2.4 and 2.5 show an illustration of the optical sectioning and rendering process of the model-scene.

Conditioned on a given model-scene $\mathcal{I}_\theta(\cdot)$ and imaging system $\tilde{h}$, pixel intensities in the 3-D image data $\mathcal{I}(\cdot)$ are generated from independent Gaussians. The means and variances of these Gaussians are derived from the PSF blurred model-scene. More for-

mally, the means used to generate the independent and Gaussian distributed pixel data are defined with the convolution

$$\mu_{\mathcal{I}_\theta}(i,j,k) = \sum_{u,v,w}^{\mathcal{W}} \mathcal{I}_\theta(u,v,w)\ \hat{h}(i-u, j-v, k-w) \tag{2.6}$$

where $\hat{h}(\cdot)$ is the quantized PSF model in (2.3). We compute the pixel variance parameters from the means as

$$\sigma^2_{\mathcal{I}_\theta}(i,j,k) = c_1 \,|\, \mu_{\mathcal{I}_\theta}(i,j,k) - \upsilon \,|\, + c_2 \,|\, \mu_{\mathcal{I}_\theta}(i,j,k) - 1 \,|\,, \tag{2.7}$$

In addition to showing the optical sectioning of the model-scene, Figures 2.4 and 2.5 also display the generative process of our data with blur from the convolved point spread function.

From equation (2.6) the mean value of the i,j,k-th pixel in $\mathcal{I}(\cdot)$ can be viewed as a weighted average of the model-scene with the PSF centered at i,j,k. The constants c_1 and c_2 in (2.7) scale the variance in a linear combination of spore opacity and pixel intensity. In the first term we observe that the pigment of a spore is not uniform across its occupying pixels, and that spores with greater opacity tend to have higher variability. The second term in the variance (2.7) approximates pixel intensity variations due to Poisson noise in the imaging system; the lower the pixel intensity, the higher the likelihood of noisy photon acquisition. The scaling constants are set to values obtained by preliminary image analysis.

2.4 Bayesian statistical inference

Given a stack of *Alternaria* image data $\mathcal{I}(i,j,k)$ in the 3-D window $\mathcal{W}$, we want to find the model $\left(n, \boldsymbol{\theta}^{(n)}\right) \in \boldsymbol{\Omega}$ that best fits the data. We formulate this as a Bayesian statistical inference problem by defining a probability distribution over the model parameter space given the image data and find a maximum. Specifically, we define a posterior

$$p\left(n, \boldsymbol{\theta}^{(n)} \,|\, \mathcal{I}\right) = k_p\, L\left(\mathcal{I} \,|\, n, \boldsymbol{\theta}^{(n)}\right)\ \pi\left(n, \boldsymbol{\theta}^{(n)}\right), \tag{2.8}$$

where k_p is a normalization constant, $L(\cdot \mid \cdot)$ is the likelihood of the image data, and $\pi(\cdot)$ is the model prior.

Conditioned on our model, the independence assumption among pixels in the image data results in a product of Gaussians for the likelihood function. Using the image model means (2.6) and variances (2.7), the likelihood is defined as

$$L\left(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)}\right) = \frac{1}{\sqrt{2\pi}\,\sigma_{\mathcal{I}_\theta}} \prod_{i,j,k}^{\mathcal{W}} \exp\left\{-\frac{\left[\mathcal{I}(i,j,k) - \mu_{\mathcal{I}_\theta}(i,j,k)\right]^2}{2\sigma^2_{\mathcal{I}_\theta}}\right\}. \tag{2.9}$$

2.4.1 Spore and imaging priors

The prior over the parameter space $\boldsymbol{\Omega}$ assumes independence between the spore structures and imaging system,

$$\pi\big(n, \boldsymbol{\theta}^{(n)}\big) = \pi_\Phi(\boldsymbol{\phi})\ \pi_\Psi\big(n, \boldsymbol{\psi}^{(n)}\big)\ . \tag{2.10}$$

The parameters in the imaging system priors over $\boldsymbol{\phi}$ are represented as independent and Gaussian distributed. The hyperparameter values for these priors are chosen to have a large variance and to be fairly uninformative. For the prior over spore structure, we observe that spores in *Alternaria* generally have the same shape, opacity and count, but their position and orientation is quite varied. We integrate this information into the spore model prior as follows.

The position of a spore is set to range uniformly over the window $\mathcal{W}$ that has volume $V_\mathcal{W}$. The rotation angles are modeled as independent and uniformly distributed on $[0, \pi]$. Since the spore sizes tend to be roughly the same, with a major axis and two minor axes of similar length, we define independent Gaussians over them with means μ_a for the major axis and μ_{bc} for the two minor axes. Lastly, we model spore opacity with a truncated Gaussian over $(0, 1]$. Thus, the density function for a spore $\boldsymbol{s}_i$ is

$$f(\boldsymbol{s}_i) = f_{x,y,z}(\boldsymbol{s}_i)\, f_{a,b,c}(\boldsymbol{s}_i)\, f_{\varphi,\vartheta,\psi}(\boldsymbol{s}_i)\, f_\lambda(\boldsymbol{s}_i)\,. \tag{2.11}$$

Each of the subdensity functions over position, size, orientation, and opacity is defined

by

$$f_{x,y,z}(\boldsymbol{s}_i) = \frac{1}{V_{\mathcal{W}}} \tag{2.12}$$

$$f_{a,b,c}(\boldsymbol{s}_i) = \frac{\sigma_a^{-1}\,\sigma_{bc}^{-2}}{(2\,\pi)^{3/2}}\,\exp\left[-\frac{(a_i-\mu_a)^2}{2\,\sigma_a^2} - \frac{(b_i-\mu_{bc})^2+(c_i-\mu_{bc})^2}{2\,\sigma_{bc}^2}\right] \tag{2.13}$$

$$f_{\varphi,\vartheta,\psi}(\boldsymbol{s}_i) = \frac{1}{\pi^3} \tag{2.14}$$

$$f_{\lambda}(\boldsymbol{s}_i) = \frac{\sigma_\lambda^{-1}}{\sqrt{2\,\pi}}\,\exp\left[-\frac{(\lambda_i-\mu_\lambda)}{2\,\sigma_\lambda^2}\right]. \tag{2.15}$$

The existence of a spore in the imaging window $\mathcal{W}$ follows a Poisson process, so we define n to be Poisson distributed with intensity ν, which is the number of spores we expect to observe. For this work, the value of ν was set to 10. Finally, we restrict the interaction between spores so they do not intersect. The spore model prior is then

$$\pi_{\Psi}\big(n, \boldsymbol{\psi}^{(n)}\big) = k_\pi^n\,\frac{\nu^n\,e^{-\nu}}{n!}\,\prod_{i=1}^{n}\,\chi(\boldsymbol{s}_i \nVdash \boldsymbol{s}_{j\neq i})\;f(\boldsymbol{s}_i)\,, \tag{2.16}$$

where k_π^n is a normalization constant for the truncated subdensity functions, $\nVdash$ denotes no geometric intersection, and $\chi(\cdot)$ is the characteristic function giving 1 for true and 0 otherwise.

2.5 Sampling

Inferring the most likely model given *Alternaria* image data is a challenging task. The posterior (2.8) is a complex distribution virtually impossible to evaluate analytically or numerically. Thus, we employ Markov chain Monte Carlo (MCMC) sampling to explore the model solution space in search of a maximum under the posterior (Sokal, 1989; Neal, 1993; Andrieu et al., 2001; Liu, 2001; Bishop, 2006).

The sampler iteratively generates random, unbiased model samples from the solution space $\boldsymbol{\Omega}$. It consists of a set of moves, or Markov chain, that create new model proposals by proposing changes to parameters in a previous sample. The sampler moves fall into

one of two categories: (1) changes to a spore, the PSF, or the background; and (2) changes to the number of spores in the model. The latter are referred to as *diffusion* moves and the former *jump* moves.

At each iteration of the sampler, the m-th move is selected for execution with probability $r(\mathsf{m})$ and a new model $(n, \tilde{\boldsymbol{\theta}}^{(n)})$ is proposed. In this chapter, a uniform distribution was used for $r(\cdot)$. Depending on how likely the new model is under the posterior and to have been proposed, it is accepted or rejected. This is the Metropolis-Hastings (MH) algorithm for MCMC (Metropolis et al., 1953; Hastings, 1970), and it is used for both diffusion and jump moves. The latter are reversible-jump MCMC moves from Green (1995, 2003).

2.5.1 Diffusion moves

The diffusion moves for modifying parameters in a spore and proposing a new one are shift, size, rotate, and $\mathsf{opacity}$ modification, as well as moves to update the PSF and background parameters. We define the proposal distributions for diffusion moves by modifying the prior in (2.10). For parameters updated in a move, we replace their subdensity in the prior with a Gaussian that has means equal to corresponding parameters in the previously accepted model. The unchanged parameters have essentially a delta function as their density.

For example, the proposal distribution for randomly selecting the j-th spore in a model $\boldsymbol{\theta}^{(n)}$ and shifting its position is given by

$$q_{\mathsf{shift}}\left(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)}\right) = \frac{k^n_{\mathsf{shift}}}{n} \left[\prod_{i \neq j}^{n} \chi(\boldsymbol{s}_i \not\vdash \tilde{\boldsymbol{s}}_j)\right] \times \frac{\sigma^{-3}_{x,y,z}}{(2\,\pi)^{3/2}} \exp\left[-\frac{(\tilde{x}_j - x_j)^2 + (\tilde{y}_j - y_j)^2 + (\tilde{z}_j - z_j)^2}{2\,\sigma^2_{x,y,z}}\right], \quad (2.17)$$

where $\sigma^2_{x,y,z}$ is a small variance and k^n_{shift} is a normalization constant that is not necessary to compute, as we will see. Similarly, the proposal for changing spore opacity is

$$q_{\mathsf{opac}}\Big(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)}\Big) = \frac{k_{\mathsf{opac}}^{n}}{n}\,\frac{\sigma_{\lambda}^{-1}}{\sqrt{2\,\pi}}\,\exp\left[-\frac{(\tilde{\lambda}_j - \lambda_j)^2}{2\,\sigma_{\lambda}^{2}}\right]. \tag{2.18}$$

Since a change in spore opacity does not modify its geometry, the intersection test $\chi(\boldsymbol{s}_i \not\vdash \tilde{\boldsymbol{s}}_j)$ is excluded. The proposal distributions for other diffusion moves are similarly constructed.

Under the Metropolis-Hastings algorithm for MCMC sampling, we combine the target distribution with the proposal distribution in a ratio to define an acceptance probability. In our case the target distribution is the posterior (2.8). For the m-th diffusion move in the algorithm, the rate of acceptance is

$$\alpha_{\mathsf{m}}\Big(n, \tilde{\boldsymbol{\theta}}^{(n)}\Big) = \min\left\{1, \frac{p(n, \tilde{\boldsymbol{\theta}}^{(n)} \mid \mathcal{I})\; q_{\mathsf{m}}(\boldsymbol{\theta}^{(n)} \mid \tilde{\boldsymbol{\theta}}^{(n)})}{p(n, \boldsymbol{\theta}^{(n)} \mid \mathcal{I})\; q_{\mathsf{m}}(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)})}\right\}. \tag{2.19}$$

The definition is derived to maintain a detailed balance condition in the Markov chain, which is a sufficient condition for convergence to the posterior (Sokal, 1989; Neal, 1993).

Intuitively, the algorithm constructs a Markov chain that explores the state space in a manner that is representative of the target distribution defined over it. This is accomplished by ensuring that the probability of being in state $\boldsymbol{\theta}$ and transitioning to $\tilde{\boldsymbol{\theta}}$ is equivalent to being in state $\tilde{\boldsymbol{\theta}}$ and transitioning back to $\boldsymbol{\theta}$, i.e.,

$$p(n, \tilde{\boldsymbol{\theta}}^{(n)} \mid \mathcal{I})\; T^{(n)}(\tilde{\boldsymbol{\theta}} \mid \boldsymbol{\theta}) = p(n, \tilde{\boldsymbol{\theta}}^{(n)} \mid \mathcal{I})\; T^{(n)}(\boldsymbol{\theta} \mid \tilde{\boldsymbol{\theta}}) \tag{2.20}$$

In the case of the Metropolis-Hastings algorithm, the transition probability is $T^{(n)}(\tilde{\boldsymbol{\theta}} \mid \boldsymbol{\theta}) = q_{\mathsf{m}}(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)})\; \alpha_{\mathsf{m}}(n, \tilde{\boldsymbol{\theta}}^{(n)})$. Appendix A provides a detailed description of MCMC sampling, including the detailed balance condition, the Metropolis-Hastings algorithm, and the convergence of a Markov chain to a target distribution.

By expansion of the defined posterior and proposal distributions, many of the terms in (2.19) cancel. This includes the difficult to compute normalization constants, much of the prior, and the Gaussian proposal distributions. As an example, for a shift move of the j-th spore, we apply the proposal distribution in (2.17) and construct the simplified

acceptance probability

$$\alpha_{\mathsf{shift}}\left(n, \tilde{\boldsymbol{\theta}}^{(n)}\right) = \min\left\{1, \frac{L(\mathcal{I} \mid n, \tilde{\boldsymbol{\theta}}^{(n)})}{L(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)})} \prod_{i \neq j}^{n} \chi(\boldsymbol{s}_i \nvDash \tilde{\boldsymbol{s}}_j)\right\}. \tag{2.21}$$

In this case, the prior and most of the proposal distribution cancel. This is due to the spore position prior (2.12) being a uniform distribution and the Gaussian proposal in (2.17) having symmetric means. The intersection test is necessary because the geometry of the j-th spore is altered and could have created an overlap with another spore. Since the rotation angles of a spore also have a uniform prior (2.14), the acceptance probability for the rotation move has exactly the same construction as the shift move. The size parameters, however, have a Gaussian prior (2.13), requiring an extra factor in their acceptance

$$\alpha_{\mathsf{size}}\left(n, \tilde{\boldsymbol{\theta}}^{(n)}\right) = \min\left\{1, \frac{L(\mathcal{I} \mid n, \tilde{\boldsymbol{\theta}}^{(n)})\, f_{a,b,c}(\tilde{\boldsymbol{s}}_j)}{L(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)})\, f_{a,b,c}(\boldsymbol{s}_j)} \prod_{i \neq j}^{n} \chi(\boldsymbol{s}_i \nvDash \tilde{\boldsymbol{s}}_j)\right\}. \tag{2.22}$$

The spore opacity parameter is similarly Gaussian distributed in its prior (2.15), but it does not modify spore geometry; so the intersection test is excluded from its acceptance

$$\alpha_{\mathsf{opac}}\left(n, \tilde{\boldsymbol{\theta}}^{(n)}\right) = \min\left\{1, \frac{L(\mathcal{I} \mid n, \tilde{\boldsymbol{\theta}}^{(n)})\, f_{\lambda}(\tilde{\boldsymbol{s}}_j)}{L(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)})\, f_{\lambda}(\boldsymbol{s}_j)}\right\}. \tag{2.23}$$

The acceptance probabilities for moves manipulating the imaging system parameters $\boldsymbol{\phi}$ are all similar to (2.23), since they are Gaussian distributed and independent of the *Alternaria* structure. For this reason we omit their definitions.

2.5.2 Jump moves

The jump moves in the sampler are birth and death of a spore. In both moves, the dimensionality of the model n is modified as a spore is added to or removed from the model. For a birth move, the proposal distribution for a new spore $\tilde{\boldsymbol{s}}$ is defined as the normalized spore density (2.11) in the model prior

$$q_{\mathsf{birth}}(\tilde{\boldsymbol{s}}) = k_{\mathsf{birth}}\, f(\tilde{\boldsymbol{s}}). \tag{2.24}$$

During a death move, a spore is randomly selected for deletion, so a proposal distribution is not needed.

In order to use the MH sampling algorithm for jump moves, we redefine the acceptance probability (2.19) to properly handle trans-dimensional moves between parameter subspaces. To do this, we follow the reversible-jump MCMC formulation (Green, 1995, 2003) and match parameter dimensions using the proposal distribution (2.24) and a change of variable scaling factor to switch between density functions. For the birth move, the acceptance probability becomes

$$\alpha_{\mathsf{birth}}\left(n{+}1, \tilde{\boldsymbol{\theta}}^{(n+1)}\right) = \min\left\{1,\ \frac{p(n{+}1, \tilde{\boldsymbol{\theta}}^{(n+1)} \mid \mathcal{I})}{p(n, \boldsymbol{\theta}^{(n)} \mid \mathcal{I})}\, \frac{r(\mathsf{death})}{r(\mathsf{birth})} \times \frac{1}{q_{\mathsf{birth}}(\tilde{\boldsymbol{s}})} \left| \frac{\partial(\tilde{\boldsymbol{\theta}}^{(n+1)})}{\partial(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}})} \right| \right\}. \tag{2.25}$$

Unlike diffusion moves, the determinant of the Jacobian matrix is necessary in jump moves because we are transitioning between density functions; a unit parallelepiped may have a different density in $p^{(n)}(\cdot)\, q_{\mathsf{birth}}(\tilde{\boldsymbol{s}})$ than in $p^{(n+1)}(\cdot)$, depending on how the variables are mapped. The determinant gives the proper scaling constant for making this change of variable (Green, 1995).

In our independent spore model, we match dimensions in a birth move by mapping the proposed spore to a new position in the parameter vector. For example, the size, shape and rotation of the proposed spore $\tilde{\boldsymbol{s}}$ is directly assigned as the spore $\boldsymbol{s}_{n+1}$ in the model $\boldsymbol{\theta}^{(n+1)}$. More formally, we define a mapping that matches dimensions between parameter subspaces,

$$G : \Phi \times \Psi^{(n+1)} \rightarrow \Phi \times \Psi^{(n+1)}, \tag{2.26}$$

by assigning a parameter vector containing n spores and a proposal $\tilde{\boldsymbol{s}}$ to another parameter

vector containing $n+1$ spores, i.e.

$$G(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}}) = \tilde{\boldsymbol{\theta}}^{(n+1)} \,. \tag{2.27}$$

The mapping is defined in more detail by indexed component functions that assign input parameters to their indexed subsets

$$G_\phi : \Phi \times \Psi^{(n+1)} \rightarrow \Phi \tag{2.28}$$

$$G_1 : \Phi \times \Psi^{(n+1)} \rightarrow \Psi \tag{2.29}$$

$$\vdots$$

$$G_{n+1} : \Phi \times \Psi^{(n+1)} \rightarrow \Psi \,. \tag{2.30}$$

In our case, each of these functions simply returns its indexed set of parameters, for example $G_\phi(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}}) = \boldsymbol{\phi}$ and $G_i(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}}) = \boldsymbol{s}_i$. The determinant of the Jacobian in (2.25) then becomes

$$\left| \frac{\partial\, G(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}})}{\partial\, (\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}})} \right| = \begin{vmatrix} \frac{\partial G_\phi}{\partial \boldsymbol{\phi}} & \frac{\partial G_\phi}{\partial \boldsymbol{s}_1} & \cdots & \frac{\partial G_\phi}{\partial \boldsymbol{s}_{n+1}} \\ \frac{\partial G_1}{\partial \boldsymbol{\phi}} & \frac{\partial G_1}{\partial \boldsymbol{s}_1} & \cdots & \frac{\partial G_1}{\partial \boldsymbol{s}_{n+1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G_{n+1}}{\partial \boldsymbol{\phi}} & \frac{\partial G_{n+1}}{\partial \boldsymbol{s}_1} & \cdots & \frac{\partial G_{n+1}}{\partial \boldsymbol{s}_{n+1}} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix} . \tag{2.31}$$

Thus, the change in dimensionality is a one-to-one mapping $(\boldsymbol{\theta}^{(n)}, \tilde{\boldsymbol{s}}) \rightarrow \tilde{\boldsymbol{\theta}}^{(n+1)}$, and the determinant of the Jacobian is 1. If we had chosen instead a non-identity mapping of parameter subsets, the resulting determinant of partial derivatives would not necessarily be 1; rather, it would be a function of our parameters. For example, a mapping could be defined that assigns half of the proposed ellipsoid size to split a spore. For non-linear mappings, this function can be involved and exacting to derive. In our case, we have found that the identity mapping yields acceptable proposals.

As we observed in the case of diffusion moves, many of the terms in the acceptance

probability (2.25) cancel, such as those of the prior. This also includes the proposal probability (2.24), since it is equivalent to the spore density (2.11); the normalization constant k_{birth} additionally cancels with a factor of the prior constant k_π^{n+1}. The acceptance probability for a birth move finally reduces to

$$\alpha_{\mathsf{birth}}\left(n{+}1, \tilde{\boldsymbol{\theta}}^{(n+1)}\right) = \min\left\{1,\ \frac{\nu}{n{+}1}\,\frac{L(\mathcal{I} \mid n{+}1, \tilde{\boldsymbol{\theta}}^{(n+1)})}{L(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)})} \times \frac{r(\mathsf{death})}{r(\mathsf{birth})}\,\prod_{i=1}^{n}\chi(\boldsymbol{s}_i \not\vdash \tilde{\boldsymbol{s}})\right\}. \tag{2.32}$$

A test for intersection remains to guard against adding a new spore that overlaps with any others already in the model.

The spore death move complements the birth move by randomly selecting a spore and removing it from the model. The acceptance probability for a death move is the inverse of (2.32), but with the intersection test removed

$$\alpha_{\mathsf{death}}\left(n{-}1, \tilde{\boldsymbol{\theta}}^{(n-1)}\right) = \min\left\{1,\ \frac{n}{\nu}\,\frac{L(\mathcal{I} \mid n{-}1, \tilde{\boldsymbol{\theta}}^{(n-1)})}{L(\mathcal{I} \mid n, \boldsymbol{\theta}^{(n)})}\,\frac{r(\mathsf{birth})}{r(\mathsf{death})}\right\}. \tag{2.33}$$

As with the diffusion moves, the jump move acceptance probabilities maintain the detailed balance condition (Green, 1995). Thus, the posterior will be the stationary distribution of the trans-dimensional Markov chain followed by the sampler.

2.6 Data-driven sampling

Unfortunately, the MCMC sampler described here suffers from a common problem many other samplers share—the amount of time required for burn-in, the sampler converge rate, is excessively long. In our case, this is primarily due to the uniform prior over spore position and orientation; birth proposals at uniformly random locations in the image window have a high rejection rate. This causes an increase in the number of iterations

required for the sampler to converge. In short, it takes the sampler a large number of iterations to generate highly likely sets of spores given some *Alternaria* image data. We ameliorate this problem by improving the birth proposals with preliminary data analysis to construct a more informative proposal distribution, so called data-driven MCMC (Tu et al., 2005, 2002; Zhu et al., 2000).

The basic idea of DD-MCMC is to apply some preliminary image analysis to the input data and construct more informative proposals for the sampler. For instance, if the goal is to fit circles in an image using an MCMC sampler, a good series of proposals is created by first doing edge detection on the images, then applying Hough voting to build a density function over possible circles in the image. The votes in the Hough accumulator are then adapted as a proposal distribution over circles for the sampler. This is similar to our situation except that, instead of searching for circles in individual images, 3-D ellipsoids are detected in a stack of *Alternaria* image data.

The current birth proposal (2.24) is based on the relatively uninformative prior distribution over spores. In general, random sampling from the prior makes for a reasonable proposal during birth moves. The uniform nature of our prior over spore position and orientation, however, may require a large number of proposals to generate an acceptable spore. The effect of this is that the sampler consumes most of its time rejecting poor proposals for birth and death of spores; that is, the rejection rate for jump moves is much higher than for diffusion moves. Thus, improving the proposal acceptance rate of the jump moves will speed up the convergence of the algorithm.

Our replacement proposal distribution for birth moves is generated directly from the *Alternaria* image data. We apply surface point detection and a Hough transform for ellipsoids to the data. This gives us a coarse estimate of parameterized spores in the data. The estimates are subsequently collected into a Hough vote accumulator for ellipsoids, which is normalized and used as the proposal distribution for birth moves. Exact detections of spores in the data are not necessary because the diffusion moves of the sampler will perfect the fit of the newly birthed ellipsoids; they only need to be rough approximations that on average increase the model posterior probability. What follows is a description of the surface point detector used to detect the estimates of spores in the data and how we

combine those estimates into a probability distribution over spores.

2.6.1 Surface point detector

The surface detector is similar to the standard two-dimensional Canny edge detection algorithm (Canny, 1986) but extended to three-dimensions. We use the gradient of a 3-D Gaussian for convolution with the image data. The result of this convolution is set of 3-D gradient vectors indexed by pixel position, comprising a gradient map. Following the approach outlined for edge detection in Forsyth and Ponce (2002), we apply non-maximal suppression and hysteresis to the 3-D gradient map and obtain estimated surface points.

Using a derivative of Gaussian convolution filter results in thick edges when applying a threshold to gradient magnitudes. The idea of non-maximal suppression is to replace several nearby edge point detections along a thick edge with one that has the largest gradient magnitude. We identify these points by tracing paths through the gradient map. When a locally maximum magnitude is achieved in a particular direction, its location is identified as an edge point if the magnitude exceeds a threshold.

Once non-maximal detections are removed, we again trace the gradient map, but this time in direction perpendicular to the gradient in order to follow edge contours. The tracing is initiated at the local maximum edge points found in the previous step. The contours are followed until the gradient drops below another, lower threshold. This strategy is often referred to as hysteresis.

As an added feature of our surface point detection algorithm, we can approximate the surface of *Alternaria* in the images for visualization. While it may not be as crisp as a visualization generated from our geometric model, it is still instructive. The surface reconstruction is accomplished as follows: for each surface point, define a small surface patch, e.g., a polygon, at the position of the detected surface point and orient the patch with a normal vector given by the gradient of the surface point. Examples of reconstructed *Alternaria* surfaces are given in Figure 2.3.

A good surface reconstruction could be used to estimate ellipsoids instead of relying on sparsely detected points. Our approach for reconstruction, however, is too naive to produce anything beyond a simple visualization; the reconstruction comprises a set of

oriented surface patches, whose size and shape are independently constructed. Further, we ignore local cues about surface point density and local curvature, both potentially useful for improving a reconstruction. In Appendix B we discuss other more advanced techniques for surface reconstruction. There we also address ideas for utilizing reconstructed surfaces in our data-driven ellipsoid detector and as a replacement for the blurry microscope images.

2.6.2 Ellipsoid estimator

A Hough transform is used to find ellipsoids from the detected surface points. Due to the relatively high dimensionality of an ellipsoid, we use a very coarse quantization for the Hough transform. We further simplify computation and reduce the number of parameters defining a spore by assuming equal minor axes in an ellipsoid. Although this results in imprecise estimates of spores in the data, it reduces the size and complexity of the Hough transform. Furthermore, coarse estimates are tolerable in our application because sampler diffusion moves will perfect the fit of proposed spores.

We construct the Hough transform in the standard way. For each detected surface point, we iterate over the quantized spores at each surface point and increment a counter in the Hough accumulator $\mathcal{H}$. In this way, the accumulator defines a discrete density function indexed by quantized parameterizations of ellipsoids. We utilize this density as a proposal distribution in the sampler birth move. The improved data-driven birth proposal is then redefined as

$$q_{\mathsf{birth}}(\tilde{\boldsymbol{s}}) = k_{\mathsf{birth}}\, f_{\mathcal{H}}(\tilde{\boldsymbol{s}} \mid \bar{\boldsymbol{s}})\, \mathcal{H}(\bar{\boldsymbol{s}})\,, \tag{2.34}$$

where $\bar{\boldsymbol{s}}$ is the spore in the Hough accumulator that acts as the mean value for $\tilde{\boldsymbol{s}}$. The density function $f_{\mathcal{H}}(\cdot)$ is then similar to the prior (2.24), except that all parameters are Gaussian with means given by $\bar{\boldsymbol{s}}$. Finally, the counts in the accumulator are included in the normalization constant k_{birth}. In practice, however, this constant does not need to be computed as it will cancel in the Metropolis-Hastings acceptance step.

We generate a spore from the data-driven proposal by first sampling an ellipsoid from

(a) $\mathcal{A}_1$ surface points

(b) $\mathcal{A}_2$ surface points

Figure 2.3: Rendering of detected surface points in the *Alternaria* data sets $\mathcal{A}_1$ (2.1a-b) and $\mathcal{A}_2$ (2.1c-d). The rendering was created by drawing polygons at each surface point with normals given by the gradient direction from the detection algorithm in Section 2.6.1.

the Hough accumulator, then conditionally generating a random sample from the Gaussian spore density. Sampling from the conditional Gaussian density is straightforward. To generate samples from the Hough accumulator, we use rejection sampling (Bishop, 2006; Andrieu et al., 2001). Specifically, we denote the number of bins in the accumulator with $|\mathcal{H}|$ and let its maximum be

$$\mathcal{H}_{\max} = \max \left\{ \mathcal{H}(\boldsymbol{s}_i) \right\}_{i=1}^{|\mathcal{H}|} . \tag{2.35}$$

Then samples can be generated from the Hough accumulator with uniform distributions over integers as follows:

1. Sample $u \sim \mathcal{U}(0,\, \mathcal{H}_{\max})$
2. Sample $i \sim \mathcal{U}(1,\, |\mathcal{H}|)$
3. If $\mathcal{H}(\boldsymbol{s}_i) \geq u$, propose $\boldsymbol{s}_i$; otherwise repeat.

Although the data-driven birth proposal is an improvement, it is still possible some spores were not properly estimated in the Hough voting process due to accumulator quantization and image blur[1]. Thus we also include the previously defined uniform birth move (2.24) in the reversible-jump sampler to allow for proposing the missed detections.

2.7 Results

We evaluated the effectiveness of the model sampler on *Alternaria* image sets $\mathcal{A}_1$ and $\mathcal{A}_2$. In addition, we tested the sampler on synthetic spore data to obtain a comparative measure for its performance on ideal data.

2.7.1 Synthetic Data Evaluation

The synthetic data were randomly generated from our model of the imaging system and spores. We created ten data sets $\mathcal{S}_1, \ldots, \mathcal{S}_{10}$ and optically sectioned them into 80 im-

[1]The data-driven Hough voting for ellipsoids is a preprocessing step, so it does not utilize information from our PSF model.

ages of size 300×300 pixels. Each set contained 10 randomly generated spores. Three-dimensional visualizations of two data sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are given in Figures 2.4a and 2.5a. The image sets were blurred with a parameterization of the PSF model in (2.3); random Poisson noise was added as well. Examples from both sets are given in Figures 2.4b-c and 2.5b-c.

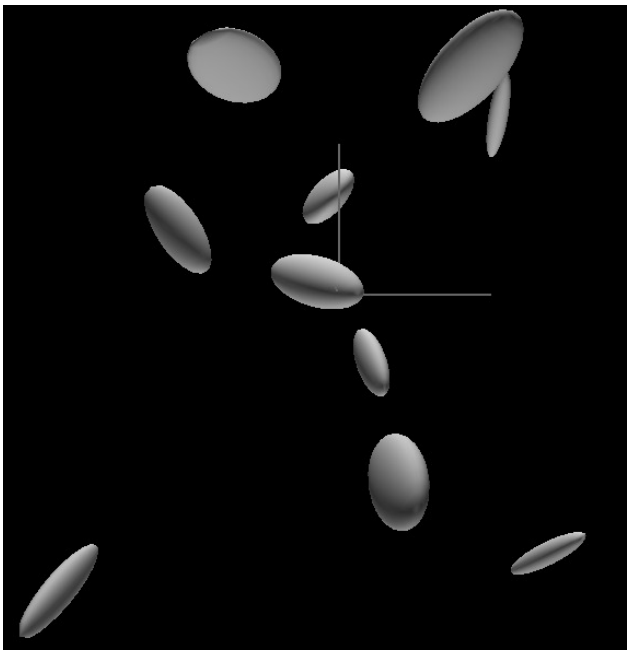
We speed up the inference time required for a good fit by running the sampler on multiple resolutions of the image data. The time required for computing our likelihood function (2.9) depends on the number of pixels in the data, so decreasing the resolution in the image stack can directly speed up the likelihood computation. Although the lower resolution images have less information to fit structure to, they quickly yield a rough fit that can be used as initialization for a sampler running at higher resolution, which further fine-tunes the inferred model.

The MCMC sampler was run for 3000 iterations at a resolution of 20%, followed by 1000 iterations at 50%. This process was repeated four times on all the data sets using a different random seed each time. On average we correctly fit 8 spores to the data sets. The results for each set are shown in Figure 2.6.

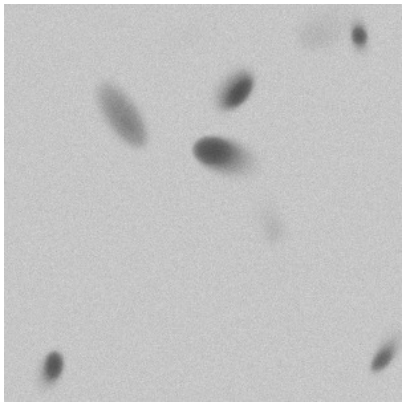
Difficulties in detection arose when two or more spores in the data were nearly parallel in their major axes and very close together, in which case one model spore was sometimes fit to both. Occasionally, multiple model spores were fit to a single spore in the data. In a few cases, no proposal was accepted for a few spores in the data. We observed that this happened when the ellipsoid in the data was relatively long and skinny, yielding little evidence in the image data. Further, due to not occupying many pixels in the image data, the Hough accumulator for these spores had fewer votes, so we expect proposal would require more iterations by either the data-driven and prior-based birth moves.

2.7.2 Sampler convergence rate

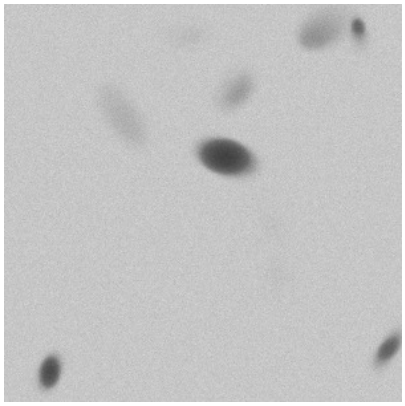
The analysis of convergence rate for the sampler was done on both synthetic data sets. For each data set, the sampler was run 4 times, with a different random seed each time. A good measure of the convergence rate of the sampler is the iterative log likelihood (2.9) of the image data given the model, which is what we show here.



(a) 3-D visualization

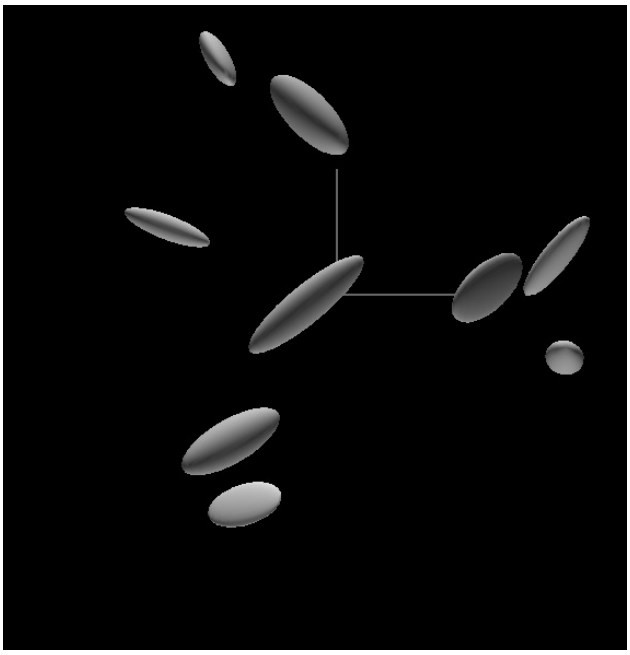


(b) 34 of 80

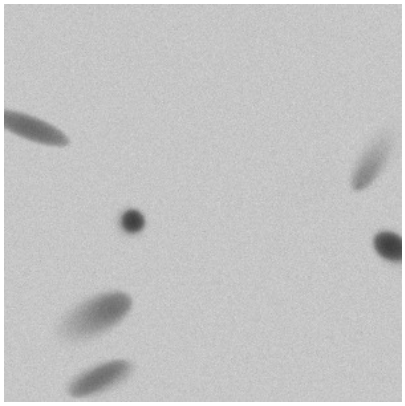


(c) 42 of 80

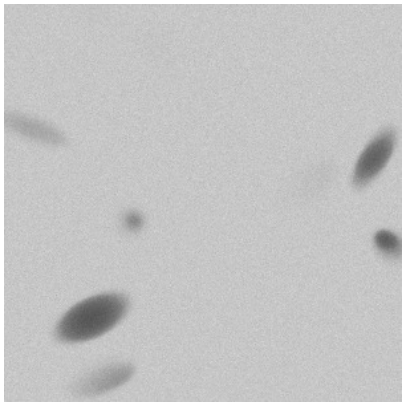
Figure 2.4: Images from synthetic spore data set $\mathcal{S}_1$. The 3-D visualization viewpoint in (a) is directed towards the origin and parallel to the z-axis. The images in (b) and (c) show the optically sectioned ellipsoids at two focal planes perpendicular to the z-axis and convolved with a sampled parameterization of the PSF.



(a) 3-D visualization



(b) 20 of 80



(c) 30 of 80

Figure 2.5: Images from synthetic spore data set $\mathcal{S}_2$. The 3-D visualization viewpoint in (a) is directed towards the origin and parallel to the z-axis. The images in (b) and (c) show the optically sectioned ellipsoids at two focal planes perpendicular to the z-axis and convolved with a sampled parameterization of the PSF.

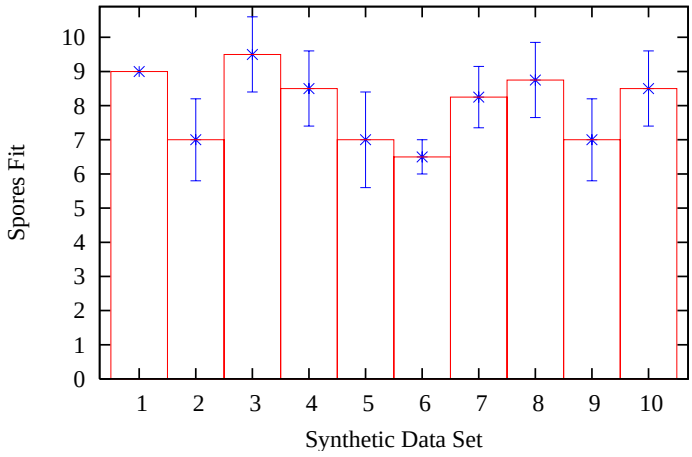
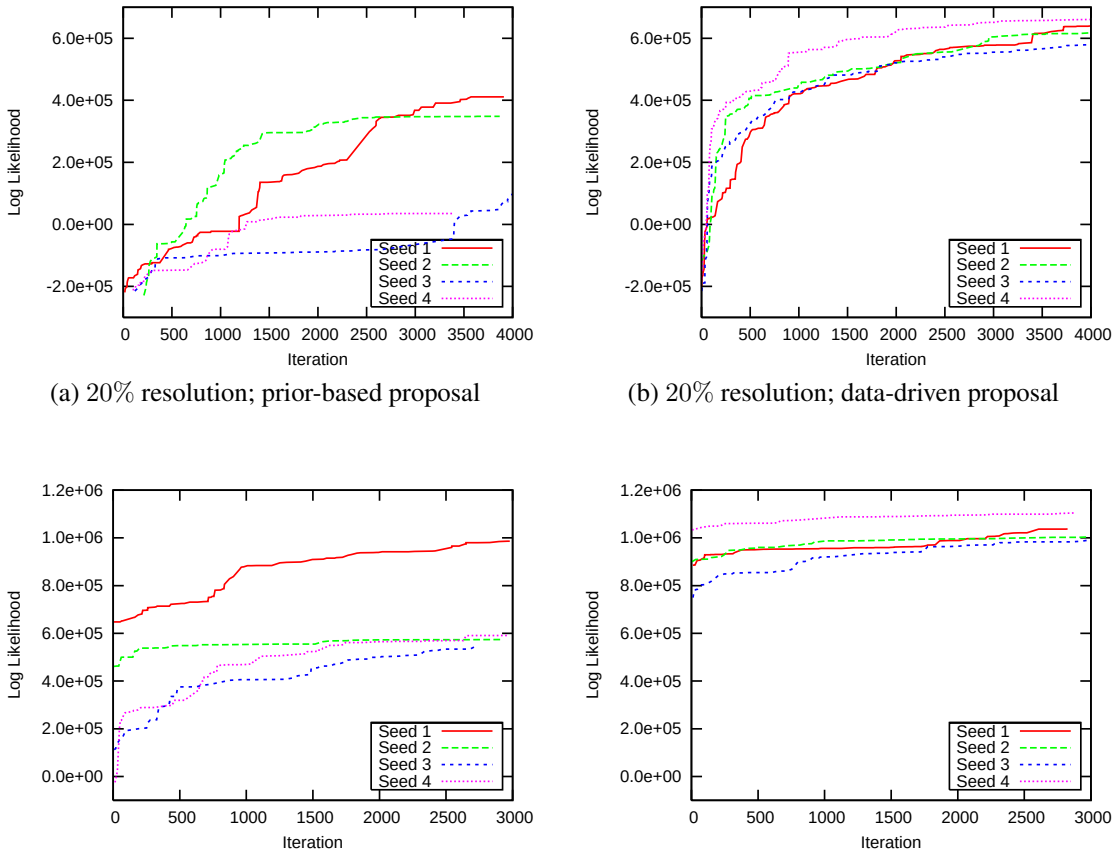


Figure 2.6: Mean number of spores correctly fit to the synthetic data sets, standard deviation bars are shown. Each synthetic set contained 10 spores and was run with four different random seeds.

The image data in the likelihood is modeled as pixel based, so computation of the log likelihood at each iteration of the sampler can require a significant amount of time. To ameliorate this effect, the sampler is run on a hierarchy of down-sampled image data. For example, the synthetic data is initially down-sampled to a resolution of 20% (5 pixels averaged into 1) and run for a number of iterations. Following this, the synthetic data is again down-sampled, but to a higher resolution and run for further iterations. As previously mentioned, the effect of this resolution hierarchy is that run-time performance of the sampler is improved by getting a rough fit for the model at a low resolution, then increasing the resolution to allow the sampler to fine-tune its fit.

To test the effects of the data-driven birth proposal on convergence, we ran the sampler for 4K iterations at 20% resolution followed by 3K iterations at 25% for both the prior-based and data-driven proposals. As predicted, adding a data-driven proposal to the birth move improves the convergence rate of the sampler dramatically; the number of iterations required by the sampler to achieve a high likelihood is quantitatively much less when using the data-driven proposal. This can be seen in the log likelihood plots of Figures 2.7 and 2.8. Notice that when the resolution is increased, the log likelihood is increased by a constant factor. A qualitative improvement in the convergence rate of the sampler to a



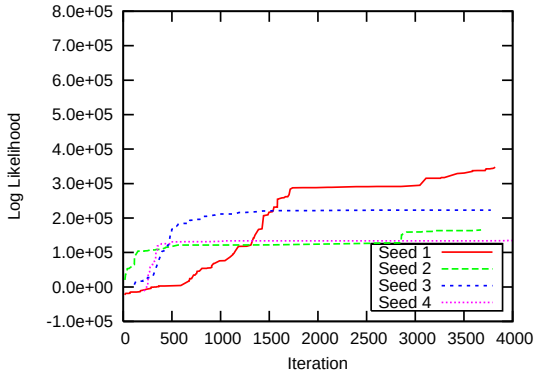
(a) 20% resolution; prior-based proposal

(b) 20% resolution; data-driven proposal

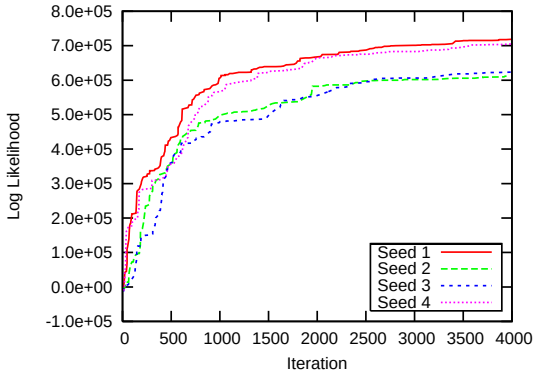
(c) 25% resolution; prior-based proposal

(d) 25% resolution; data-driven proposal

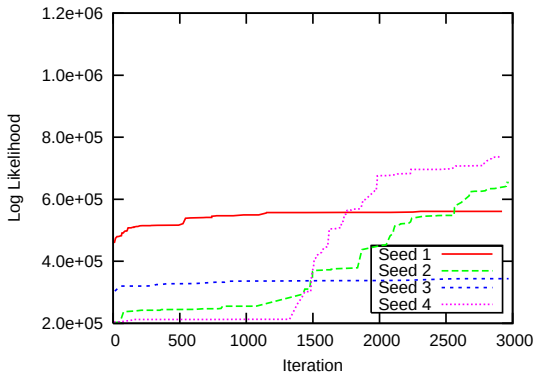
Figure 2.7: Log likelihood plots for synthetic spore data set $\mathcal{S}_1$ (Figure 2.4) comparing the prior-based proposal against data-driven. Plots (a) and (c) used the model prior for birth proposals with the resolution increasing from 20% after 4000 iterations to 25%. Plots (b) and (d) used data-driven birth proposals with a similar resolution increase from 20% to 25%. When the resolution is increased, the log likelihood is scaled by a constant.



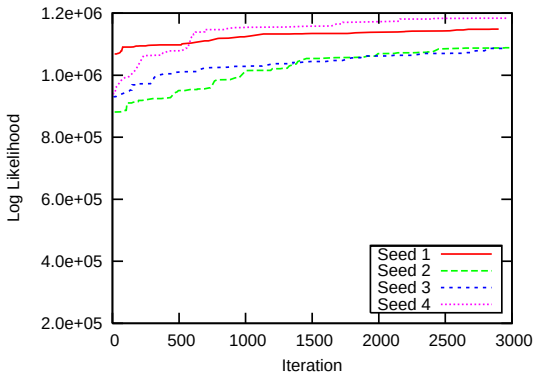
(a) 20% resolution; prior-based proposal



(b) 20% resolution; data-driven proposal

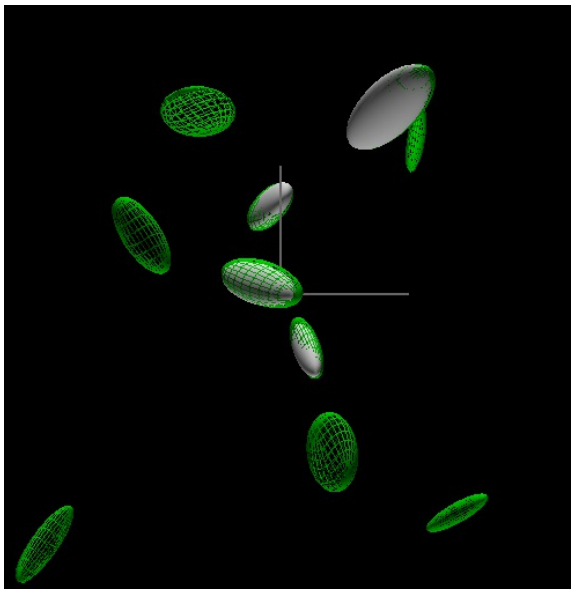


(c) 25% resolution; prior-based proposal

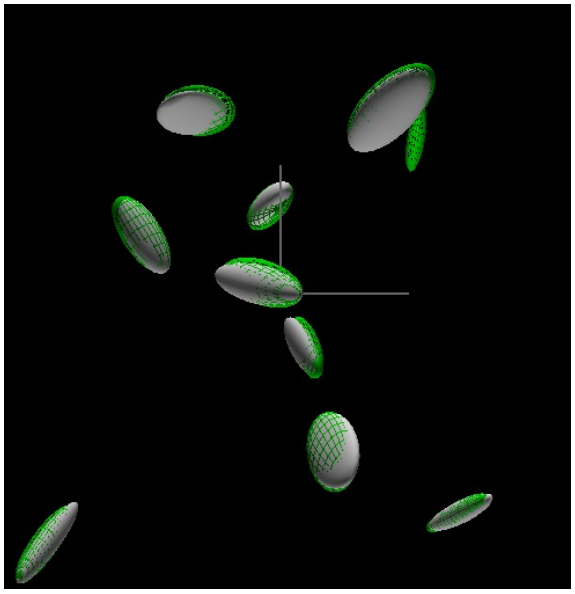


(d) 25% resolution; data-driven proposal

Figure 2.8: Log likelihood plots for synthetic spore data set $\mathcal{S}_2$ (Figure 2.5) comparing the prior-based proposal against data-driven. Plots (a) and (c) used the model prior for birth proposals with the resolution increasing from 20% after 4000 iterations to 25%. Plots (b) and (d) used data-driven birth proposals with a similar resolution increase from 20% to 25%. When the resolution is increased, the log likelihood is scaled by a constant.

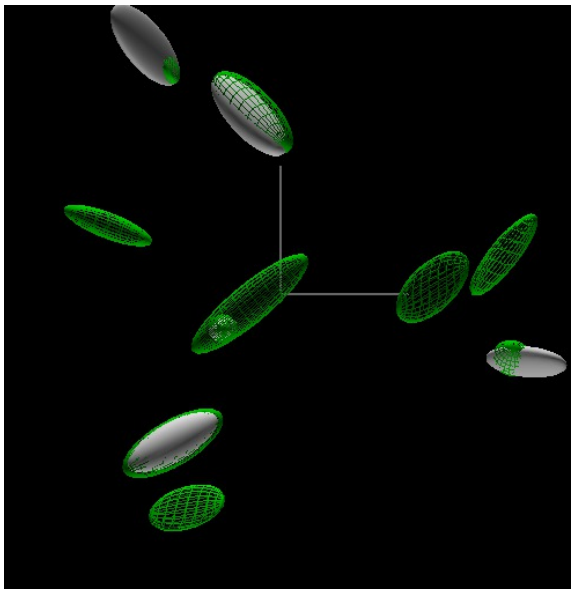


(a) Prior-based proposal

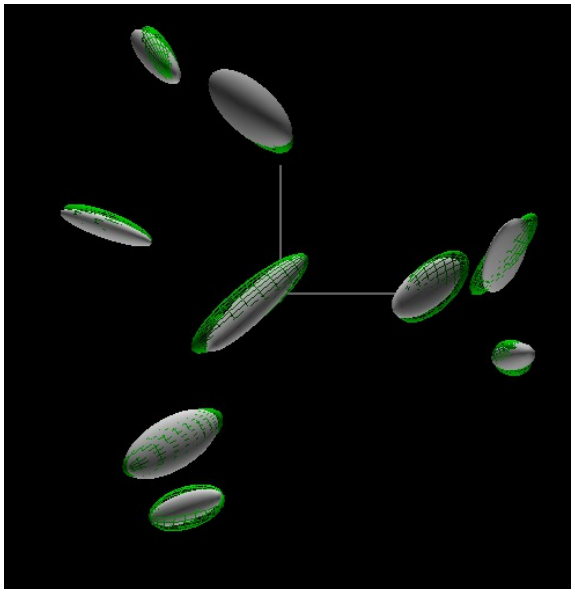


(b) Data-driven proposal

Figure 2.9: Qualitative comparison of samples from synthetic spore data set $\mathcal{S}_1$ after 8400 iterations using the prior-based birth proposal (a) and the data-driven proposal (b). The green wire-frame spores are the synthetic data and the gray solids are the estimates. Notice that the prior-based proposals missed many of the spores.



(a) Prior-based proposal



(b) Data-driven proposal

Figure 2.10: Qualitative comparison of samples from synthetic spore data set $\mathcal{S}_2$ after 8400 iterations using the prior-based birth proposal (a) and the data-driven proposal (b). The green wire-frame spores are the synthetic data and the gray solids are the estimates. Notice that the prior-based proposals missed many of the spores.

	α		β		γ	
	mean	stdev	mean	stdev	mean	stdev
$\mathcal{A}_1$	0.93	0.001	0.82	0.44	1.31	0.30
$\mathcal{A}_2$	0.93	0.030	1.06	0.14	1.35	0.33

Table 2.1: Mean PSF model parameters inferred from the *Alternaria* data from four random starting states. As expected, the fit parameters are similar for the data sets, which were imaged under the same microscope.

good model can be seen in Figures 2.9 and 2.10.

2.7.3 Alternaria evaluation

Two sets of *Alternaria* image stacks were evaluated: $\mathcal{A}_1$ comprising 102 images of size 800×800 pixels and $\mathcal{A}_2$ with 80 images of size 700×700. Images of these sets can be seen in Figure 2.1. The number of spores in the data sets were manually counted and found to be 17 and 21, for $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively.

We ran the sampler on both *Alternaria* data sets for 500 iterations at a resolution of 20%. As with the synthetic evaluation, four instances of the sampler were run on the data sets, each time with a different random seed. Figure 2.11 shows a 3-D rendering of a fit model for each data set next to the detected *Alternaria* surface used for generating data-driven birth proposals.

In $\mathcal{A}_1$ the average number of spores detected was $6\,(35.2\%)$, and $8.75\,(41.7\%)$ for $\mathcal{A}_2$. While we did not achieve 80% accuracy, as in the synthetic case, the results are still noteworthy considering the amount of non-spore structure and substantial blur in the data.

The average inferred background intensity for $\mathcal{A}_1$ and $\mathcal{A}_2$ was 0.78 and 0.75 with a negligible standard deviation. Table 2.1 gives the inferred PSF model parameters for the data sets. The standard deviation is relatively high for PSF parameters β and γ. This is most likely due to the sampler adding variance to the PSF in order to accommodate the large quantity of non-spore structure in the images.

We tested the effect of using our model of the PSF for fitting spores versus using a 3-D Gaussian and a delta function. In the case of the Gaussian, a σ of 0.6 was chosen for

	Model PSF		Gaussian PSF		Delta PSF	
	mean	stdev	mean	stdev	mean	stdev
$\mathcal{A}_1$	6.0	0.9	4.75	0.4	2.5	0.5
$\mathcal{A}_2$	8.75	0.8	7.25	2.9	6.25	2.2

Table 2.2: Mean number of spores correctly fit to *Alternaria* data using a delta function, Gaussian, and our model as the PSF. Four random starting states were used for each data set. It is clear that fitting our model of the PSF improves spore detection. Data set $\mathcal{A}_1$ has 17 spores and $\mathcal{A}_2$ has 21.

all dimensions by empirical analysis of the *Alternaria* image data. This is consistent with work done previously to approximate the PSF (Al-Awadhi et al., 2004; Al-Awadhi, 2001; Shaw and Rawlins, 1991).

Table 2.2 lists the average number of spores detected using each of the PSFs during model inference. Figure 2.12 shows images in $\mathcal{A}_1$ compared to inferred model-scene images that are convolved with each of the PSF types. These results combined show that fitting our model of the PSF to the image data most closely resembles the imaging effects of the microscope and enables a more accurate estimate of structure in the images.

2.8 Conclusion

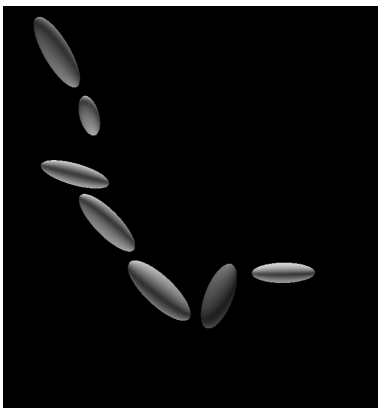
Much biological structure is represented well by three-dimensional structure models, particularly for the purposes of quantitative analysis. We have shown how parts of biological structure, such as spores, can be with modeled with geometric primitives, like ellipsoids, and fit using statistical inference.

We developed a generative likelihood model for observed image data that combines our structure and imaging models. Our approach to fitting is Bayesian statistical inference. We present a reversible-jump sampler for fitting independent ellipsoids and the imaging model to a stack of microscope images simultaneously. Our results showed that fitting structure in microscope images is improved by modeling the point spread function of the imaging system. We observe that the parameters of the imaging system are fairly consistently fit, and that ellipsoids are successfully fit to synthetic data. We also

demonstrate, through the use of data-driven MCMC that ellipsoids can be fit to spore substructures in the images of *Alternaria*

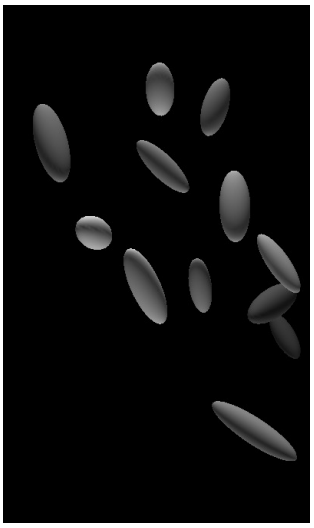
In the next chapter, we extend this work to model the overall structure of *Alternaria* with a grammar of its growth. We present an idea to fit substructures in the data, such as spores and hyphae, under the constraint that their combined structure is an instance of the grammar. One type of grammar that may be useful for this task is a stochastic L-system (Lindenmayer, 1968, 1975), which is commonly used to generate realistic instances of plants in computer graphics.

(a) Surface of $\mathcal{A}_1$



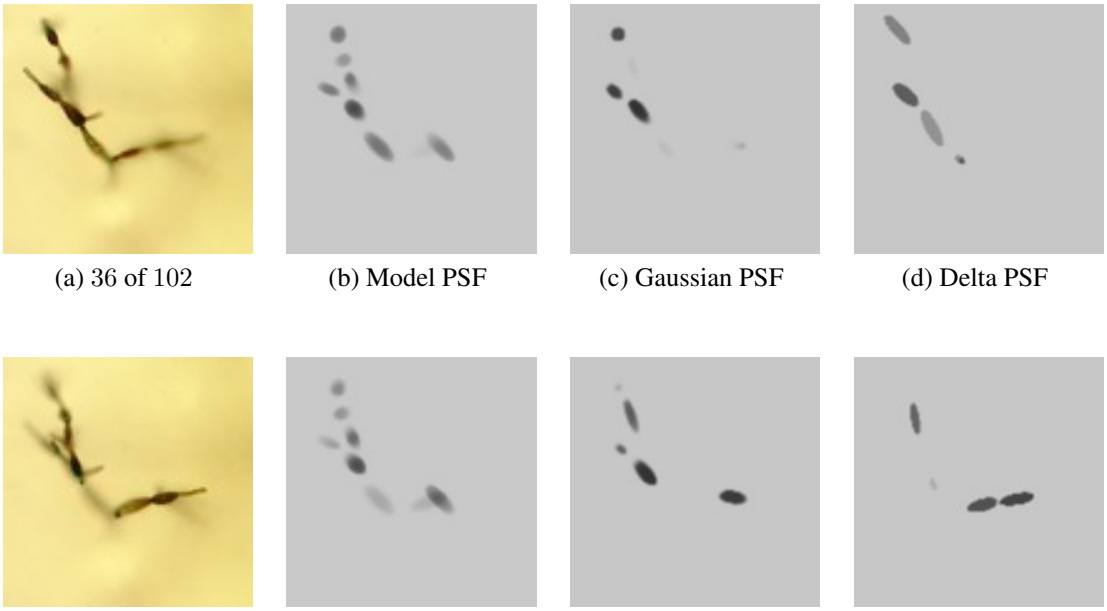
(b) Spores in $\mathcal{A}_1$

(c) Surface of $\mathcal{A}_2$



(d) Spores in $\mathcal{A}_2$

Figure 2.11: Reconstructed surface of *Alternaria* in the image stacks and 3-D renderings of corresponding inferred spore models. The surface detection algorithm for data-driven birth proposals generated the views in (a) and (c). Perceived structure in these images is known only to the viewer. Figures (b) and (d) represent detected spore structure.



(a) 36 of 102 (b) Model PSF (c) Gaussian PSF (d) Delta PSF

(e) 48 of 102 (f) Model PSF (g) Gaussian PSF (h) Delta PSF

Figure 2.12: Illustration of the effects for three different PSFs used to detect spores. From top to bottom, column (e) contains images 36 and 48 out of 102 from *Alternaria* data set $\mathcal{A}_1$. The other columns are corresponding images from inferred model-scenes convolved with the learned PSF model (f), a 3-D Gaussian (g), and a delta function (h). Notice that the images in (f) most closely resemble the *Alternaria* data in (e), indicating that our PSF model is substantively closer to the true PSF than a Gaussian and a delta function.

CHAPTER 3

Inferring Grammar-based Models for Biological Structure

3.1 Introduction

The function of an object is often closely related to its structural form. As a result, the process of understanding what a novel item *is* or *does* frequently begins with an inspection of its structure. This is particularly true in biology, where scientific inquiries of microscopic specimens focus on observing and quantifying structure in images under varying experimental conditions to test hypotheses of specimen functionality. Manually obtaining such results, however, is expensive and time-consuming. In Chapter 2 we presented a method to automatically infer independent structural components of biological specimens from microscopic images. In this chapter we build on that work by detailing a more complete representation for biological structure that uses a grammar to describe patterns of growth. We also present an algorithm to infer instances of the structure model from the same microscopic images.

Many biological specimens comprise a set of connected substructures that are recursively related and can be described by a formal set of rules explaining their growth. The set of rules is a grammar for growth and is similar to Lindenmayer-systems (Lindenmayer, 1975) used in graphics. L-systems are related to context-free grammars. They consist of a set of production rules containing terminal and non-terminal symbols that are recursively substituted to produce a string of terminals. The terminal symbols in the rules represent substructures of the plant, such as a unit stem, branching stem, or leaf. By stochastically and recursively applying these rules, an instance of the grammar is generated. We consider such a grammar as a basis for building a probabilistic specimen model to infer from data. The model is constructed so that repeated application of the grammar rules can generate a parameterization of it. Thus, our approach focuses on fitting a complete model of the specimen, unlike other methods that fit only individual and independent substructures

of specimen (e.g., Chapter 2; Al-Awadhi et al., 2004).

Images formed under a transmitted-light microscope contain a significant amount of blur due to the high magnification and shallow depth-of-field in the optics. This makes accurate localization of structure in the images difficult. Rather than try to eliminate the blur from the images through deblurring methods (Conchello, 1998; Holmes, 1992), we follow the approach of Chapter 2 and model the optical system of the microscope. This enables a fuller understanding of the image formation process and the ability to unlock structural information captured in the image blur. Combining a grammar-based structure model for a specimen with a model for the optics of the imaging system is an innovative and powerful way to understand microscopic images accurately.

Inferring such models is analytically very difficult; the number of parameters, their interdependence, and the fact that the dimensions of the model is itself a parameter, create a space that is prohibitively complex to work with. Thus, we create a Markov chain Monte Carlo sampler (Sokal, 1989; Neal, 1993; Andrieu et al., 2001; Liu, 2001; Bishop, 2006) to efficiently explore the parameter space in search of a likely set of parameters that generated the data. The moves of the sampler that guide its search through the model parameter space effectively embody the rules of the grammar for the specimen. Furthermore, the sampler infers both the structure and imaging models simultaneously so that each can benefit from an improved fit of the other. Since the dimensionality of the model is unknown, we further construct a reversible-jump MCMC (Green, 1995, 2003) sampler to handle model selection and traverse the multi-dimensional parameter spaces.

A good example of a biological specimen whose structure is recursive in nature is *Alternaria*, the microscopic genus of fungus introduced in Chapter 2. To aid in the analysis of *Alternaria* and illustrate our ideas for structure modeling and inference, we developed a grammar-based model for *Alternaria* and sampling methods for inference. Figure 2.1 shows the self-similarity that exists within two examples of *Alternaria* in three-dimensional microscopic image stacks. Our approach focuses on developing a set of simple rules that can generate these self-similar structures and fitting instances of these rules to 3-D stacks of images.

We are aware of only one previous instance that uses an L-system model during the

process of biological structure recognition (Samal et al., 2002). However, the L-system model was not directly used for the recognition task; rather, it generated synthetic plant images for training a rule-based species classifier. The primary focus of the work was feature detection and a classification task, where the features were obtained using standard image analysis techniques on the images generated by the L-system model.

3.2 Stochastic grammar for structure

L-systems were first introduced as a mathematical abstraction for modeling cellular interactions in plants (Lindenmayer, 1968). L-systems are a type of formal grammar similar to context free grammars (Manning and Schutze, 1999) with the exception that all production rules are applied in parallel and simultaneously replace all letters in a word (Prusinkiewicz and Lindenmayer, 1990). A parametric stochastic context-free L-system is defined by a set of symbols, production rules, and probabilities for rule application. For example,

$$G = (V, \Sigma, R, r_0, \pi)\,, \tag{3.1}$$

where V is the set of non-terminal symbols that are replaced during production rule rewriting, and Σ is the set of terminal symbols comprising the grammar alphabet. The collection of production rules, denoted R, maps $V \rightarrow V \cup \Sigma$; the base production rule r_0 is an axiom consisting of terminal symbols only. The grammar becomes stochastic by including a probability distribution π defined over production rules. The probability distribution π characterizes how frequently each of the rules is applied, making sentence production non-deterministic. Sentences are produced by randomly selecting a production rule to apply and replacing each non-terminal symbol from V with combinations of symbols from both the non-terminals and the alphabet $V \cup \Sigma$. The axiom rule r_0 is used to complete sentences by replacing non-terminals with symbols from the alphabet only. Recursively following this process for many iterations can generate complex, self-similar structures (see Figure 3.1).

L-systems have been successfully used to visualize and simulate a wide range of

plants and trees (Oppenheimer, 1986; Weber and Penn, 1995). This includes modeling peach trees (Allen et al., 2005), *Fraxinus pennsylvanica* shoots (Hammel et al., 1995), proteins (Escuela et al., 2005), and other herbaceous plants (Prusinkiewicz et al., 1988). They have also been used to model and render entire plant ecosystems (Deussen et al., 1998).

3.2.1 Alternaria L-system

The fungi of genus *Alternaria* grow similarly to plants. They have a long vegetative hyphae, like a stem, with branches that have a three-dimensional, repeating pattern (Simmons, 1999). Each branch begins as a hypha capable of producing reproductive spores, known as a primary conidiophore development. A hypha cell in a branch can develop another hypha cell or a spore through apical growth at its tip. When a spore develops, several structures can occur depending on the species: another spore, a lateral intra-conidium hypha branch coming from one cell of the spore, an apical conidium terminus hypha branch coming from the tip, or a sub-conidium conidiophore hypha branch coming from the hypha cell immediately before the spore.

In general, the fungus continually produces new hyphae cells and spores. These in turn develop into more hyphae cells and spores in the manner previously described. The overall structure is then defined by a recursive growth pattern with self-similar structure. To model this growth process we use a parametric, stochastic, and context-free L-system.

In our grammar for *Alternaria* (Spriggs et al., 2007), the set of parameters and probability distributions are determined from morphological characteristics obtained by plant pathologists observing the structure. What follows is a description of the rules and symbols in the grammar. We represent how the long vegetative hyphae grows and develops branches with the rule

$$V_{\mathsf{hypha}} \;\rightarrow\; H(\pi_v, \pi_h)\; B\; V_{\mathsf{hypha}}\,. \tag{3.2}$$

The non-terminal $H(\pi_v, \pi_h)$ stochastically expands to a string of hyphae cells whose count is distributed according to π_v. The size and orientation of each cylinder shaped

hypha is further drawn from the probability distribution π_h. The orientation of each cell is defined relative to its connecting structure by a rotation matrix with two angles φ, ϑ.

A branch off of the main vegetative hyphae is replaced by further developments of *Alternaria* substructures, including a single chain of hyphae,

$$B \;\rightarrow\; H(\pi_v, \pi_h)\, C_1\,. \tag{3.3}$$

The symbol C_1 is stochastically replaced with one of several conidiophore developments. These include a sub-conidium hypha C_3, followed by a spore; a single hypha cell $h(\pi_h)$; or no change resulting in a pause in development. We write these rules as

$$C_1 \;\rightarrow\; C_3\; s(\pi_s)\; C_2 \;\mid\; h(\pi_h)\; C_1 \;\mid\; C_1\,, \tag{3.4}$$

Each of the three possible developments is randomly chosen based on a discrete density function defined by species-specific information from the mycologist. Since L-systems simulate growth by applying rules simultaneously and in parallel, the no-change rule is necessary to represent slower growth by part of the structure.

We complete the grammar for *Alternaria* with a pair of rules that describe branching pattens near spores. Specifically, a spore is expanded by another spore, growth of an apical or lateral branch, or switching to hypha development:

$$C_2 \;\rightarrow\; s(\pi_s)\; C_2 \;\mid\; \mathit{Apical}\; C_1 \;\mid\; \mathit{Lateral}\; C_1 \;\mid\; C_2\,, \tag{3.5}$$

$$C_3 \;\rightarrow\; h(\pi_h)\; C_1 \;\mid\; C_3\,. \tag{3.6}$$

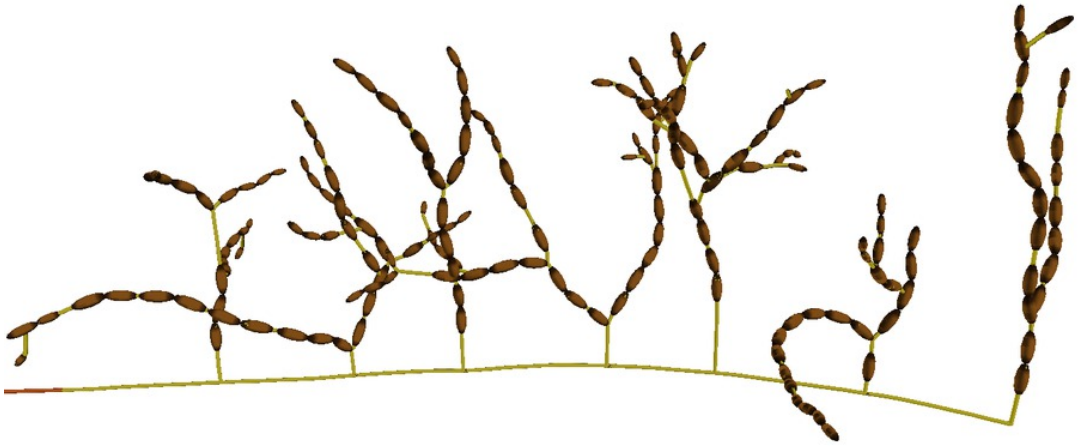
Figure 3.1 shows instances of *Alternaria* generated by this L-system with the on-line tool available at (Spriggs, 2007).

3.3 Modeling

We combine the grammar for *Alternaria* with the imaging system in Chapter 2 to build a generative model for the observed microscope image stacks. In this and the next sec-

(a) 3 iterations

(b) 12 iterations



(c) An instance of vegetative hyphae with branches

Figure 3.1: Three-dimensional representation generated by the L-system for *Alternaria*. Spores are modeled with ellipsoids and the hyphae with cylinders. The top-left panel shows the growth of a simple structure after three iterations (a), followed by a branch of development after 12 iterations (b), and an instance of the complete grammar (c). The images were generated using the *Alternaria* L-system tool available on-line at Spriggs (2007).

tion we present a statistical model for the complete structure of *Alternaria* based on its grammar. We also describe how the image formation process utilizes our model for the imaging system and an instance of the *Alternaria* model to generate data. Given a stack of captured images containing *Alternaria*, we aim to infer an instance of the grammar that best fits the data

3.3.1 Grammar-based structure

We model the structure of *Alternaria* for statistical inference based on its grammar for growth. We represent its hyphae and spores as an ordered set of cylinders and ellipsoid substructures, and enforce connectedness among these primitives elements to one apical growth and multiple lateral branches. The cylinder and ellipsoid substructures are the geometric primitives used in our model. The model has a root position and direction of growth given by $(p_o, \varphi_o, \vartheta_o)$, where the position is in the 3-D imaging window $\mathcal{W}$. The growth direction is defined by two Euler angles for symmetric objects, i.e., ellipsoids and cylinders. Denote the space containing all root position and orientations by $\mathbf{P}$.

Let the i-th apical hypha with m_h number of branch hyphae be defined as a collection of geometry parameters and topology indices that describe which substructures are attached in the ordered set

$$\boldsymbol{h}_i^{(m_h)} = (l, w, \varphi, \vartheta, \lambda, j, k_1, \ldots, k_{m_h}) \,. \tag{3.7}$$

The length and width of the hypha cylinder is denoted l, w, while its orientation is given by two Euler angles relative to the growth direction of its apical parent. The cylinders are radial symmetric, so only two angles of rotation are necessary to describe all possible orientations. As in the independent ellipsoid model of Chapter 2 the substructures of the complete *Alternaria* model have an average opacity $\lambda \in [0, 1]$, or absorption rate, in the image.

The parameter subset $(j, k_1, \ldots, k_{m_h})$ define topological information for a hypha substructure. The index j specifies which of the other substructures in the ordered is the apical out-growth of this one. The lateral branching substructures are indexed by $k_1, \ldots, k_{m_h}$.

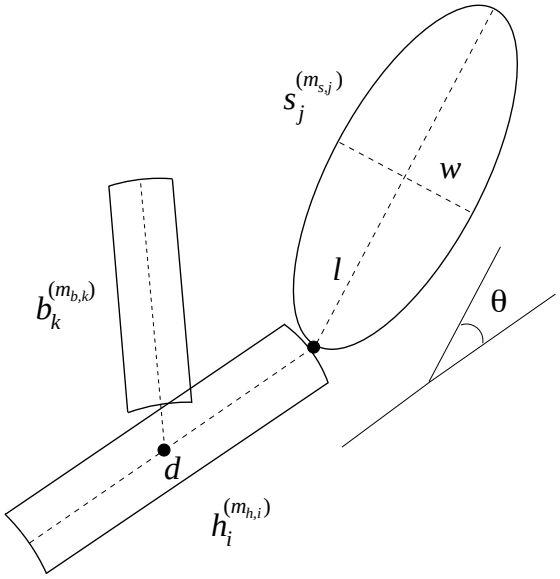


Figure 3.2: An example of a spore $\boldsymbol{s}_j$, lateral branch $\boldsymbol{b}_k$, and apical hypha $\boldsymbol{h}_i$, and how they are connected in the model of *Alternaria*.

We similarly define the spore and branch hypha substructures, $\boldsymbol{s}_j^{(m_s)}$ and $\boldsymbol{b}_k^{(m_b)}$. A lateral branch hypha, however, has two additional parameters. Since observing infinite recursion in the growth of *Alternaria* is highly unlikely, we parameterize the level of each lateral branch substructure with an integer $t \geq 0$. This gives the depth level of a hypha with respect to the root element. Branch hyphae also parameterize the normalized position along the major axis of its parent where the branch is located, $d \in [0, 1]$. The position of all descendant substructures is determined by their size and relative orientation to their parent. The root substructure of the model has a base orientation and position as previously described. The diagram in Figure 3.2 illustrates the geometric and topological substructure parameters and their relationships for a small example.

For the purposes of defining our statistical representation and inference algorithm, we describe the parameter space over the complete structure model for *Alternaria*. Let $\boldsymbol{n} = (n_h, n_s, n_b)$ be the number of substructure elements in the model. Then the number of branch hyphae for each substructure lies in the space

$$\mathbf{M} = \left\{ \mathbf{m} : \sum_{i=1}^{n_h} m_{h,i} + \sum_{i=1}^{n_s} m_{s,i} + \sum_{i=1}^{n_b} m_{b,i} = n_b \right\} . \tag{3.8}$$

We further define parameter spaces over all ordered sets of substructure types. For all ordered sets of n_h apical hypha with $\boldsymbol{m}_h$ branches, this is given by

$$\mathbf{H}^{(n_h, \boldsymbol{m}_h)} = \mathbf{H}_1^{(m_{h,1})} \times \cdots \times \mathbf{H}_{n_h}^{(m_{h,n_h})} . \tag{3.9}$$

The parameterizations over all sets of spores and branches are similarly defined as $\mathbf{S}^{(n_s, \boldsymbol{m}_s)}$ and $\mathbf{B}^{(n_b, \boldsymbol{m}_b)}$.

By combining the subspaces for root position and orientation, branch hypha distribution, and ordered sets of $\boldsymbol{n}$ substructures, we define the space over all *Alternaria* models as

$$\boldsymbol{\Psi}^{(\boldsymbol{n})} = \bigcup_{\mathbf{m} \in \mathbf{M}} \mathbf{P} \times \mathbf{H}^{(n_h, \boldsymbol{m}_h)} \times \mathbf{S}^{(n_s, \boldsymbol{m}_s)} \times \mathbf{B}^{(n_b, \boldsymbol{m}_b)} . \tag{3.10}$$

The construction of the space is such that an instance of the grammar for *Alternaria* can be mapped into it.

3.3.2 Image formation

To handle the significant amount of blur in our image data, we utilize the model for brightfield microscope imaging systems presented in Chapter 2. Specifically, we follow Section 2.3.2 and represent the point spread function of the microscope with the 3-D response function $\tilde{h}(x, y, z)$ defined in equation (2.3). The model for the imaging system is parameterized by α, β, γ of the PSF and υ for the ambient background pixel intensity.

Similar to before, let $\boldsymbol{\theta}^{(\boldsymbol{n})} = (\boldsymbol{\phi}, \boldsymbol{\psi}^{(\boldsymbol{n})})$ be an instance of the parameter space $\boldsymbol{\Phi} \times \boldsymbol{\Psi}^{(\boldsymbol{n})}$ defined over the grammar-based structure and imaging system models. Then the space of potential solution spanning all structure and imaging configurations is

$$\boldsymbol{\Omega} = \bigcup_{\boldsymbol{n} \in \mathbb{N}^3} \boldsymbol{n} \times \boldsymbol{\Phi} \times \boldsymbol{\Psi}^{(\boldsymbol{n})} . \tag{3.11}$$

Given a $\left(\boldsymbol{n}, \boldsymbol{\theta}^{(n)}\right) \in \boldsymbol{\Omega}$, we generate a model-scene image stack $\mathcal{I}_\theta(i,j,k)$ by intersecting all of the geometric substructure primitives in the structure model with a set of equally spaced planes parallel to the microscope focal plane. The stack of images can be thought of as an estimate of the unobserved microscope images, without any distortions from the imaging system. The image stack approximates the optical sectioning performed by the microscope at stepped focal lengths.

The pixels intensity values of the model-scene are given by the imaging system background parameter and substructure intensities. Background pixels of $\mathcal{I}_\theta(\cdot)$ have the highest saturation with value υ. Pixels inside a plane intersected with a cylinder or ellipsoid belonging to a substructure with opacity λ have the value $\upsilon\,(1-\lambda)$. Figure 3.4 shows an illustration of the optical sectioning and rendering process of the model-scene for the complete *Alternaria* structure model.

We model the image data as statistically generated by the structure representation captured under the effects of the imaging system. Conditioned on a model-scene $\mathcal{I}_\theta(\cdot)$ and imaging system $\tilde{h}(\cdot)$, pixel intensities in the 3-D image data are generated from independent Gaussians. The means of these Gaussians $\mu_{\mathcal{I}_\theta}(i,j,k)$ are derived exactly as in (2.6) from the model-scene convolved with the PSF. The variances $\sigma^2_{\mathcal{I}_\theta}(i,j,k)$ are also a function of the convolved model-scene and PSF and defined in (2.7). The effects of blurring the model-scene with the PSF to generate hypothesis of observed image data can be seen in Figure 3.4.

3.4 Inference

For a stack of *Alternaria* image data $\mathcal{I}(i,j,k)$ in the 3-D window $\mathcal{W}$, our goal is to discover the set of connected cylinders and ellipsoids in the model $(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})}) \in \boldsymbol{\Omega}$ that best fits the data. To do this, we follow a similar process as in Chapter 2 and formulate a Bayesian statistical inference problem by defining a posterior over the model space given the image data and search for a maximum. The posterior distribution in this case has the form

$$p\left(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})} \mid \mathcal{I}\right) = k_p \, L\left(\mathcal{I} \mid \boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})}\right) \, \pi\left(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})}\right), \tag{3.12}$$

with the normalization constant k_p, $L(\cdot \mid \cdot)$ is the likelihood of the image data, and $\pi(\cdot)$ is the model prior.

Conditioned on our structure and imaging models, we apply the i.i.d. Gaussian pixel assumption of Chapter 2 with means and variances given by the model-scene convolved with the PSF. The likelihood function over image stacks is then the same as before, with a product of Gaussian pixel probabilities, as in equation (2.9). The prior information for our complete *Alternaria* structure model is quite different than the independent spore model, however, and a more detailed description of it is given below.

3.4.1 Structure and imaging priors

The prior over parameters in $\boldsymbol{\Omega}$ assumes independence between the structure and imaging models and is defined as

$$\pi\left(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})}\right) = \pi_{\Phi}(\boldsymbol{\phi}) \; \pi_{\Psi}\left(\boldsymbol{n}, \boldsymbol{\psi}^{(\boldsymbol{n})}\right) \; . \tag{3.13}$$

The priors for the imaging parameters $\boldsymbol{\phi}$ are distributed according to independent Gaussians with relatively uninformative hyperparameters. The position of the *Alternaria* root p_o ranges uniformly over the 3-D image window $\mathcal{W}$ that has volume $V_{\mathcal{W}}$. Since the orientation and position of a substructure in *Alternaria* is determined by the configuration of its parent and its own internal parameters, we model each substructure as conditionally independent given its parent.

The density function for each substructure is composed of independent subdensities defined over its parameters. We assume that the two minor axis of spore ellipsoids are equal, so for both ellipsoids and cylinders we only need to specify orientation and size parameters. The Euler orientation angle φ is Gaussian distributed over $[0, \pi]$, and ϑ is uniformly distributed over $[-\pi, \pi]$. We define cylinder and ellipsoid size with width w and length l parameters, which are also Gaussian distributed. The opacity λ is just as before, uniformly distributed over $[0, 1]$. The probability a substructure is added either laterally or

apically is p_h, p_s, p_b. The lateral position d of a branch hypha is Gaussian distributed over $[0, 1]$, and the probability that a branch is created at depth t is geometrically distributed.

Let $\boldsymbol{a}_j$ be the parent substructure of hypha $\boldsymbol{h}_i$ in $\boldsymbol{\psi}^{(\boldsymbol{n})}$. Then the density function for a hypha is given by the set of independent subdensities over branch type, size, orientation, and opacity,

$$f_{\boldsymbol{h}}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) = p_h \, f_{w,l}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) \, f_{\varphi,\vartheta}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) \, f_{\lambda}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) \, . \tag{3.14}$$

The probability of lateral attachment p_h is a constant. Each of the other subdensity functions are defined as

$$f_{w,l}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) = \frac{\sigma_w^{-1} \, \sigma_l^{-1}}{2\,\pi} \, \exp\left[-\frac{(w_i - w_j)^2}{2\,\sigma_w^2} - \frac{(l_i - \mu_l)^2}{2\,\sigma_l^2} \right], \tag{3.15}$$

$$f_{\varphi,\vartheta}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) = \frac{1}{\pi} \, \frac{\sigma_\vartheta^{-1}}{\sqrt{2\,\pi}} \, \exp\left[-\frac{(\vartheta_i - \mu_\vartheta)^2}{2\,\sigma_\vartheta^2} \right], \tag{3.16}$$

$$f_{\lambda}(\boldsymbol{h}_i \mid \boldsymbol{a}_j) = \frac{\sigma_\lambda^{-1}}{\sqrt{2\,\pi}} \, \exp\left[-\frac{(\lambda_i - \lambda_j)^2}{2\,\sigma_\lambda^2} \right]. \tag{3.17}$$

The density function for a spore $f_{\boldsymbol{s}}(\boldsymbol{s}_i \mid \boldsymbol{a}_j)$ is similarly defined. The function for a branch hypha, however, is slightly different with additional factors for normalized branch position d and level t,

$$f_{\boldsymbol{b}}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) = p_b \, f_{w,l}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) \, f_{\varphi,\vartheta}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) \, f_{\lambda}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) \, f_{d}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) \, f_{t}(\boldsymbol{b}_i \mid \boldsymbol{a}_j) \, . \tag{3.18}$$

While the size, orientation, and opacity subdensities are the same as above, the attaching branch position d and topology depth parameter t are distributed according to

$$f_d(\boldsymbol{b}_i \,|\, \boldsymbol{a}_j) = \frac{\sigma_d^{-1}}{\sqrt{2\,\pi}} \exp\left[-\frac{(d_i - \mu_d)^2}{2\,\sigma_d^2}\right], \tag{3.19}$$

$$f_t(\boldsymbol{b}_i \,|\, \boldsymbol{a}_j) = (1 - \alpha_t)^{t_i}\ \alpha\,. \tag{3.20}$$

We model the probability of one or more substructure types existing in the imaging window $\mathcal{W}$ with a Poisson process parameterized by intensities ν_h, ν_s, ν_b. To eliminate the possibility of intersection, we include a term in the prior that restricts substructure interaction. Specifically, we aim to eliminate intersection between spores and hyphae. Intersection in this case is defined as simple geometric overlap between cylinders and ellipsoids. Due to the degrees of freedom in the model, parameterizations of structure could result in self-intersection, which is obviously not possible in the real data. The prior probability for an *Alternaria* model is then

$$\begin{aligned}\pi_\Psi\big(\boldsymbol{n}, \boldsymbol{\psi}^{(\boldsymbol{n})}\big) = k_\pi^{\boldsymbol{n}}\, \frac{1}{V_{\mathcal{W}}}\, \frac{\nu_h^{n_h}\, e^{-\nu_h}}{n_h!} \prod_{i=1}^{n_h} \chi(\boldsymbol{h}_i \nvdash \boldsymbol{a}_{j\neq i})\ f_{\boldsymbol{h}}\left(\boldsymbol{h}_i \,|\, \mathsf{parent}(\boldsymbol{h}_i)\right) \\ \times\ \frac{\nu_s^{n_s}\, e^{-\nu_s}}{n_s!} \prod_{i=1}^{n_s} \ldots\ \frac{\nu_b^{n_b}\, e^{-\nu_b}}{n_b!} \prod_{i=1}^{n_b} \ldots\,,\end{aligned} \tag{3.21}$$

where $k_\pi^{\boldsymbol{n}}$ is a normalization constant for the truncated subdensity functions, $\boldsymbol{a}$ is any type of substructure in the model, and $\nvdash$ denotes no geometric intersection. The characteristic function $\chi(\cdot)$ yields 1 for true and 0 otherwise.

3.5 Sampling

As in Chapter 2, we use reversible-jump Markov chain Monte Carlo sampling for inference of the most likely model under the posterior (3.12). We sample within a topology of the *Alternaria* structure and imaging system using diffusion moves, and sample changes to the topology using jump moves. By alternating between these two types of moves, we explore the full parameter space of the *Alternaria* structure imaged under a microscope.

We construct the moves in the sampler such that it follows a Markov chain that converges to the posterior and draws representative samples, while keeping track of the sample with maximum probability.

We use the Metropolis-Hastings (MH) algorithm for MCMC (Metropolis et al., 1953; Hastings, 1970) under both diffusion and jump moves. At each iteration, the m-th move is run with probability $r(\mathsf{m})$ and a new model $(\boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})})$ is proposed. If the proposed model is a likely to have generated the observed data under the posterior, it is accepted with high probability. For jump moves, we modify the standard MH acceptance probability to enable transitions between parameter spaces containing different *Alternaria* topologies (Green, 1995, 2003).

3.5.1 Sampling within a structure topology

The diffusion moves for sampling with in a topology of *Alternaria* and modifying a substructure in $(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})})$ include rotate, size, $\mathsf{opacity}$, shift, and $\mathsf{lateral}$. We also define moves to update the PSF and background parameters. The proposal distributions for diffusion moves are obtained by modifying the prior (3.13). For parameters updated in a move, we replace their subdensity in the prior with a Gaussian that has means equal to corresponding parameters in the previously accepted model.

The proposal for substructure size follows the same pattern as the proposal for position change in (2.17). Specifically, the size proposal distribution for randomly selecting the j-th hypha and resizing it is

$$q_{\mathsf{size}}\left(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)}\right) = \frac{k^{\boldsymbol{n}}_{\mathsf{size}}}{n_h} \left[\prod_{i \neq j}^{\boldsymbol{n}} \chi(\boldsymbol{a}_i \not\vdash \tilde{\boldsymbol{h}}_j)\right] \times \frac{\sigma_{w,l}^{-2}}{(2\,\pi)^{(3/2)}} \exp\left[-\frac{(\tilde{w}_j - w_j)^2 + (\tilde{l}_j - l_j)^2}{2\,\sigma_{w,l}^2}\right], \tag{3.22}$$

where $\sigma^2_{w,l}$ is a small variance and $k^{\boldsymbol{n}}_{\mathsf{size}}$ is a normalization constant. The structure model of *Alternaria* is connected, so it's position is determined by the root element position and the size/orientation of all the substructures. For this reason, the shift proposal (2.17) is

slightly modified to be applied to all the substructures in the model. All the other moves, such as opacity and rotate, are similar as before. For a branching hypha, we define a proposal to change the connection point d with the base structure

$$q_{\mathsf{lateral}}\Big(\tilde{\boldsymbol{\theta}}^{(n)} \mid \boldsymbol{\theta}^{(n)}\Big) = \frac{k^{\boldsymbol{n}}_{\mathsf{lateral}}}{n_b} \left[\prod_{i\neq j}^{\boldsymbol{n}} \chi(\boldsymbol{a}_i \nVdash \tilde{\boldsymbol{b}}_j)\right] \frac{\sigma_d^{-1}}{\sqrt{2\,\pi}} \exp\left[-\frac{(\tilde{d}_j - d_j)^2}{2\,\sigma_d^2}\right], \tag{3.23}$$

The proposal distributions for other diffusion moves are constructed in the same way.

We use the MH algorithm to generate samples from the posterior 3.12. For the m-th diffusion move, the acceptance probability of a proposed change to the model is

$$\alpha_{\mathsf{m}}\Big(\boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})}\Big) = \min\left\{1, \frac{p(\boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})} \mid \mathcal{I})\, q_{\mathsf{m}}(\boldsymbol{\theta}^{(\boldsymbol{n})} \mid \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})})}{p(\boldsymbol{n}, \boldsymbol{\theta}^{(n)} \mid \mathcal{I})\, q_{\mathsf{m}}(\tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})} \mid \boldsymbol{\theta}^{(\boldsymbol{n})})}\right\}. \tag{3.24}$$

This probability ratio maintains the detailed balance condition in the Markov chain and ensures convergence to the target distribution. For details, refer to Appendix A.

For the size diffusion move, expanding the acceptance probability (3.24) results in many terms canceling out. This includes the difficult to compute normalization constants, the proposal distributions, and most of the prior. The result is a ratio of likelihoods, size prior ratio, and a test for intersection. The acceptance probability for changing the size of the j-th hypha is then

$$\alpha_{\mathsf{size}}\Big(\boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})}\Big) = \min\left\{1, \frac{L(\mathcal{I} \mid \boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})})\, f_{w,l}(\tilde{\boldsymbol{h}}_j)}{L(\mathcal{I} \mid \boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})})\, f_{w,l}(\boldsymbol{h}_j)} \prod_{i\neq j}^{\boldsymbol{n}} \chi(\boldsymbol{a}_i \nVdash \tilde{\boldsymbol{h}}_j)\right\}. \tag{3.25}$$

The rotate, opacity, and shift moves all similarly reduce, with the opacity acceptance not requiring an intersection test. A change to the lateral attachment point of a branch hypha $\boldsymbol{b}_j$ follows this form as well

$$\alpha_{\mathsf{lateral}}\Big(\boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})}\Big) = \min\left\{1, \frac{L(\mathcal{I} \mid \boldsymbol{n}, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n})})\, f_{d}(\tilde{\boldsymbol{b}}_j)}{L(\mathcal{I} \mid \boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})})\, f_{d}(\boldsymbol{b}_j)} \prod_{i\neq j}^{\boldsymbol{n}} \chi(\boldsymbol{a}_i \nVdash \tilde{\boldsymbol{b}}_j)\right\}. \tag{3.26}$$

The diffusion acceptance probabilities for the imaging system parameters ϕ are constructed in the same way, but as with the opacity move, excluding the intersection test.

3.5.2 Sampling structure topologies

The jump moves in the sampler modify the topology of the *Alternaria* structure model by proposing to add or remove substructure pieces and accepting or rejecting. A set of birth/death moves create and prune spore and hypha substructures at apical or lateral positions. For apically connected hyphae and spores, split/merge moves break apart or join together two substructures. A lateral branch can be split or merged with its parent, as well. Finally, a set of switch moves transition one or more hypha to a spore and vice-versa. This collection of sampler moves embody the grammar rules from Section 3.2 and guide the sampler to discovering likely topologies given a stack of observed images.

Apically attaching a hypha to another substructure is only possible if that substructure does not already have an apical attachment. For a particular topology, let n_{birth} be the number substructures that have no attachment at there apical growth point. A hypha birth move consists of randomly selecting one of these n_{birth} substructures, say $\boldsymbol{a}_i$, and attaching a hypha $\tilde{\boldsymbol{h}}$ generated from the normalized hypha density function (3.14) in the model prior. We define the probability of this birth proposal as

$$q_{\mathsf{birth}}(\tilde{\boldsymbol{h}} \mid \boldsymbol{a}_i) = \frac{k_{\mathsf{birth}}}{n_{\mathsf{birth}}} \, f_{\boldsymbol{h}}(\tilde{\boldsymbol{h}} \mid \boldsymbol{a}_i) \, . \tag{3.27}$$

During a hypha death move, one of n_{birth} substructures with an apical hypha is randomly selected to be pruned. We propose removing only one substructure per move, so the hypha to prune must not have any lateral or apical attachments. A proposal distribution over the death move is not necessary, since the selected substructure is completely removed.

The complementary proposals for split/merge consist of selecting a single hypha and splitting it in two, or selecting two apically connected hyphae and merging them. For the split move, we use the width, orientation, and opacity as the mean value of Gaussians to generate the new hyphae; half the length is the mean of a Gaussian giving the lengths of the split. The probability of selecting the i-th of n_{split} hyphae and splitting it is given by

$$q_{\mathsf{split}}(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2 \,|\, \boldsymbol{h}_i) = \frac{k_{\mathsf{split}}}{n_{\mathsf{split}}}\, g_{\boldsymbol{h}}(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2 \,|\, \boldsymbol{h}_i)\,. \tag{3.28}$$

The density $g_{\boldsymbol{h}}(\cdot)$ comprises the product over split hypha width, length, orientation, and opacity subdensities. For example,

$$g_w(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2 \,|\, \boldsymbol{h}_i) = \frac{\sigma_w^{-2}}{2\,\pi}\, \exp\left[-\frac{(\tilde{w}_1 - w_i)^2 + (\tilde{w}_2 - w_i)^2}{2\,\sigma_w^2}\right] \tag{3.29}$$

$$g_l(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2 \,|\, \boldsymbol{h}_i) = \frac{\sigma_l^{-2}}{2\,\pi}\, \exp\left[-\frac{(\tilde{l}_1 - l_i/2)^2 + (\tilde{l}_2 - l_i/2)^2}{2\,\sigma_l^2}\right], \tag{3.30}$$

with the orientation and opacity subdensities constructed in the same way as the width (3.29). The merge move is defined analogously to 3.28 with the average width, orientation, and opacity of two hyphae used as Gaussian means for the merged hypha. The length is Gaussian distributed according to the sum of the merged lengths. We denote this proposal $q_{\mathsf{merge}}(\tilde{\boldsymbol{h}} \,|\, \boldsymbol{h}_i, \boldsymbol{h}_j)$ and define it in the same way as above.

Both sets of birth/death and split/merge jump moves are defined over spore ellipsoids with slight modifications from hypha cylinders. The proposal distribution for switching one type of substructure from another utilizes previously defined prior parameters and the fact that the geometric parameters have the same interpretation for spore ellipsoids as hypha cylinders.

The proposal distributions for adding, removing, and switching structure generate candidate topology changes. We evaluate how good the proposals are and either accept or reject them based on the reversible-jump Metropolis-Hastings algorithm (Green, 1995, 2003). As with the diffusion moves, the jump move acceptance probabilities are constructed to maintain the detailed balance condition. Thus, the posterior will be the stationary distribution of the trans-dimensional Markov chain followed by the sampler. For the a hypha birth move, we define the reversible-jump acceptance probability to be

$$\alpha_{\mathsf{birth}}\Big(\boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)}\Big) = \min\left\{1,\ \frac{p(\boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)} \mid \mathcal{I})}{p(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})} \mid \mathcal{I})} \frac{r(\mathsf{death})}{r(\mathsf{birth})} \times \frac{1}{q_{\mathsf{birth}}(\tilde{\boldsymbol{h}} \mid \boldsymbol{a}_j)} \left| \frac{\partial(\tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)})}{\partial(\boldsymbol{\theta}^{(\boldsymbol{n})}, \tilde{\boldsymbol{h}})} \right| \right\}. \tag{3.31}$$

The proposed hypha is directly inserted into the model, so the change in dimensionality is a one-to-one mapping from $(\tilde{\boldsymbol{h}}, \boldsymbol{\theta}^{(\boldsymbol{n})}) \rightarrow \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)}$, making the Jacobian 1. Let n_{birth} be the number of substructures that can be extended apically with a proposed hypha. Then the apical hypha birth move (3.31) reduces to

$$\alpha_{\mathsf{birth}}\Big(\boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)}\Big) = \min\left\{1,\ \frac{n_{\mathsf{birth}}\,\nu_h}{n_h+1} \frac{L(\mathcal{I} \mid \boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)})}{L(\mathcal{I} \mid \boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})})} \times \frac{r(\mathsf{death})}{r(\mathsf{birth})} \prod_{i=1}^{\boldsymbol{n}} \chi(\boldsymbol{a}_i \not\vdash \tilde{\boldsymbol{h}}) \right\}. \tag{3.32}$$

The normalization constant and density in the hypha proposal (3.27) cancel with the prior. A test for intersection remains to guard against adding a new spore that overlaps with any others already in the model.

The hypha death move complements the birth move by proposing to prune a hypha that does not have any attaching substructures, e.g., one that was just birthed. Of these n_{birth} candidate hyphae, one is randomly selected with uniform probability and removed from the model. The acceptance probability for a hypha death move is the inverse of (3.32), but with the unnecessary intersection test removed,

$$\alpha_{\mathsf{death}}\Big(\boldsymbol{n}-1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}-1)}\Big) = \min\left\{1,\ \frac{n_h}{n_{\mathsf{birth}}\,\nu_h} \frac{L(\mathcal{I} \mid \boldsymbol{n}-1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}-1)})}{L(\mathcal{I} \mid \boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})})} \frac{r(\mathsf{birth})}{r(\mathsf{death})} \right\}. \tag{3.33}$$

The acceptance probabilities for lateral hypha and spore birth/death moves are constructed in the same fashion.

The split move for a hypha proposes two new hyphae of similar size to replace it. We use the split proposal (3.28) to conditionally generate the split hyphae. We further include the merge proposal probability $q_{\mathsf{merge}}(\cdot)$ in the acceptance ratio to match dimensions,

$$\alpha_{\mathsf{split}}\left(\boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)}\right) = \min\left\{1,\ \frac{p(\boldsymbol{n}+1, \tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)} \mid \mathcal{I})}{p(\boldsymbol{n}, \boldsymbol{\theta}^{(\boldsymbol{n})} \mid \mathcal{I})} \frac{r(\mathsf{merge})}{r(\mathsf{split})} \times \frac{q_{\mathsf{merge}}(\boldsymbol{h}_i \mid \tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2)}{q_{\mathsf{split}}(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2 \mid \boldsymbol{h}_i)} \left| \frac{\partial(\tilde{\boldsymbol{\theta}}^{(\boldsymbol{n}+1)}, \boldsymbol{h}_i)}{\partial(\boldsymbol{\theta}^{(\boldsymbol{n})}, \tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2)} \right| \right\}. \tag{3.34}$$

As with the death move, the merge acceptance probability is the reciprocal of its complementary move ratio. We create similar merge/split moves for the spore substructure.

The switch moves transitions a hypha to a spore and back again. For this proposed change, we use the previous substructure as the mean value for a proposal of the other. All attachments are transitioned, and the acceptance probability resembles those from the diffusion moves.

3.5.3 Data-driven MCMC

The spores in the data are typically much larger than the hyphae and more darkly pigmented (Figure 2.1). However, unlike in Chapter 2, where we fit independent ellipsoids, here we fit connected cylinders and ellipsoids to hyphae and spores. As such, we observe that a hypha is often incorrectly fit to a spore and vice-versa. While we maintain a switch move in the sampler to transition a substructure to another type, proposing the correct substructure can require many iterations of the sampler. Thus we improve both the birth and switch moves by analyzing the data before inference and building data-driven proposals for spores. We use a similar process for proposal distribution construction as in Chapter 2 and Tu et al. (2005, 2002); Zhu et al. (2000).

The replacement proposal distribution is similar to what we used for independent spore detection in Section 2.6. We use a gradient-based surface point detection algorithm and a coarse Hough transform for ellipsoids to obtain rough estimates of spores in the data. The estimates are collected into a spore likelihood table, which we use as the new

	α		β		γ	
	mean	stdev	mean	stdev	mean	stdev
$\mathcal{A}_1$	0.99	0.001	0.91	0.08	0.75	0.26
$\mathcal{A}_2$	0.84	0.14	0.68	0.4	0.64	0.24

Table 3.1: Mean PSF model parameters inferred from the *Alternaria* data from 10 random starting states. The larger variance in the parameters for the second set is most likely from not fitting as much structure.

proposal distribution. Although the estimates from the Hough transform are coarse, it is tolerable because diffusion moves in the sampler will perfect the fit of proposed spores.

We also used data-driven methods in the sampler to speed-up the initial estimate of the base structure in the model. We follow the assumption that the imaged growth of *Alternaria* begins at the bottom of the microscopic image stack and proceeds upward.

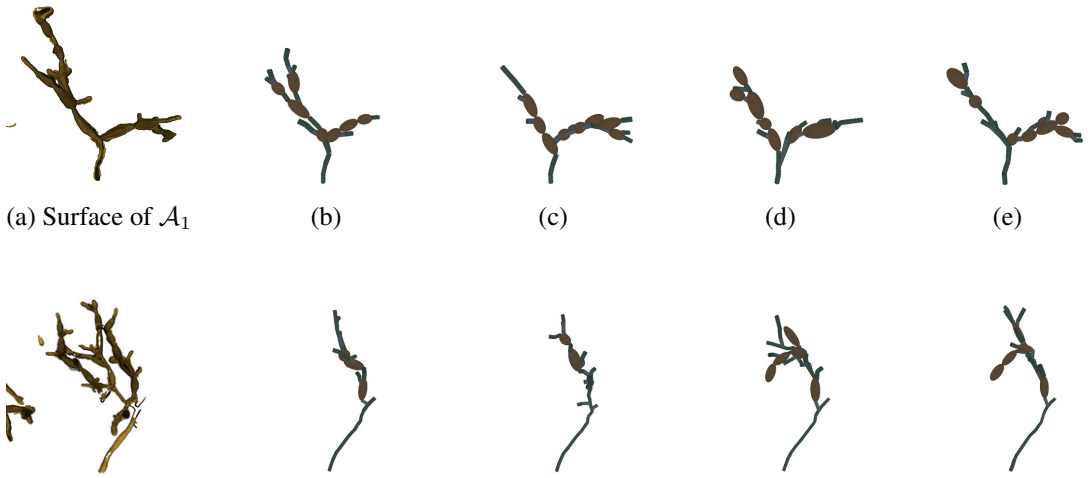
3.6 Results

We evaluated the effectiveness of the model sampler on *Alternaria* image sets $\mathcal{A}_1$ and $\mathcal{A}_2$, shown in Figure 2.1. $\mathcal{A}_1$ is composed of 102 images of size 800×800 pixels and $\mathcal{A}_2$ has 82 images of size 700×700. Since the data are so large, we down-sample them along rows and columns to 20% of their original size. However, since the number of images in each stack is already disproportionately small, we did not decrease the resolution in depth.

We ran the sampler from 10 random starting states on both data sets, each for $20,000$ iterations. Figure 3.3 shows four of the ten models fit to each data set. The sampler had a more difficult time fitting the structure in $\mathcal{A}_2$; a narrow lateral hypha spawned very large areas of structure. With more iterations we would expect to begin to fit more of it.

The average inferred background intensity for $\mathcal{A}_1$ and $\mathcal{A}_2$ was 0.74 and 0.72 with a negligible standard deviation. Table 3.1 gives the inferred PSF model parameters for the data sets. The PSF parameters for $\mathcal{A}_2$ have larger variance because not as much structure was fit in the images as $\mathcal{A}_1$.

Figure 3.4 shows two images from $\mathcal{A}_1$ at different depths compared to corresponding



(a) Surface of $\mathcal{A}_1$ (b) (c) (d) (e)

(f) Surface of $\mathcal{A}_2$ (g) (h) (i) (j)

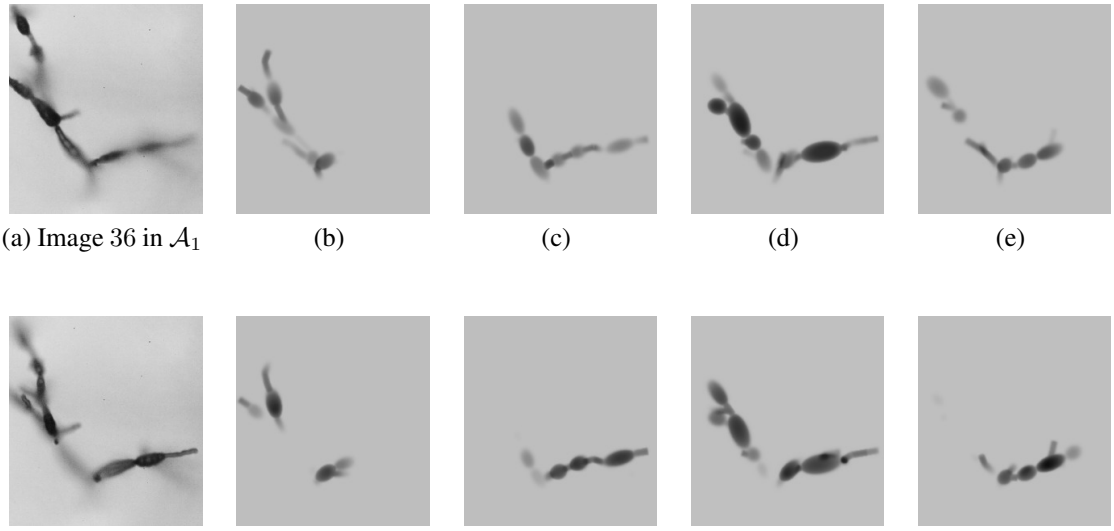
Figure 3.3: The sampler was run on data sets $\mathcal{A}_1$ and $\mathcal{A}_2$ from 10 random starting states. The first row shows a rendering of surface points from the gradient-based detection for data-driven proposals (a) from set $\mathcal{A}_1$ and four of the inferred models (b)–(e). The second row shows similar results for set $\mathcal{A}_2$ (f)–(j). We are clearly fitting *Alternaria* structure in the data. If we continue to run the sampler, more of the structure would be fit, particularly in the case of $\mathcal{A}_2$.

inferred model-scene images. We construct the model-scene images by optically sectioning the *Alternaria* model and convolving it with the point-spread function. From these images we observe that simultaneously fitting structure and imaging models closely resembles the image formation process, enabling us to obtain a more accurate fit to the data.

3.7 Conclusion

Learning the structure of an object is one of the first steps in trying to understand its function. Biologists recognize this fact and conduct many experiments that require analyzing images of microscopic structures. We have shown that combining a grammar-based specimen model with an imaging model is useful to automatically learn the 3-D form of biological specimens in microscopic image stacks. From these inferred models, we can extract quantitative information, and even learn about categories or species of the structure. This is in contrast to simply counting pixels in the image plane occupied by the projected 2-D shape of a specimen. In our approach, we understand geometric structure in 3-D and can give quantitative information about its size, shape, and topological configuration.

In the following chapters, we broaden the approach presented here to modeling more general categories of man-made objects composed of 3-D blocks, such as furniture. We continue to build upon our basic idea of representing objects as a composition of 3-D geometric primitives that are independent of the imaging system viewing them. We further show how categories of object structure, comprising 3-D shape statistics and topologies, can be learned from data. Although not pursued in the biological structure work described here, the methods we develop for learning object categories could be applied to *Alternaria* for learning species-specific parameters across microscopic stacks of images. This would enable us to quantify the defining traits of a species and to automatically classify which species an instance of *Alternaria* belongs to.



(a) Image 36 in $\mathcal{A}_1$ (b) (c) (d) (e)

(f) Image 48 in $\mathcal{A}_1$ (g) (h) (i) (j)

Figure 3.4: Simultaneously modeling the structure and imaging system more accurately explains blurred microscopic images. Row one shows image 36 of $\mathcal{A}_1$ compared with images at the same depth from the inferred model-scenes. Image 48 in the stack is shown in the second row. The generated model-scene is the optically sectioned images of a fit *Alternaria* and imaging system. Each column shows images generated by the model for $\mathcal{A}_1$ corresponding to the same column of figure 3.3, e.g., images 3.4b and 3.4g are from the model in figure 3.3b. The model-scene images for $\mathcal{A}_2$ have similar results.

CHAPTER 4

Fitting 3-D Models of Object Structure to Single View Images

4.1 Introduction

In Chapters 2 and 3 we presented an approach for fitting three-dimensional models of connected geometric primitives to image stacks of *Alternaria* captured under a microscope. We separated the object structure model from the imaging system and inferred both from data simultaneously. In this chapter and the next, we continue modeling object structure with a collection of 3-D geometric primitives, but transition from biological structure toward more general, man-made objects composed of block-like parts, such as furniture. Moreover, rather than inferring structure from stacks of images comprising a 3-D data set, we learn structure models from single view 2-D images captured by a standard camera. The basic idea of our approach is to fit connected 3-D blocks to detected image features of furniture objects, such as edge points. Figure 4.1 shows an example image of a furniture object, the detected edges used for structure inference, and the 3-D blocks representing object structure fit to the edges.

Although the data and domain of structure are different for furniture objects and biological structure, much of the approach presented here builds on our previous work for modeling and inferring *Alternaria*. We continue to separate our 3-D representation for objects from our model of the imaging system. This enables separating the variation of object structure from the variation of the camera viewing it. We additionally continue to model objects as comprising assemblages of 3-D geometric primitives. However, in contrast with *Alternaria* where we represented hypha and spore substructures with cylinders and ellipsoids, here we use cuboids, or blocks, for furniture parts. We also extend our generative representation for object, camera, and image data, and further broaden our Bayesian statistical inference framework for fitting these models to data simultaneously. Thus, our approach for understanding furniture structure is, to a high degree, related to

(a) Input image

(b) Detected edge map

(c) Projected 3-D model contours

Figure 4.1: Example image of a furniture object (a), its detected edge map (b), and the block-based structure and camera models fit to the edges (c). We follow a statistical, generative representation for the detected edge points in (b) from the projected model contours in (c). The representation generates edge point distances and orientations from the object model with Gaussian error and accounts for noisy edge points and missing detections.

fitting *Alternaria*. The primary difference is that here we use single view 2-D images and model man-made, arbitrarily constructed objects.

The goal of our work with furniture objects is to learn models of structure from collections of single view images. Specifically, we want to learn 3-D geometric primitive assemblages for a category of objects and statistics over the shape and relative position of those blocks. One challenge is developing good approaches for generating these models. A second challenge is developing effective and scalable inference methods to learn model details from readily available training data. For this chapter we focus on the latter—inferring the size, position, and pose of a 3-D block model and the camera capturing it from single view images. In Chapter 5 we present our approach for learning assemblages of blocks that represent an object category.

Given the form of a furniture category model, such as tables, the problem we face is fitting 3-D blocks of the model to a 2-D image without knowing the camera parameters. This is a difficult inference problem. In this chapter, we develop a robust approach for simultaneously fitting a 3-D block model and camera parameters. We accomplish this with an effective likelihood that connects geometric structure to image data. We also introduce a novel sampling approach to fit the object and camera models to data. As in many other inference problems, we need to deal with a large number of local maxima in the posterior over parameters. Our problem is further complicated by the intricate relationship between object and camera parameters. For example, changing the object size and focal length leads to somewhat similar effects in the image. Thus reasonably adjusting the perspective effect in a sampling paradigm requires having the notion of variable correlation built into the sampler. If the sampler simply proposes changes to one of these two parameters independent of the other, or a random combination of them, then the proposal is likely to be rejected.

To effectively search the parameter space over object and camera models, we combine Metropolis-Hastings and stochastic dynamics[1] MCMC sampling algorithms (Sokal, 1989; Neal, 1993; Andrieu et al., 2001; Liu, 2001; Bishop, 2006). We observe that each sampler has advantages in exploring our parameter space, and combine the two in such a way as to harness the benefits of both. For the case of stochastic dynamics, we build two types of samplers: Langevin (Neal, 1993) and Hyperdynamics (Voter, 1997a). The dynamics are based on gradient information over our posterior and excel at following correlated regions of parameter space. To compute the gradient information for stochastic dynamics, we use numerical differentiation which can be expensive. The Metropolis-Hastings (MH) algorithm is significantly faster at proposing samples and has the ability to traverse large regions of parameter space quickly. One issue with the MH algorithm, however, is that proposal distributions often assume independence among parameters, so acceptance of a move through parameter space with high correlation can require many proposals.

[1] We refer to stochastic dynamics in a similar sense as Neal (1993), which is related to molecular dynamics, and more broadly, to statistical physics.

Our specific combination of sampling algorithms achieves broad exploration of model parameter space and fast convergence to regions of high probability. We follow Langevin dynamics (Neal, 1993) to locate local maxima in our posterior, and then use subsequent moves by the Langevin sampler to estimate the covariance matrix of the local region of the posterior distribution. We then switch to Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970) moves based on that covariance matrix, which is an effective way to leave the region of that local maxima by approximating parameter dependencies and increasing exploration in directions of high variance. Finally, to further consider maxima that MH might be slow to visit, we occasionally transition to Hyperdynamics sampling (Voter, 1997a). We find this combination of sampling approaches to be effective, overcoming some of the limitations of each individual sampler, and we expect that it is applicable to a range of problems.

We demonstrate the effectiveness of our approach using a simple model for a basic furniture object, tables. We are able to fit a basic table model to all images in a set of 32 single view images of tables using the sampling configuration. We emphasize that while the model was chosen to be especially simple for studying the inference problem, the method was developed to apply generally.

4.1.1 Related work

Many of the technical problems addressed in this chapter on sampling structure are related to the work of Sminchisescu and Triggs (2001, 2002, 2003) in the context of tracking human figures. Similar to our situation, a strong 3-D model, together with standard projective geometry, leads to a sampling situation where the variables are significantly correlated and sampling does not appear viable without addressing this. While tracking human pose with a particle filter (Sminchisescu and Triggs, 2001, 2003), they introduce covariance biased sampling for an already parametrized likelihood. Here we extend this idea to where the likelihood is far less understood. In particular, we take advantage of the Langevin sampling results to estimate the local structure of the likelihood, and then use that for Metropolis-Hastings exploration of the likelihood. We also extend a another sampling application of Sminchisescu and Triggs (2002) to vision, namely Hyperdynam-

ics (Voter, 1997a).

Our work relates to a large body of work on model based vision, where determining the pose of a known object in an image is a well known problem (Binford, 1971; Brooks, 1981; Pentland, 1987, 1990; Lowe, 1987, 1991; Huttenlocher and Ullman, 1990), and doing so is a special case of determining an unknown camera. We go beyond that here by developing an approach for more general imaging systems, and models that require significant variation to capture a category.

4.2 Structure and imaging model

We describe in this section our model for 3-D object structure in an image and the camera capturing it. We consider image edges as generated by the current model hypothesis and projected by the hypothesized camera. Although edges are just one of many features available, they provide much of the structural form of an object. Taking a Bayesian approach we reverse this forward model, and from edges detected in an image, simultaneously fit the most likely 3-D object model and camera to have generated them.

Our goal is to learn the structural form of many objects. In this chapter, however, we focus on a simplified representation for a a single 3-D table object composed of blocks. In this way, we reduce the complexity from learning arbitrary structure models to developing algorithms for inferring a given 3-D structure model from a 2-D image. In chapter 5 we utilize the ideas and algorithms developed here and give an approach for learning structure models of multiple object categories. In what follows, we first describe our simple example model for a table, followed by the details of our camera abstraction, and then the image formation process and the corresponding likelihood model.

4.2.1 Table model

We use a simple parametrized model that encodes the structural form of a table furniture object. Thus, for this chapter, we specify the model topology by hand and construct tables from cuboid polyhedra, or blocks. An advantage of using blocks is that they generate edges corresponding to potential edges in an image under many different lighting con-

ditions. The table model comprises a set of five such blocks: its surface and four legs. We constrain the legs to be equally sized, of square width and positioned under the table symmetrically with respect to the length and width of the surface. The position of the table is specified as the projected center point $p_o \in \mathbb{R}^3$ of the surface onto the ground (x, y-plane). The rotation of the table is about a vertical axis from its position through the center of the surface.

Although our model topology is specified in advance, it is highly variable. A total of 10 parameters enable us to fit it to any number of possible configurations, potentially even good fits to objects only resembling tables. The parameters for the table include the size of a surface block w, h, l and the square leg thickness t; the symmetric position of the legs under the surface with respect to the width d_w and length d_l; and a rotation angle φ about a central axis on the surface. We define the structure parameters of a table as

$$\boldsymbol{s} = (p_o, \varphi, w, h, l, t, d_w, d_l) \tag{4.1}$$

Non-negative ranges are defined over each of the size parameters, and the rotation angle varies over $[-\pi, \pi]$. Figure 4.2 shows an example of the table model.

4.2.2 Camera model

In the paradigm of learning about an object from a single view, the full specification of the camera and the object position and scale leads to a redundant set of parameters. We choose a minimal set for inference that retains full expressiveness as follows.

Without a priori information we are unable to distinguish the actual size of an object from its distance to the camera. For this reason we constrain the camera to be at a fixed distance from the world origin and accept knowing the size of an object up to a scaling factor. If at some point we learn what the scaling factor is, we would be able to plug this in and know actual sizes and positions of objects in the world.

We assume that objects of interest are variably positioned near the horizontal ground plane ($y = 0$) and constrain the camera to always look at the world origin. Because we allow the object to rotate around its vertical axis, we only need to specify the camera

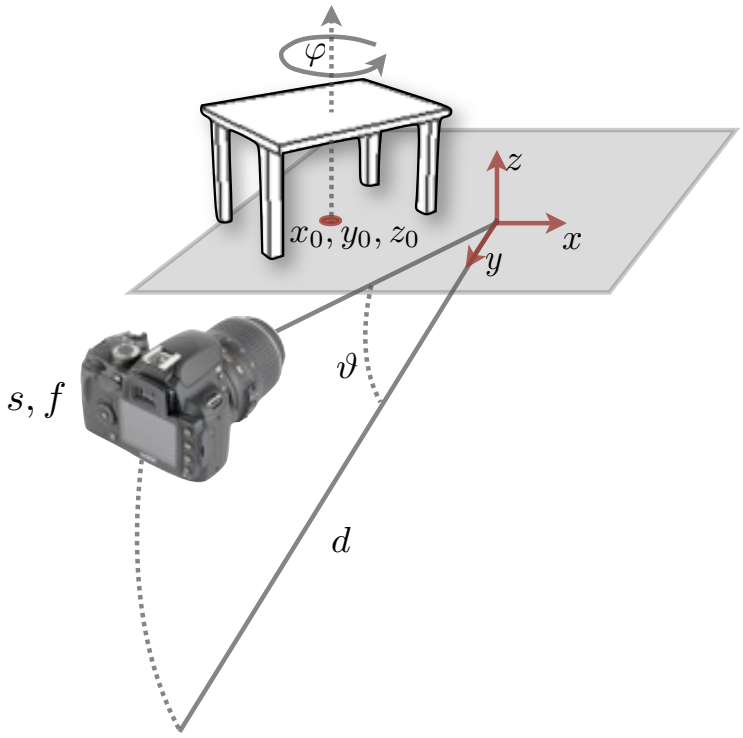


Figure 4.2: The camera model is constrained to reduce the ambiguity introduced in learning from a single view of an object. We position the camera at a fixed distance and direct its focus at the origin; rotation is allowed about the x-axis. Since the object model is allowed to move about the scene and rotate, this model is capable of capturing most images of a scene.

zenith angle, ϑ. Thus we set the horizontal x-coordinate of the camera in the world to zero and allow ϑ to be the only variable extrinsic parameter. In other words, the position of the camera is constrained to a circular arc on the y, z-plane. See Figure 4.2 for an illustration.

We further model the amount of perspective in the image from the camera by parameterizing its focal length, f, and inferring it from the image. The focal length parameter strongly interacts with the scale, s, of the objects in the world. It affects the convergence of parallel lines, however, and specifies a unique image. Our camera instance parameters is then given by

$$\boldsymbol{c} = (\vartheta,\ f,\ s)\ , \tag{4.2}$$

where $\vartheta \in [-\pi/2, \pi/2]$, and $f, s > 0$.

4.2.3 Generative edge model

We model image edges as generated by the projected polyhedron contours of our object representation. Each point along the projected contours statistically generates a detected edge point in the image. Since we detect edges with a standard Canny algorithm (Canny, 1986), the generated detections comprise position and gradient information. Specifically, each point on a projected contour generates an edge point position with some Gaussian error, and a gradient direction similar to the projected contour orientation, also with Gaussian error.

Suppose the correspondences between points on a projected model contour and their generated edge point detections are known. We model the edge points as i.i.d. Gaussian with respect to their distance and orientation away from the model points. We define the distance d_{ij} of a data point from a model point as the perpendicular distance from x_i to the projected model edge that created m_j. Likewise, the orientation angle of a data point from a model point is defined as the angle ϕ_{ij} between the gradient vector $\mathbf{g}_i$ of x_i and the perpendicular direction of the edge $\mathbf{v}_j$ for m_j. Specifically,

$$d_{ij} = \| x_i - m_j \| \tag{4.3}$$

$$\phi_{ij} = \cos^{-1}\left(\frac{\mathbf{g}_i^{\mathrm{T}}\,\mathbf{v}_j}{\|\mathbf{g}_i\|\,\|\mathbf{v}_j\|}\right). \tag{4.4}$$

We limit the maximum distance an edge point can be from a model point to accommodate background clutter in the scene.

For an image and its detected edge points, we test the validity of an object model generating it by rendering the model contours into a scene image and computing a likelihood. We assume independence of observed edge point detections conditioned on the model. For each pixel, we decompose the likelihood into four exclusive terms: the likelihood of a detected edge point, background, a noisy edge point, or a missing detection. The term chosen is decided by the edge detection result and model point generating it, if any. The likelihood of an image given camera and object parameters $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{s})$ is then

$$L(\mathcal{I} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{N} \left[e_\theta(x_i)\, w_{1i} + e_{\mathsf{bg}}\, w_{2i} + e_{\mathsf{noise}}\, w_{3i} + e_{\mathsf{miss}}\, w_{4i} \right], \tag{4.5}$$

where the likelihood of the i-th edge point being generated by the j-th model point is given by

$$e_\theta(x_i) = \frac{\sigma_d^{-1}\,\sigma_\phi^{-1}}{2\,\pi} \exp\left[-\frac{d_{ij}^2}{2\sigma_d^2} - \frac{\phi_{ij}^2}{2\sigma_\phi^2} \right]. \tag{4.6}$$

If a point on a model contour does not match to any edge point in the data, then the pixel positioned under the projected model point is missing an edge point with constant probability e_{miss}; there is no evidence in the data for this piece of the model. A complimentary mismatch occurs when a detected edge point does not match to any model point. We label such detections as noisy edge points occurring with probability e_{noise}. Finally, a pixel contains background with probability e_{bg} when no edge detection is made and no projected model points are nearby.

We assume that the correspondence is known between the i-th edge point pixel and the

j-th model point generating it. With this information, we assign binary values to weights $\mathbf{w}_i$ to indicate the type of correspondence estimated for the i-th pixel. The weights are assigned such that they sum to one for each pixel. In Appendix C we consider the case of avoiding hard, binary assignments to the w_{ij}, and instead allow them to be continuous weights estimated from training data; we learn four different weight vectors, for use in each of the four pixel assignment types. In Chapter 5 we extend the image likelihood model beyond just edge points and further formalize point correspondence estimation.

We combine the likelihood with a prior over the object and camera model parameters defining a posterior distribution for Bayesian inference

$$p(\boldsymbol{\theta} \,|\, \mathcal{I}) = k\, L(\mathcal{I} \,|\, \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\,. \tag{4.7}$$

We model all parameter priors in $\pi(\cdot)$ as Gaussian, with the exception of the rotation angle φ of the object, which we set as uniform over $[-\pi, \pi]$. Although the prior is fairly simple, the resulting posterior distribution is extremely complex, due to the geometric primitive representation and their projection. For this reason direct optimization is out of reach and we use a suite of sampling approaches to fit our model to data. But before describing our sampling strategy, we first show how model and edge point correspondences are estimated.

In order to compute the likelihood and posterior of the image data, we must estimate model and data edge point correspondences. For each model edge point m_r we find the set of candidate data edge points $\{e_i, e_j, e_k, \ldots\}$ whose gradient vectors would generate intersecting paths with m_r if traced through the image. From the candidate set we declare the closest data edge point e_i as corresponding to m_r. At this point we have a one-to-many correspondence between a data edge point and multiple model points, $e_i \leftrightarrow \{m_r, m_s, m_t, \ldots\}$; it is possible for e_i to be closest among all data edge points to more than one model point in the direction of its gradient vector. To resolve this and generate a one-to-one correspondence, we further select the model point from this set that is closest to form the correspondence $e_i \leftrightarrow m_r$. In Chapter 5 we consider the case of allowing a one-to-many correspondence between edge and model points.

The correspondence computation is relatively expensive for each new model hypothesis we evaluate under the likelihood. To speed up this processing we precompute parts of the edge point correspondence and store them. For each edge point detected in the image at program initialization, we trace across the image in the direction of its gradient vector and store potential model point correspondences with precomputed Gaussian distance and discretized angle probabilities. Thus during sampling when we compute the likelihood, we need only render the model image and iterate over the projected model points; we can look-up in constant time the data edge point correspondences and their probabilities, saving a tremendous amount of computation per iteration.

Our inference method of sampling requires the image likelihood to be computed frequently. So we accelerate the object model projection into an image by using offscreen rendering. We call OpenGL rendering routines implemented for pbuffers on modern graphics hardware to generate the projected wire-frame model image of the polyhedron. Hidden, or occluded, edges in the wire-frame model pose a problem, since they do not appear in the data. We remove them using a combination of stencil and depth buffers. For each polygon in the object model, we render its wire frame into the stencil buffer, creating a mask. We then render the filled polygon into the depth buffer, testing against the mask in the stencil buffer. This results in the depth buffer containing only non-wire-frame portions of the polygon. Once this is done for all polygons, we render their wire-frames into the color buffer testing against the depth buffer to occlude hidden lines.

4.3 Sampling

We sample the posterior to find the best set of parameters that fit an image. Given enough iterations, a good sampler converges to the target distribution and an optimal value would be readily discovered in the process. However, our posterior distribution is highly convoluted with many sharp narrow ridges for close fits to the edge points. In our domain, as in many similar problems, standard sampling techniques tend to get trapped in these local extrema for long periods of time. Our strategy is to combine a mixture of sampling techniques with different strengths in exploring the posterior distribution.

In particular, we cycle through Langevin dynamics for fast descent into deep wells and generation of samples used to locally estimate the mode of that well. We then switch to a covariance scaled Metropolis-Hastings sampler that uses the Langevin samples to construct a proposal distribution that samples much more broadly in directions of highest variation, which increases the likelihood of escape from the local extrema. Finally, we use hyperdynamics sampling to bias the posterior function towards areas of transition between extrema, accelerating the chances of moving between extrema. What follows is a description of each sampler.

4.3.1 Langevin dynamics

Langevin sampling is a type of stochastic dynamics that uses an analogy from physical systems to generate representative samples from a target distribution (Neal, 1993; Bishop, 2006). The idea is similar to stochastic gradient descent, where the sampler rapidly descends to local minima, or states, and it spends a much of its time there generating samples. Occasionally it transitions to a new state, with a frequency dependent upon the barrier between the two states.

In this setting, we let the negative log of the posterior (4.7) represent the potential energy function in a hypothetical Hamiltonian system, with our parameters $\boldsymbol{\theta} = (\boldsymbol{c}, \boldsymbol{s})$ comprising the position variables and an introduced, artificial momentum $\mathbf{r} = d\boldsymbol{\theta}/d\tau$ in phase space. We also define a kinetic energy for the introduced momentum $\mathbf{r}$, so that

$$E(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta} \,|\, \mathcal{I}) \tag{4.8}$$

$$K(\mathbf{r}) = \frac{1}{2} \sum_{i=1}^{N} r_i^2 \,. \tag{4.9}$$

The total energy in phase space is then given by the Hamiltonian $H(\boldsymbol{\theta},\, \mathbf{r}) = E(\boldsymbol{\theta}) + K(\mathbf{r})$, which we use in the canonical distribution over phase space

$$p(\boldsymbol{\theta},\, \mathbf{r}) = \frac{1}{Z_H} \exp\left(-H\left(\boldsymbol{\theta},\, \mathbf{r}\right)\right) \,. \tag{4.10}$$

Applying a force to a particle in this system is equivalent to updating its position by a change in momentum, which is given by the negative gradient of the potential energy function,

$$\frac{d\mathbf{r}}{d\tau} = -\nabla E(\boldsymbol{\theta})\,. \tag{4.11}$$

The dynamics conserve total energy and volume in phase space, leaving the canonical distribution invariant. We use this fact in a stochastic setting with the Langevin equation to generate samples. Specifically, we discretize time with a fixed step size ϵ and generate samples from the posterior as follows

$$\tilde{\theta}_i = \theta_i - \frac{\epsilon^2}{2}\frac{\partial E}{\partial \theta_i}(\boldsymbol{\theta}) + \epsilon\,\eta_i\ , \tag{4.12}$$

where η_i is sampled from a Gaussian with mean zero and variance one. We estimate the gradient of our posterior with finite differences in each of the model parameters.

While effective, this discretization can introduce large amounts of error and bias in the samples if ϵ is large. Further, if ϵ is selected to be small, the samples degenerate to a random walk behavior. This process is effective, however, for quickly converging to local minima in the energy function and exploring the region once within. Figure 4.3 shows how varying ϵ effects the Langevin dynamics on Müller's potential (Müller, 1980)

$$V(x,y) = \sum_{i=1}^{4} A_i\,\exp\left[a_i\,(x-x_i)^2 + b_i\,(x-x_i)(y-y_i) + c_i\,(y-y_i)^2\right], \tag{4.13}$$

where A, a, b, c, x, y are defined as in (Müller, 1980) with the constants

$$A = (-200, -100, -170, 15) \tag{4.14}$$

$$a = (-1, -1, -6.5, 0.7) \tag{4.15}$$

$$b = (0, 0, 11, 0.6) \tag{4.16}$$

$$c = (-10, -10, -6.5, 0.7) \tag{4.17}$$

$$x = (1, 0, -0.5, -1) \tag{4.18}$$

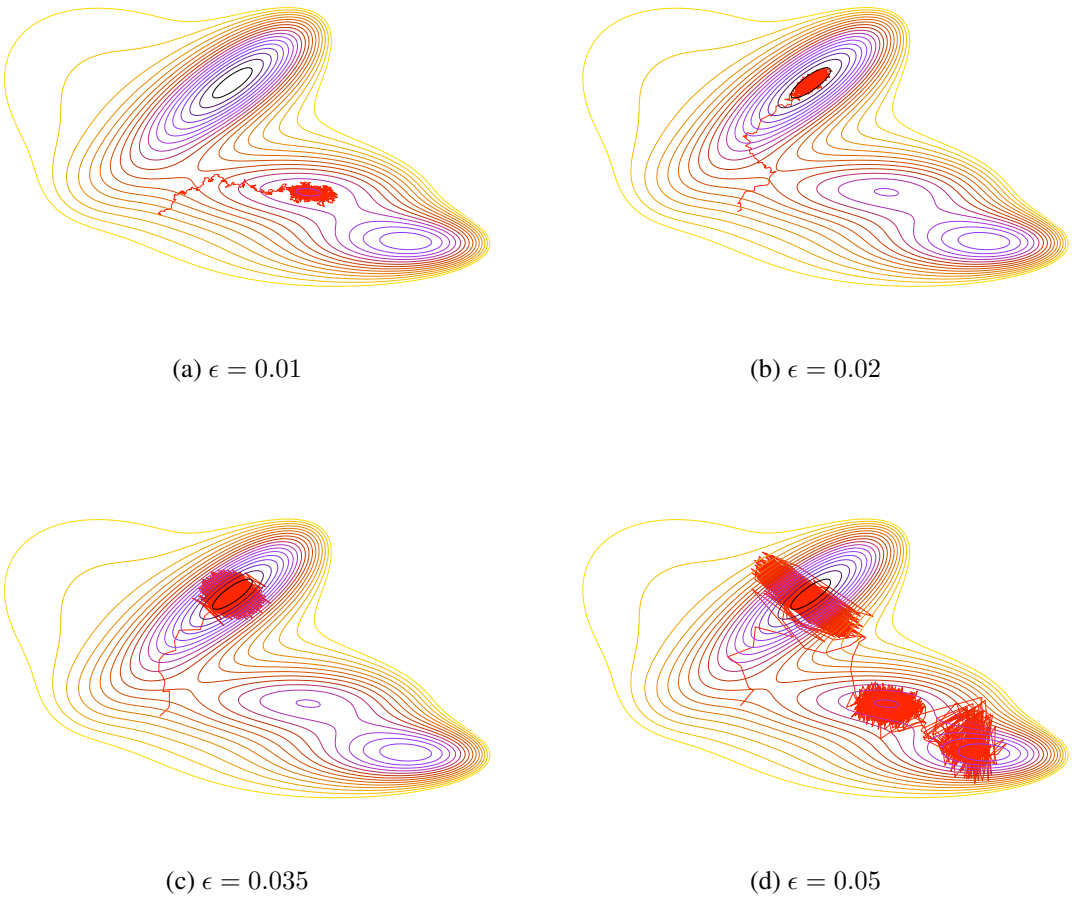
$$y = (0, 0.5, 1.5, 1)\,. \tag{4.19}$$

The example in Figure 4.3 shows how Langevin dynamics are effective at exploring local minima. We should note that Bussi and Parrinello (2007) recently described how to improve a Langevin sampler to more accurately follow the dynamics (4.11) and integrate over phase space. Thus future applications of Langevin sampling should consult this work for potentially improved results. In Chapter 5 we also describe an improved stochastic dynamics algorithm for generating samples to approximate the integration based on the Verlet algorithm (Verlet, 1967, 1968).

4.3.2 Covariance scaled Metropolis-Hastings

When we run the Langevin sampler on our log posterior potential, it converges quickly to a local minimum and spends most of its time exploring this state. Our idea is to use what appears to be a limitation as a way to sample from the distribution more effectively overall. To do this we approximate the mode of the state the Langevin sampler is currently trapped in with a multivariate Gaussian and estimate its parameters from the generated samples. We then use this approximation as a proposal distribution in the Metropolis-Hastings algorithm to efficiently explore correlated parameter directions of maximum variance and quickly jump to new states.

The Metropolis-Hastings (MH) algorithm is a powerful MCMC sampling technique to generate unbiased and representative samples from a target distribution (Metropolis et al., 1953; Hastings, 1970; Neal, 1993; Forsyth et al., 2001; Bishop, 2006). The central



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.02$

(c) $\epsilon = 0.035$

(d) $\epsilon = 0.05$

Figure 4.3: Langevin dynamics sampling on Müller's potential (4.13) under varying time discretization step sizes ϵ. Each panel shows the result of the dynamics after 5000 iterations, initialized from the same point. Notice in (a) the path of the dynamics has a large amount of random walk. As ϵ increases, however, the rate of convergence to local minima increases as well. When the step size becomes large (d), rapid exploration of all minima is possible, but with a significant amount of sample bias.

concept of the algorithm is to propose samples from a distribution $q(\tilde{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta})$, which can be easily sampled, and accept or reject the samples with probability

$$\alpha(\tilde{\boldsymbol{\theta}} \,|\boldsymbol{\theta}) = \min \left\{ 1, \frac{p(\tilde{\boldsymbol{\theta}})\, q(\boldsymbol{\theta} \,|\, \tilde{\boldsymbol{\theta}})}{p(\boldsymbol{\theta})\, q(\tilde{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta})} \right\} . \tag{4.20}$$

Since the sampler follows a Markov chain, and it maintains the detailed balance condition

$$p(\boldsymbol{\theta})\, q(\tilde{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta})\, \alpha(\tilde{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta}) = p(\tilde{\boldsymbol{\theta}})\, q(\boldsymbol{\theta} \,|\, \tilde{\boldsymbol{\theta}})\, \alpha(\boldsymbol{\theta} \,|\, \tilde{\boldsymbol{\theta}}) \,, \tag{4.21}$$

it is sufficient that the sampler will have as its invariant the posterior, assuming there are no zero probability transitions.

Depending on the proposal distribution, however, the MH sampler can take a very long time to explore the state space thoroughly. One could increase the variance in the proposal distributions, but without knowing which parameters have the most variation or correlation in the current state, the rejection rate will likely be higher. Instead of doing this, we construct a Gaussian proposal distribution whose covariance matrix $\boldsymbol{\Sigma}^{(N)}$ is estimated with maximum likelihood from samples previously drawn during N iterations of Langevin dynamics.

The samples generated during the run of the Langevin sampler are highly concentrated around a local mode in the posterior. If we eigen-decompose the covariance matrix estimated from these samples,

$$\boldsymbol{\Sigma}^{(N)} = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}} \,, \tag{4.22}$$

we see that after a number of iterations the principle components $\mathbf{u}_i$ change very little, implying convergence to a good estimate of the local mode the sampler is trapped in. This is illustrated in Figure 4.4, which shows the Frobenius norm of estimates for the eigenvectors of $\boldsymbol{\Sigma}^{(N)}$ successively subtracted at 100 iteration intervals.

Rather than use our estimate of the covariance matrix directly in the MH proposal distribution, we take inspiration from Sminchisescu and Triggs (2001, 2003) and rescale its k largest principle components

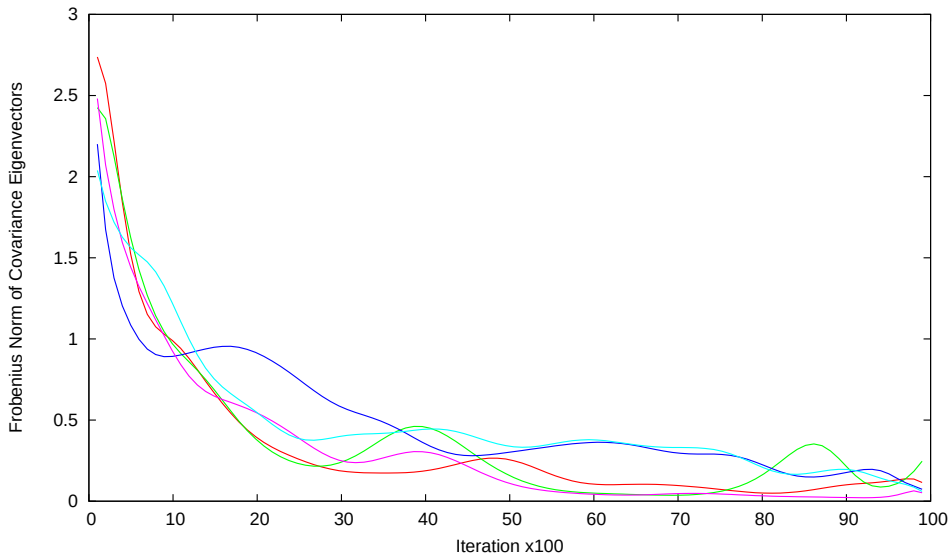


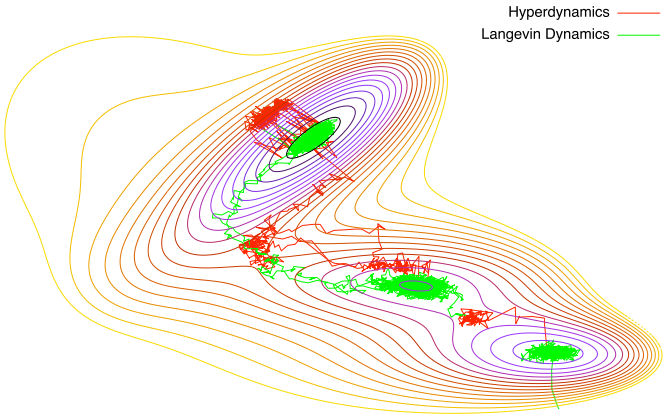
Figure 4.4: Frobenius norm for covariance matrix eigenvectors estimated and subtracted at successive intervals of 100 iterations of the Langevin sampler. Each curve represents starting the sampler from a random state on five different images

$$\hat{\mathbf{\Sigma}}^{(N)} = s \sum_{i=1}^{k} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}} + \sum_{j=k+1}^{D} \lambda_j \mathbf{u}_j \mathbf{u}_j^{\mathrm{T}} \ . \tag{4.23}$$

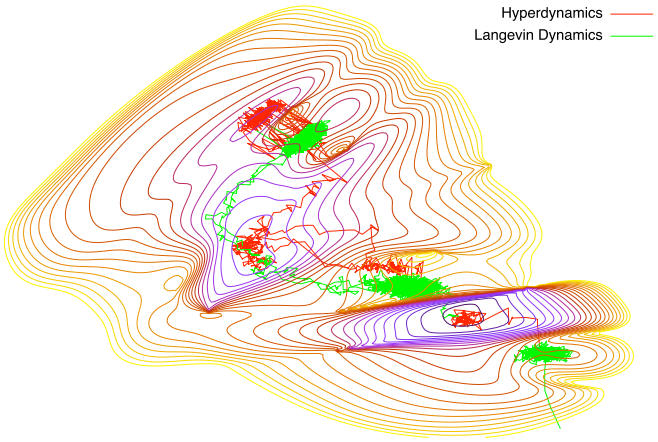
This has the effect of magnifying the variation in directions of most uncertainty in the current state. In our model space, that means correlated parameters representing large changes with little observable difference, i.e., depth, will be sampled more broadly.

4.3.3 Hyperdynamics

Although covariance scaled Metropolis-Hastings has the potential to jump large distances in parameter space with higher acceptance rate, it can still get trapped. This is especially likely to happen when the Langevin dynamics descends into a particularly deep and narrow minima. In this case there is little variation and the estimated Gaussian for the local mode will have extremely small variation. We approach this problem with hyperdynamics, which was first introduced in computational chemistry by Voter (1997a,b) and later introduced for importance sampling in vision problems by Sminchisescu and Triggs (2002). By applying a particular biasing function to the potential energy, we virtually ac-



(a) Müller's potential $V(x, y)$



(b) Biased potential $V(x, y) + V_b(x, y)$

Figure 4.5: Sampling Müller's potential energy function with Langevin dynamics and hyperdynamics. The samplers were cycled eight times, with 1000 iterations of Langevin followed by 100 iterations of hyperdynamics per cycle. The sampler parameters were $\epsilon = 0.02, h = 125, d = 0.05$. Notice how the hyperdynamics concentrates samples around saddle points, while the Langevin dynamics fall back into local minima. Cycling the two algorithms accelerates transitioning between local minima.

celerate the rate of sampling by favoring areas of state transition, enabling more frequent jumps between local minima.

We accomplish hyperdynamics by utilizing the Langevin dynamics abstraction for physical systems and crafting a special bias function, which we add to the potential energy function to favor areas of transition, e.g., saddle points. The challenge of this method is to define a computationally tractable and effective bias function without a priori knowledge of the transition points in the potential.

In order to maintain representative sampling, the bias function need satisfy only one constraint: that it is zero at all points of transition between states. In practice, however, it is sufficient to approximate this constraint with a more local estimate of transition areas defined by saddle points in the potential. This estimate can be described according to the first and second derivatives of the potential. More formally, let $\mathbf{g} = \nabla E(\boldsymbol{\theta})$ and $\mathbf{H}$ be the Hessian matrix with elements $\partial^2 E(\boldsymbol{\theta})/\partial\theta_i\theta_j$. Then the saddle points are those with

$$\lambda_1 < 0 \tag{4.24}$$

$$g_{1p} = \mathbf{C}_1^{\mathrm{T}}\mathbf{g} = 0\,, \tag{4.25}$$

where λ_1 is the lowest eigenvalue of $\mathbf{H}$ and g_{1p} is the projection of the lowest eigenvector $\mathbf{C}_1^{\mathrm{T}}$ of $\mathbf{H}$ onto the gradient. Voter (1997a) applies this definition of a saddle point to define the bias function

$$E_b(\boldsymbol{\theta}) = \frac{h}{2}\left(1 + \frac{\lambda_1}{(\lambda_1^2 + g_{1p}^2/d^2)^{1/2}}\right)\,. \tag{4.26}$$

The function is derived by assuming a sine wave potential with height h and period $2\pi d$ and constructing a function to cancel it; the additive bias raises the value of the potential to the transition points, $h/2$.

Adding the hyperdynamics bias function to the Langevin sampler (4.12) yields a new stochastic dynamics equation

$$\tilde{\theta}_i = \theta_i + \frac{\epsilon^2}{2}\frac{\partial(E + E_b)}{\partial\theta_i}(\boldsymbol{\theta}) + \epsilon\,\eta_i\,. \tag{4.27}$$

Including the bias function in the Langevin equation requires computation of third order derivatives, which directly could be very expensive. Voter (1997a), however, presents numerical methods to estimate λ_1 and g_{1p} so that (4.27) can be computed entirely in terms of first order derivatives. While this makes the computation more efficient, it is still relatively expensive, requiring many evaluations of the likelihood. Fortunately, the number of iterations needed to transition to a nearby saddle point is small; it is analogous to the number of iterations required by the Langevin sampler to reach the bottom of a local minimum. Figure 4.5 shows an example of how cycling Langevin dynamics with hyperdynamics effectively transitions between local minima in Müller's potential (4.13) and biased potential.

By cycling the hyperdynamics sampler with the standard Langevin sampler, we accelerate the rate at which we can explore local minima in the potential energy function. This is particularly true in cases where the Langevin sampler is trapped in a deep narrow regions of the potential with little variation in sampling. When we switch to hyperdynamics for even just a short number of iterations in this case, the minima is inverted just as steeply and we move quickly to a nearby transition area. Furthermore, the parameters explored in a path through the transition area exhibit the minimum amount of change in the potential between states. In our problem of fitting 3-D structure models to an image, this means the hyperdynamics accelerate exploration of parameters that exhibit the smallest amount of change in the model scene affecting the posterior. As with covariance scaled MH, this is most likely in the ambiguous direction of depth.

4.4 Results and Discussion

We evaluated our sampling strategy and models by inferring them on a set of 32 table images. The edges in all the images were detected with the same parametrization, resulting in many edge points that could be considered noise, or in some cases, missing from major portions of table structure. The fitting process for each image was initialized from

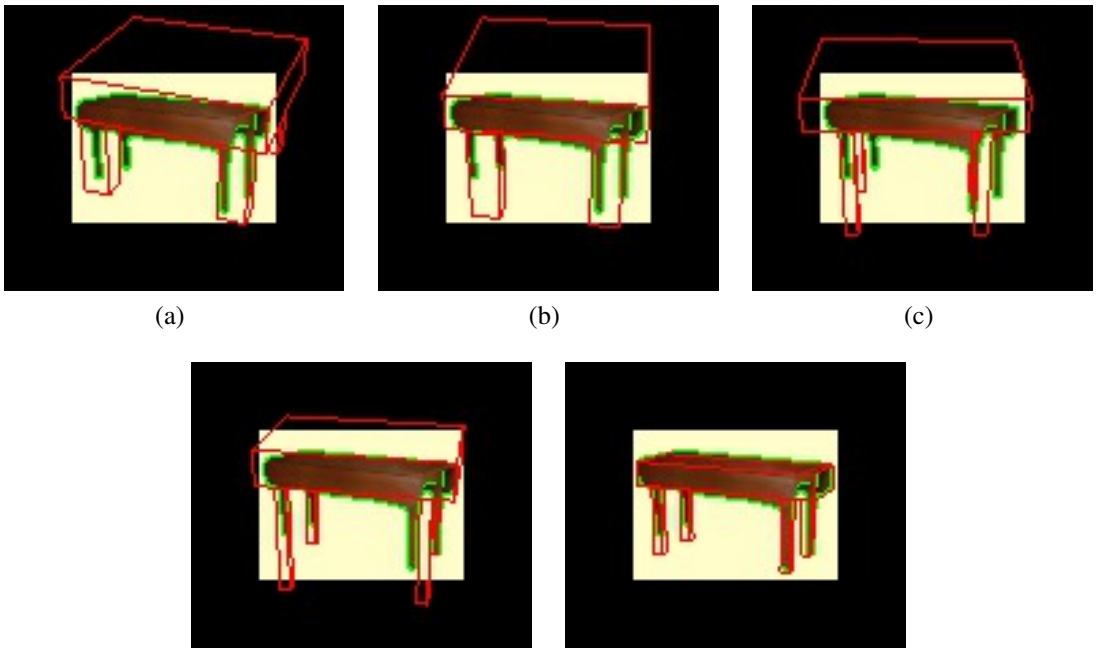
a different random state drawn from our fairly uninformative prior.

We detected the edge points in the data images with a gradient-based Canny edge detector Canny (1986). We further used non-maximal suppression and hysteresis in the edge detector to aid in generating continuous sequences of strong edges in the image. The detector is similar to the one described in Section 2.6.1, but for single 2-D images.

We cycled through each of the Langevin, covariance scaled Metropolis-Hastings and hyperdynamics samplers five times, with most obtaining a good fit after just a couple of cycles. The Langevin and Metropolis-Hastings samplers were run for 10K iterations during each cycle, this was followed by 50 iterations of hyperdynamics. The hyperdynamics sampler takes a significantly longer time for sample generation due to the complex numerical approximations that must be calculated for the bias function. However, 50 iterations was often enough to position the model parameters at a transition point so a new state could be reached. Fig. 4.6 shows a sequence of random samples from the Langevin dynamics throughout the sampler cycling process.

As shown in Figures 4.7 and 4.8, we accurately fit most of the table and camera models to the images. If we continue to run the sampler, the fits continue to improve for all images. The image in the top left corner of Figure 4.7 is particularly interesting because of the poor edge detection that occurred. We observe from this fit that even though a substantial number of edge points are noise, nearly half the table surface edge points are missing, and the back leg has no detection at all, we are still able to make a somewhat accurate fit to this image.

As we stated in the introduction, our overall goal is to learn the form of general 3-D structure models for objects. We believe that the novel inference process we have presented in this paper is a good first step on a process to learning structure models. To some extent we can already reason about the structure of tables; we have fit a highly configurable model and can now consider the statistics of such a model after fitting it to a collection of images. In Chapter 5 we do the same for other objects, in a less specified way, and learn how to discriminate between various classes of structures that share parts and appear similar.



(a) (b) (c)

(d) (e)

Figure 4.6: From left-to-right and top-to-bottom, a sequence of inference for one table image after each cycle of the samplers. The image is drawn randomly from the Langevin sampler. Notice the extensive exploration of parameters affecting the depth of the model being fit. This is due to the covariance scaled Metropolis-Hastings sampling and transition accelerated hyperdynamics

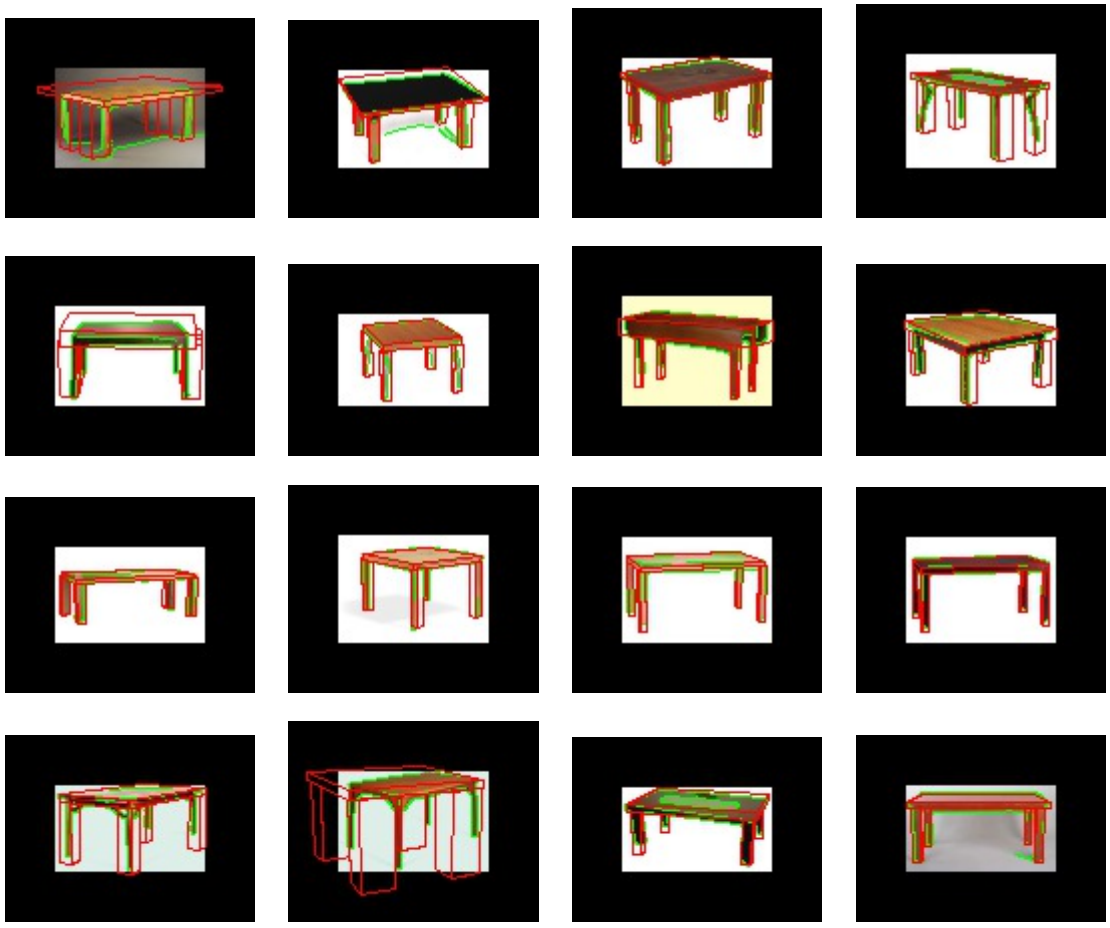


Figure 4.7: First 16 of 32 images of tables fit with our camera and object models. The fit model is rendered in red and the detected edge points are shown in green

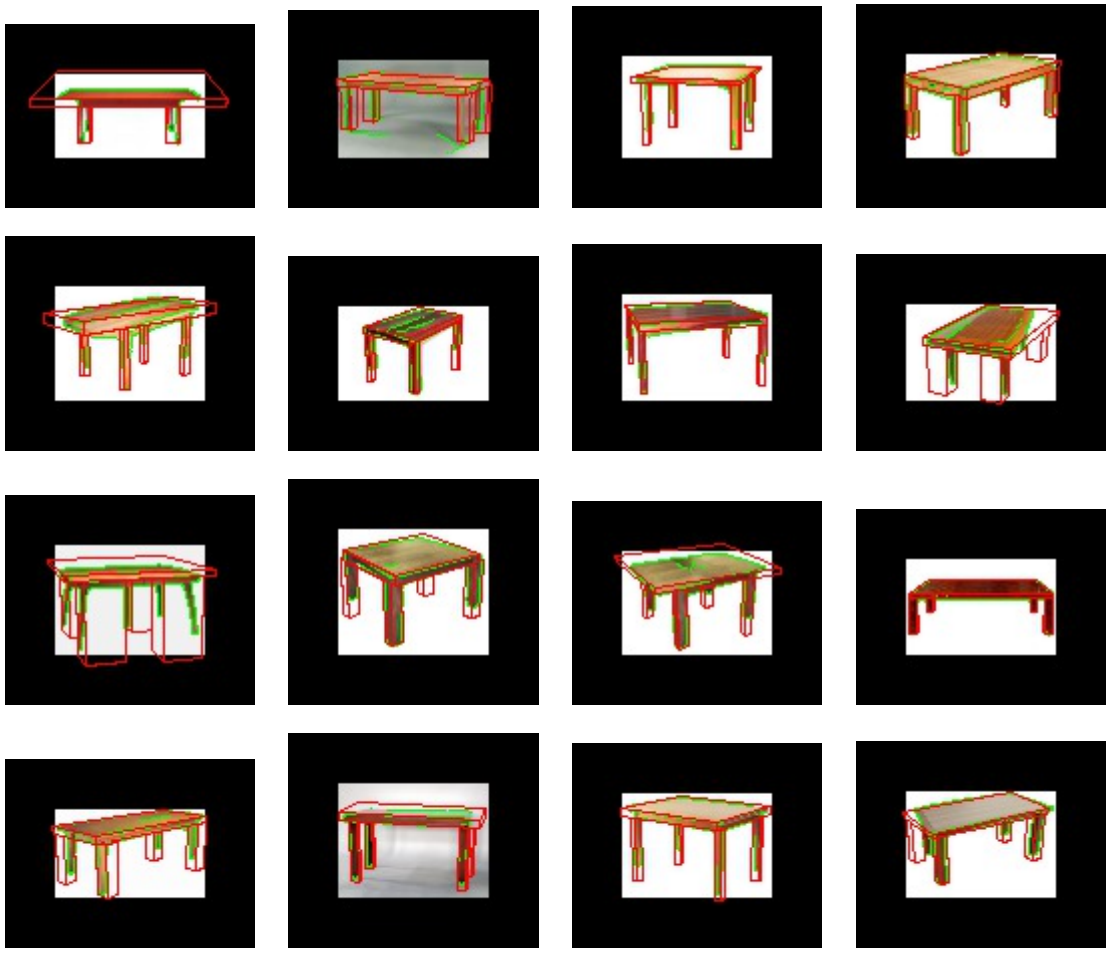


Figure 4.8: Second 16 of 32 images of tables fit with our camera and object models. The fit model is rendered in red and the detected edge points are shown in green

CHAPTER 5

Learning Categories of Object Structure

5.1 Introduction

In this chapter we develop an approach to learn stochastic three-dimensional geometric models of object categories from single view images. We continue the generative approach of previous chapters and build a statistical model that generates object categories, instances and detected image features, such as edge points. Figure 5.1 illustrates the main idea of our generative model. Exploiting such models for object recognition systems enables going beyond simple labeling and opens up opportunities to reason about object function and utility. In particular, we can understand how an object integrates into the scene (perhaps it is an obstacle), how the form of a particular instance is related to others in its category (perhaps it is exceptionally tall and narrow), and how categories themselves are related.

Capturing the wide variation in both topology and geometry within object categories, and finding good estimates for the underlying statistics, suggests a large scale learning approach. We propose exploiting the growing number of labeled single-view images to learn such models. While our approach is extendable to utilize multiple views of the same object, large quantities of such data are rare. Moreover, the key issue we are interested in is learning about category variation, not reconstructing shape for a few object instances. For example, if we are limited to 100 training images, we would prefer to have 100 images of different examples, rather than 10 views of 10 examples.

Representing, learning, and applying statistical geometric properties of objects is potentially simpler in the context of 3-D models. In contrast, statistical models that encode view-based appearance and part configuration statistics must deal with confounding information due to the imaging process. For example, right angles in 3-D can have a wide variety of angles in the image plane. In this case the representations for structure and

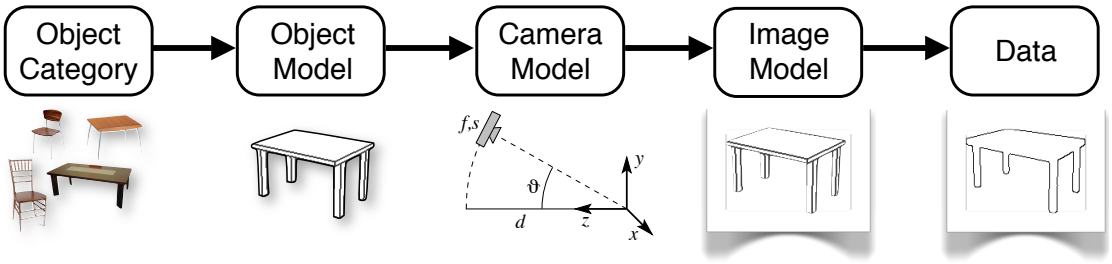


Figure 5.1: Summary of our generative approach to representing 3-D object category models and instances. We begin by sampling an object category comprising a structure topology and part statistics over relative size and position. From this we generate an instance of an object category, such as a table, and capture an image of it under a model of the camera. The image of the projected model block contours then generates edges as they might be detected in image data.

pose variation is the same, implying that the represented geometry is less specific and less informative. By comparison, structure variation encoded in 3-D models is simpler and more precise because the models are solely linked to the object.

To deal with the effect of an unknown camera, we estimate the camera parameters simultaneously while fitting the model hypothesis, similar to Chapter 4. A 3-D model hypothesis is a relatively strong hint as to what the camera might be. Furthermore, we make the observation that structure variation due to standard camera projection is quite unlike typical category variation. Hence, in the context of a given object model hypothesis, the fact that the camera is unknown is not a significant impediment, and much can be estimated about the camera under that hypothesis.

We develop our approach with object models that are expressible as a spatially contiguous assemblage of blocks. We also include in the model a prior on right angles between blocks. We further simplify matters by considering images where there are minimal distracting features in the background. We experiment with images from five categories of furniture objects. Within this domain, we are able to automatically learn topologies. The models can then be used to identify object category in a recognition test using statistical inference. Recognition of objects in clutter is likely effective with this approach, but we have yet to integrate support for occlusion of object parts into our inference process.

We learn the parameters of each category model using Bayesian inference over multi-

ple image examples for the category. Thus we have a number of parameters specifying the category topology that apply to all images of objects from the category. As a side effect, the inference process finds instance parameters that apply specifically to each object. For example, all tables have legs and a top, but the proportions of the parts differ among the instances. In addition, the camera parameters for each image are determined, as these are simultaneously fit with the object models. The object and camera hypotheses are combined with an imaging model to provide the image likelihood that drives the inference process.

For learning we need to find parameters that give a high likelihood of the data from multiple examples. Since we are searching for model topologies, we need to search among models with varying dimension. For this we use the trans-dimensional sampling framework of Green (1995, 2003), as was done in Chapters 2 and 3. We explore the parameter space of a particular dimension in a manner similar to Chapter 4, by combining Metropolis-Hastings and stochastic dynamics MCMC sampling (Sokal, 1989; Neal, 1993; Andrieu et al., 2001; Liu, 2001; Bishop, 2006). As developed below, these two classes of samplers have complementary strengths for our problem, although for efficiency reasons we do not follow exactly the same mix as Chapter 4. We continue, however, to arrange the sampling so that the mixture mantains convergence to the posterior distribution. This ensures that the space will be completely explored, given enough time.

5.1.1 Related work

Most recent work on learning representations for object categories has focused on view-based appearance and part configuration statistics (Fergus et al., 2003; Fei-Fei et al., 2004; Leibe et al., 2004; Sivic et al., 2005; Shotton et al., 2005; Crandall and Huttenlocher, 2006; Leordeanu et al., 2007; Opelt et al., 2008; Ferrari et al., 2009). These approaches typically rely on effective interest point descriptors that are somewhat resilient to changes in view and pose (Berg and Malik, 2001; Belongie and Malik, 2001; Lowe, 2004; Kadir et al., 2004; Ferrari et al., 2008). A second force favoring learning 2-D representations is the explosion of readily available images compared with that for 3-D structure, and thus treating category learning as statistical pattern recognition is more convenient in the

domain of 2-D images. However, some researchers have started imposing more projective geometry into the spatial models. For example, Savarese and Fei-Fei (2007, 2008) build a model where arranged parts are linked by a fundamental matrix. Their training process is helped by multiple examples of the same objects, but notably they are able to use training data with clutter. Their approach is different than ours in that models are built more bottom up, and this process is somewhat reliant on the presence of surface textures. Our work is driven by parametric parts that provide strong cues when they are appropriate. A different strategy proposed by Hoiem et al. (2007) is to fit a deformable 3-D volume to cars, driven largely by appearance cues mapped onto the model. Their choice of modeling in 3-D simplifies a number of issues, and provides for more natural integration with work in understanding scene geometry (Hoiem et al., 2006), as is the case for us. However, our modeling approach is different in that we focus on learning topologies for assemblages of parametrized parts, instead of working with deformation of a single structure. Our interest in learning structure topologies also relates to recent work in learning abstract topologies (Tenenbaum et al., 2006; Kemp and Tenenbaum, 2008) and structure models for 2-D images of objects (Zhu et al., 2006; Zhu and Mumford, 2006) constrained by grammar representations.

5.2 Our approach

From an image collection of an object category, we learn a three-dimensional structure model that probabilistically describes the form and appearance of the category. We accomplish this by inferring instance parameters of object and camera models for each image, and jointly learning across these a category-level organization of object parts (topology) and their distributions. Since an object category typically has multiple, closely related structure topologies, e.g. chairs with and without armrests, we learn sub-categories of structure. This enables us to capture variation within the 3-D structure of object categories and can be used to recognize or detect instances of our model in new images.

In our approach we present a generative model for the 3-D structure of an object, the camera viewing it and the image captured (Figure 5.2). Our representation of an object

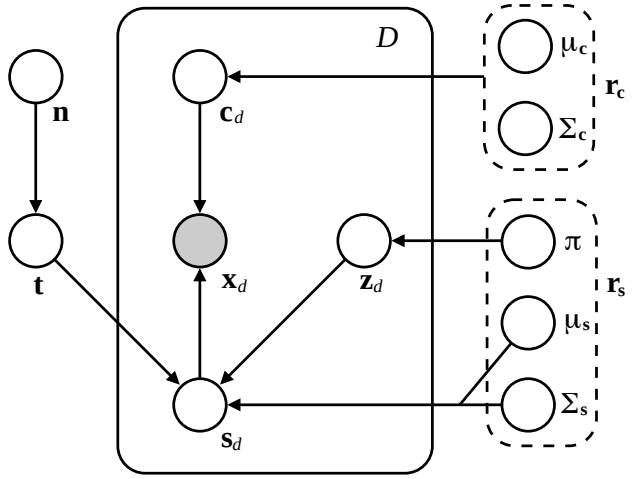


Figure 5.2: Graphical model for the generative approach to images of objects from categories described by stochastic geometric models. The category level parameters are the number of parts, $\mathbf{n}$, their interconnections (topology), $\mathbf{t}$, the structure statistics $\mathbf{r_s}$, and the camera statistics, $\mathbf{r_c}$. Hyperparameters for category level parameters are omitted for clarity. A sample of category level parameters provides a statistical model for a given category, which is then sampled for the camera and object structure values $\mathbf{c}_d$ and $\mathbf{s}_d$, optionally selected from a cluster within the category by $\mathbf{z}_d$. $\mathbf{c}_d$ and $\mathbf{s}_d$ yield a distribution over image features $\mathbf{x}_d$.

comprises a set of 3-D parts linked together by a learned topology. The parts are geometric primitives representing unit pieces of structure and are generated from distribution parameters specific to the object category. The topology of a category characterizes the spatial relationship between parts, which together form the 3-D object model. To accommodate structural variation within an object category, we formulate sub-categories that give rise to object instances which are structurally similar but have some small differences in topology. We further represent the camera capturing the view of an object into an image, enabling an understanding of imaged objects under arbitrary views. Finally, conditioned on the object and camera models, we model independently detected image features, such as edge and surface points, as generated by object parts projected under the camera model. By combining the object, camera and image models, we have a process to generate images of objects that we can use for model inference.

Following a Bayesian strategy, we reverse our forward model and, from detected features in an image, simultaneously fit the most likely 3-D object model and camera to have

generated them. Using the inferred object and camera for a set of images in a category, we learn the form of the category topology and part distributions. In this way we have two types of parameters in our model: per image and per category. We infer both types of parameters simultaneously from a set of training images of an object category. For recognition or detection in a new image, we need only infer the instance parameters.

In describing our model and its process of inference, we first introduce some notation and parameter descriptions. For a single image, we label the corresponding set of structure parts in our object model $\mathbf{s}$ and the camera capturing it $\mathbf{c}$. The topology shared across multiple images generated by the same object category is given by $\mathbf{t}$. We label the similarly shared cluster and distribution parameters for structure sub-categories $\mathbf{r_s}$ and camera distribution $\mathbf{r_c}$. The number of parts in the object model is unknown a priori, making the model parameter set variably sized. For an object with M sub-categories, we denote the number of parts in each object model as $\mathbf{n} = n_1, \ldots, n_M$. Since the dimension of our model depends upon the number of parts, we denote the set of model parameters for one image

$$\boldsymbol{\theta}^{(\mathbf{n})} = (\mathbf{c}, \mathbf{s}, \mathbf{t}, \mathbf{r_c}, \mathbf{r_s}, \mathbf{n}) \ . \tag{5.1}$$

The camera parameters for an image are shared across the sub-categories. We could learn sub-category camera parameters, but multiple structure motifs of an object typically are independent of how they are viewed, e.g., chairs with armrests are similarly viewed as those without. So we label the parameters for a single image under the m^{th} sub-category as a subset of $\boldsymbol{\theta}^{(\mathbf{n})}$,

$$\boldsymbol{\theta}_m^{(n_m)} = (\mathbf{c}, \mathbf{s}_m, \mathbf{t}_m, \mathbf{r_c}, \mathbf{r}_{\mathbf{s}m}, n_m) \ , \tag{5.2}$$

which has a shared camera and generating distribution.

Given a set of D images containing examples of an object category, our goal is to learn the model $\boldsymbol{\Theta}^{(\mathbf{n})}$ generating them from detected features sets $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_D$. In addition to category-level parameters shared across instances, $\boldsymbol{\Theta}^{(\mathbf{n})}$ comprises camera models $\mathbf{C} = \mathbf{c}_1, \ldots, \mathbf{c}_D$ and structure part parameters $\mathbf{S}_m = \mathbf{s}_{m1}, \ldots, \mathbf{s}_{mD}$ for the m^{th}

sub-category generating each image. Our posterior over the parameters then takes the form

$$p\left(\boldsymbol{\Theta}^{(\mathbf{n})} \mid \mathbf{X}\right) = p\left(\mathbf{X}, \boldsymbol{\Theta}^{(\mathbf{n})}\right) / \int p\left(\mathbf{X}, \boldsymbol{\Theta}^{(\mathbf{n})}\right) d\boldsymbol{\Theta}^{(\mathbf{n})} . \tag{5.3}$$

The integral behaves as a constant and is not computed; it is canceled out during the inference process, as we show. The joint density function over the features and parameters, however, is the core of our inference and requires further description.

Since instance parameters $\mathbf{C}$ and $\mathbf{S}$ are bound to the feature data $\mathbf{X}$ in the image set, we separate the joint density into a likelihood and prior

$$\begin{aligned} p\left(\mathbf{X}, \boldsymbol{\Theta}^{(\mathbf{n})}\right) &= p^{(\mathbf{n})}(\mathbf{X}, \mathbf{C}, \mathbf{S}, \mathbf{t}, \mathbf{r_c}, \mathbf{r_s}, \mathbf{n}) && (5.4) \\ &= p^{(\mathbf{n})}(\mathbf{X}, \mathbf{C}, \mathbf{S} \mid \mathbf{t}, \mathbf{r_c}, \mathbf{r_s})\, p^{(\mathbf{n})}(\mathbf{t}, \mathbf{r_c}, \mathbf{r_s}, \mathbf{n}) , && (5.5) \end{aligned}$$

where we use the notation $p^{(\mathbf{n})}(\cdot)$ for a density function corresponding to $\mathbf{n}$ parts. Conditioned on the category parameters, we assume that the D sets of image features and instance parameters are independent, giving

$$p^{(\mathbf{n})}(\mathbf{X}, \mathbf{C}, \mathbf{S} \mid \mathbf{t}, \mathbf{r_c}, \mathbf{r_s}) = \prod_{d=1}^{D} p^{(\mathbf{n})}(\mathbf{x}_d, \mathbf{c}_d, \mathbf{s}_d \mid \mathbf{t}, \boldsymbol{r}_\mathbf{c}, \boldsymbol{r}_\mathbf{s}) . \tag{5.6}$$

This seems a fairly safe assumption; if two images contain examples of an object, then their particular appearances are typically independent. An exception to this assumption, however, includes inadequately modeling sub-category structure variation within a class of objects.

From the independent sets of features and instance parameters in (5.6), we develop a likelihood clustering model over sub-categories of object structure. The feature data and structure parameters are generated by a sub-category cluster with weights and distribution defined by $\mathbf{r_s} = (\boldsymbol{\pi}, \boldsymbol{\mu}_\mathbf{s}, \boldsymbol{\Sigma}_\mathbf{s})$. As previously mentioned, the camera is shared across clusters, and drawn from a distribution defined by $\mathbf{r_c} = (\boldsymbol{\mu}_\mathbf{c}, \boldsymbol{\Sigma}_\mathbf{c})$. We formalize the likelihood of an object, camera, and image features under M clusters as

$$
\begin{aligned}
&p^{(\mathbf{n})}(\mathbf{x}_d, \mathbf{c}_d, \mathbf{s}_d \,|\, \mathbf{t}, \boldsymbol{r}_{\mathbf{c}}, \boldsymbol{r}_{\mathbf{s}}) \\
&\quad = \sum_{m=1}^{M} \pi_m \; \underbrace{p^{(n_m)}(\mathbf{x}_d \,|\, \mathbf{c}_d, \mathbf{s}_{md})}_{\text{Image}} \; \underbrace{p(\mathbf{c}_d \,|\, \boldsymbol{\mu}_{\mathbf{c}}, \boldsymbol{\Sigma}_{\mathbf{c}})}_{\text{Camera}} \; \underbrace{p^{(n_m)}(\mathbf{s}_{md} \,|\, \mathbf{t}_m, \boldsymbol{\mu}_{\mathbf{s}m}, \boldsymbol{\Sigma}_{\mathbf{s}m})}_{\text{Object}} \,.
\end{aligned} \tag{5.7}
$$

We arrive at equation (5.7) by introducing a binary assignment vector $\mathbf{z}$ for each image feature set, such that $z_m = 1$ if the m^{th} cluster generated it and 0 otherwise. The cluster weights are then given by $\pi_m = p(z_m = 1)$. By assuming the feature set and instance parameters to be conditionally independent given the object sub-category, we formally derive the likelihood clustering in (5.7) as follows

$$
\begin{aligned}
&p^{(\mathbf{n})}(\mathbf{x}, \mathbf{c}, \mathbf{s} \,|\, \mathbf{t}, \mathbf{r}_{\mathbf{c}}, \mathbf{r}_{\mathbf{s}}) \\
&\quad = \sum_{\mathbf{z}} p^{(\mathbf{n})}(\mathbf{x}, \mathbf{c}, \mathbf{s}, \mathbf{z} \,|\, \mathbf{t}, \mathbf{r}_{\mathbf{c}}, \mathbf{r}_{\mathbf{s}}) && (5.8) \\
&\quad = \sum_{\mathbf{z}} \prod_{m=1}^{M} p^{(\mathbf{n})}(\mathbf{x}, \mathbf{c}, \mathbf{s}_m, z_m \,|\, \mathbf{t}, \mathbf{r}_{\mathbf{c}}, \mathbf{r}_{\mathbf{s}})^{z_m} && (5.9) \\
&\quad = \sum_{m=1}^{M} p\,(z_m = 1 \,|\, \pi_m) \;\; p^{(n_m)}(\mathbf{x} \,|\, \mathbf{c}, \mathbf{s}_m, \mathbf{t}_m) \; p(\mathbf{c} \,|\, \mathbf{r}_{\mathbf{c}}) \; p^{(n_m)}(\mathbf{s}_m \,|\, \mathbf{t}_m, \mathbf{r}_{\mathbf{s}m}) \,. && (5.10)
\end{aligned}
$$

In this way, we define each independent component of our likelihood for a single image.

For the prior probability distribution over model parameters, we assume category parameter independence, with the clustered topologies conditionally independent given the number of parts in the model. The prior in (5.5) becomes

$$
p^{(\mathbf{n})}(\mathbf{t}, \mathbf{r}_{\mathbf{c}}, \mathbf{r}_{\mathbf{s}}, \mathbf{n}) = p(\mathbf{r}_{\mathbf{c}}) \;\; \prod_{m=1}^{M} p^{(n_m)}(\mathbf{t}_m \,|\, n_m) \; p^{(n_m)}(\mathbf{r}_{\mathbf{s}m}) \; p(n_m) \,. \tag{5.11}
$$

For the category parameters in the camera and structure models, $\mathbf{r}_{\mathbf{c}}$ and $\mathbf{r}_{\mathbf{s}}$, we use Gaussian statistics with weak Gamma priors that are empirically chosen. We set the number of parts in the object sub-categories, $\mathbf{n}$, to be geometrically distributed. We set the prior over edges in the topology given $\mathbf{n}$ to be uniform.

The joint density over image features and model parameters created from the likelihood (5.6) and prior (5.11) describes our generative model and Bayesian approach for learning 3-D object structure in images. Figure 5.2 shows the graphical version of this model and summarizes its parameter relationships. In the next few sections, we detail the object, camera, and image components of this model.

5.2.1 Object model

We model object structure as a set of connected three-dimensional block constructs representing object parts. We account for symmetric structure in an object category, e.g., legs of a table or chair, by introducing compound block constructs. We define two constructs for symmetrically aligned pairs (2) or quartets (4) of blocks. This simplification facilitates learning general category structure while not introducing unnecessary inference overhead. Unless otherwise specified, we will use blocks to specify both simple and compound blocks, as they are handled similarly.

The connections between blocks are made at a point on adjacent, parallel faces. We consider the organization of these connections as a graph defining the structural topology of an object category, where the nodes in the graph represent structural parts and the edges give the connections. We further treat the edges as directed, inducing attachment dependence among parts.

Each block has three internal parameters representing its width, height, and length. Blocks representing symmetric pairs or quartets have one or two additional parameters defining the relative positioning of their sub-blocks. Blocks potentially have two external attachment parameters u, v for each face; we allow one other block attachment per face. We further constrain blocks to attach to at most one other block. giving a directed tree for the topology and enabling conditional independence among attachments. Note that blocks can be visually "attached" to additional blocks that they abut, but representing them as attachments makes the model more complex and is not necessary.

We position the connected blocks in an object coordinate system defined by a point $p_o \in \mathbb{R}^3$ on one of the blocks. Since we constrain the blocks to be connected at right angles on parallel faces, the position of other blocks within the object coordinate system

is entirely defined by p_o and the attachments points between blocks. Despite its simplicity, this model can approximate a surprising range of man made objects.

Combined with a y-axis rotation angle, φ, about its position, our structure model is sufficiently configurable to approximate the form of several furniture categories. For a set of n connected blocks of the form $\mathbf{b} = (w,\, h,\, l,\, u_1,\, v_1, \ldots)$, we denote the object structure model by

$$\mathbf{s} = (p_o,\, \varphi,\, \mathbf{b}_1, \ldots, \mathbf{b}_n)\,. \tag{5.12}$$

The object structure is assumed Gaussian distributed according to $\boldsymbol{\mu_s}, \boldsymbol{\Sigma_s}$ in the likelihood (5.7). Since the instance parameters in the object model are conditionally independent given the category, the covariance matrix is diagonal.

As previously described, the category topology is a set of directed edges in a graph between blocks at their attachment points. For a block $\mathbf{b}_i$ attaching to $\mathbf{b}_j$ on faces defined by the k^{th} size parameter, the topology edge set is defined as

$$\mathbf{t} = \left(i,\, j,\, k :\, \mathbf{b}_i \xleftarrow{k} \mathbf{b}_j\right)\,. \tag{5.13}$$

5.2.2 Camera model

Since our approach for representing the camera capturing an image in this chapter is related to Chapter 4, we use the same model for a camera as in Section 4.2.2. We continue to constrain the camera to always look at the origin of world coordinates and specify its zenith rotation angle about the x-axis with $\vartheta \in [-\pi/2, \pi/2]$. We also parameterize the camera focal length with $f > 0$ and its scale of objects in the world with $s > 0$. Figure 4.2 illustrates the details of how our constrained camera interacts with an object in the scene to provide arbitrary views. Similar to Chapter 4, we specify an instance of the camera with $\boldsymbol{c} = (\vartheta,\, f,\, s)$.

We pursue learning the statistics over camera configurations within an object category. The statistics effectively describes the likely views a particular category is imaged under. For example, tables are usually seen from the top and not the bottom. Thus, as in

the object model, the camera instance parameters in (5.7) are modeled as Gaussian with category parameters $\boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}$.

5.2.3 Image model

We expand the edge-focused image model from Chapter 4 and represent an image as a collection of detected feature sets that are statistically generated by an instance of our object and camera. We model each of the image feature sets as arising from a corresponding feature generator that depends on projected object information. For example, we generate edge points from projected object contours and image foreground from colored surface points. Figure 5.3 illustrates this representation of detected image features. Our likelihood over image feature sets, conditioned on an object and camera model, captures the process by which features are generated and measures how well a model explains their observations.

Given an object and camera, a feature generator stochastically produces the response of a detector at every pixel of an image. Thus each pixel has a non-zero probability of a feature being generated over it by the model, which we assume is independent from all other pixels' chances, given the model. Our image model is then per pixel, and we compute the likelihood of a feature detector's response per pixel given object and camera information.

We formally define the likelihood of image feature sets as a product over per pixel observations. For the d^{th} image with N_d pixels, we assume independence, as previously mentioned, between per pixel feature responses conditioned on the model. We further assume independence among the G different types of generated features detectable in the image. Given the detected feature sets $\mathbf{x}_d = \mathbf{x}_{d1}, \ldots, \mathbf{x}_{dG}$ in the d^{th}, we expand the image component of equation (5.7) to

$$p^{(n_m)}(\mathbf{x}_d \,|\, \mathbf{c}_d, \mathbf{s}_{md}, \mathbf{t}_m) = \prod_{g=1}^{G} \prod_{i=1}^{N_d} f_{\theta g}^{(n_m)}(x_{dgi})\,. \tag{5.14}$$

The function $f_{\theta g}^{(n_m)}(\cdot)$ measures the likelihood of a feature generator producing the response of a detector at each pixel using our object and camera models.

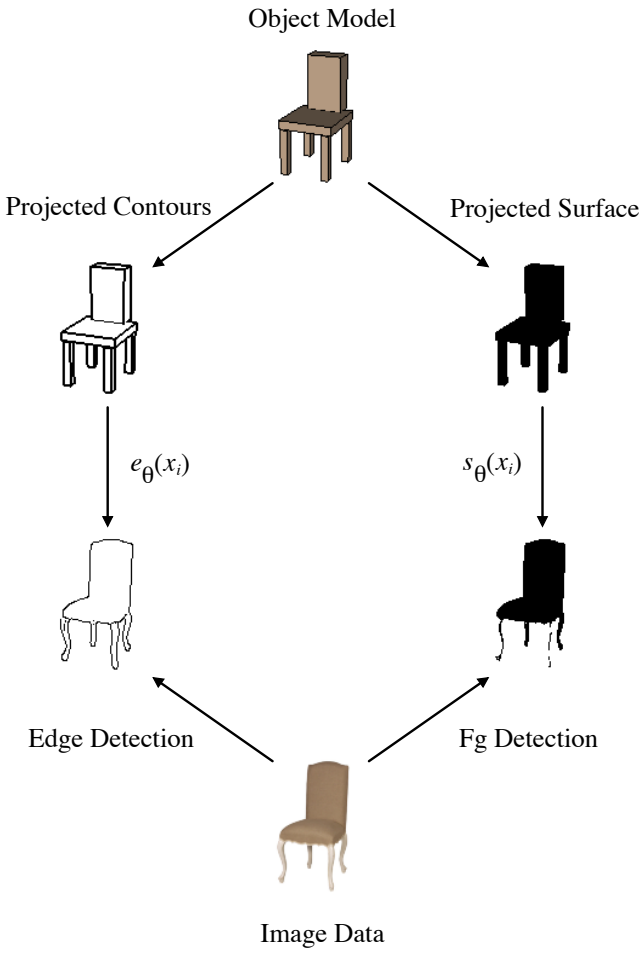


Figure 5.3: Example of the generative image model for detected features. The left side of the figure gives a rendering of the object and camera models fit to the image on the right side. The rightward arrows show the process of statistical generation of image features. The leftward arrows are feature detection in the image data.

It may not necessarily be the case that the G detected feature sets are independent. As in the pixel independence assumption, however, conditioning on the model provides a way to lessen this dependency and simplify our model. We further observe that detected features of different types do not always have strong dependencies. This is particularly true for edge and surface points. Since edge points are located in surface regions of high color transition, and most surface color is not in these regions, it is unlikely a strong dependency exists between a particular surface point and an edge detection. For these reasons we believe it is reasonable to assume such independence.

Using our approach to the image likelihood, we can model many different types detectable features. We currently model edge points and image foreground, as they are straightforward to extract, provide a good representation of object part structure and location, and are readily modeled by our object representation. As Figure 5.3 shows, the projected object contours model detected edge points, and surface points represent the detected image foreground. Our foreground representation is essentially a binary color indicating whether a pixel contains a surface point in the foreground. We could easily extend this to account for more color in the foreground surface points and add another feature generator to the image model. We first describe how we model edge point generation followed by surface point generation.

Edge point generator

Image edge points occur at pixel locations where there is a large change in color relative to nearby pixels. We model edge point location and orientation as generated from projected 3-D contours of our object model. The object contours arise where two or more surfaces meet with different orientation, each having potentially different color or shading. The projected object contour points are positioned in a hypothesized model image and contain orientation information. This representation is consistent with edges detected using gradient-based methods that give edge point pixel locations and a gradient vector indicating edge orientation.

An edge detector gives a response at each pixel and indicates whether an it contains an edge point. Since the feature generator likelihood in (5.14) is computed over all detection

responses in an image, we define the edge generator likelihood as

$$\prod_{i=1}^{N} f_\theta(x_i) = \prod_{i=1}^{N} e_\theta(x_i)^{\mathcal{E}_i} \cdot e'_\theta(x_i)^{(1-\mathcal{E}_i)}\,, \tag{5.15}$$

where the probability density function $e_\theta(\cdot)$ gives the likelihood of a detected edge point at the i^{th} pixel, and $e'_\theta(\cdot)$ is the density for pixel locations not containing an edge point. The two density functions are selected per pixel by an indicator $\mathcal{E}_i$, which is 1 if the pixel is an edge point and 0 otherwise. We have suppressed the image and generator indices d, g and sub-density index (n_m) for clarity.

The edge point density $e_\theta(\cdot)$ is defined over detected edges that have been generated by projected contour points of the object model. We define the i^{th} edge point generated from the j^{th} model point to have some Gaussian distributed displacement d_{ij} in the perpendicular direction of the projected model contour. We further define the gradient direction of the generated edge point to have Gaussian error in its angle difference ϕ_{ij} with the perpendicular direction of the projected contour. Thus, we define the likelihood to be the product of two Gaussians, assuming independence. Let m_j be the known model point to have generated x_i, then

$$e_\theta(x_i) = c_e\, \mathcal{N}\left(d_{ij}; 0, \sigma_d\right) \mathcal{N}\left(\phi_{ij}; 0, \sigma_\phi\right) \tag{5.16}$$

where the perpendicular distance between x_i and m_j and angular difference between edge point gradient $\mathbf{g}_i$ and model contour perpendicular $\mathbf{v}_j$ are defined as in Chapter 4

$$d_{ij} = \| x_i - m_j \| \tag{5.17}$$

$$\phi_{ij} = \cos^{-1}\left(\frac{\mathbf{g}_i^{\mathrm{T}}\, \mathbf{v}_j}{\|\mathbf{g}_i\|\, \|\mathbf{v}_j\|}\right). \tag{5.18}$$

The range of d_{ij} is ≥ 0, and the angle ϕ_{ij} is in $[0, 1]$.

Pixels not containing an edge point still give an edge detection response from a nearby projected model contour. Suppose we know the projected model point generating each of

these non-edge responses. Then we define the probability of an edge detection response x_i that does not contain an edge point as

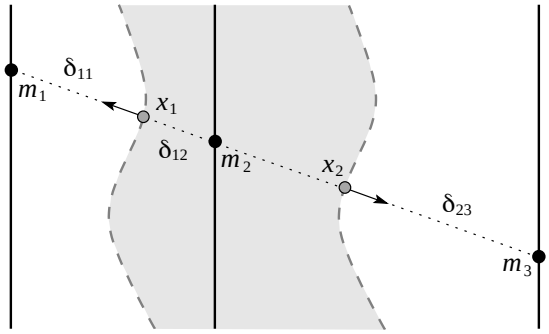
$$e'_\theta(x_i) = 1 - \int_{\mathbf{x}'_i} e_\theta(x)\, dx\,, \tag{5.19}$$

where $\mathbf{x}'_i$ is the space of all edge detection responses at the same pixel location as x_i, but that contain an edge point. That is, we define $e'_\theta(x_i)$ as the complement of the probability a model point generates any detection response containing an edge point at the i^{th} pixel.

Unfortunately, during model inference with actual detected edge points in an image, we do not know the correspondence between hypothesized model points and the edge detection responses x_i. We could search for the most likely correspondence linking edge detection responses and model points, but there are exponentially many of them. Therefore, we build uncertainty into the point correspondences by redefining the edge point generator density over several candidate model points for each edge point and develop an efficient approximation of its most likely correspondence.

We model an edge point with no correspondence information as generated by one of several candidate model points, and assume that each model point generates at most one edge point. If we detect an edge point at the i^{th} pixel of an image, it is modeled as being generated by one of K_i projected model contour points m_k that are nearby. We simplify computing nearby point correspondences by linking points on the hypothesized model contour to their closest image edge point in the direction of the edge gradient. Creating this linkage based on the detected edge gradient instead of the model contour perpendicular has some the practical advantages, including being able to quickly find the candidate model points. This is accomplished as follows.

For each image edge point, we compute the distance along the edge gradient to points on the projected model contours. Under the assumption that a model point generates at most one edge point, we link a model point to its closest edge point using the computed distances. Each edge point will then have a disjoint set of model points it is linked with. Figure 5.4 illustrates a simple example of this process. The model point set can be empty, however, due to no points along the edge gradient or the distance being greater than a



(a)

	m_1	m_2	m_3
x_1	δ_{11}	δ_{12}	δ_{13}
x_2	δ_{21}	δ_{22}	δ_{23}

(b)

	m_1	m_2	m_3
x_1	1	1	0
x_2	0	0	1

(c)

x_1	x_2
m_1	m_3
m_2	

(d)

Figure 5.4: Example point correspondence resolution linking three projected model contours (solid) with two edges (dashed) of an image object (shaded). For each edge point, a set of nearby model points in the gradient direction is found and used in the edge density function (5.20). All points in (a) are co-linear and parallel to the image gradients at x_1 and x_2. Distances between edge points and model points along the gradient are computed in (b); the shaded distances, $\delta_{12} < \delta_{11} < \delta_{23}$, are the smallest and labeled in (a). Model to edge point linkages are then made in (c) based on the closest edge point. Final linking of nearby model point sets to each edge point is summarized in (d).

threshold. In this case, the edge point is not linked to any model points and is considered noise.

Given a set of K_i linked model points, we redefine the density for an edge point x_i in our generative image model. Since we do not know which of the model points actually generated the edge point, we average across their Gaussian response of (5.16) with equal weights. The edge density function then becomes

$$\widetilde{e}_\theta(x_i) = \frac{1}{K_i} \sum_{k=1}^{K_i} \mathcal{N}\left(\widetilde{d}_{ik}; 0, \sigma_d\right) \mathcal{N}\left(\phi_{ik}; 0, \sigma_\phi\right) , \tag{5.20}$$

where the perpendicular distance $\widetilde{d}_{ik}$ from a model point is also redefined as

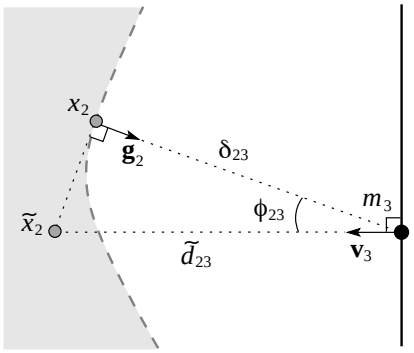


Figure 5.5: Distance and angle representation of $\widetilde{d}_{23}$ and ϕ_{23} for the edge point x_2 and model point m_3 in Figure 5.4. The point $\widetilde{x}_2$ is the generative approximation of x_2 that is perpendicular to both the projected model contour and gradient $\mathbf{g}_2$.

$$\widetilde{d}_{ik} \quad = \quad \| \widetilde{x}_i - m_k \| . \tag{5.21}$$

The point $\widetilde{x}_i$ is the generative approximation of x_i that is perpendicular to both the model contour at m_k and the gradient $\mathbf{g}_i$. If the edge point was found to be noise, however, due to no nearby model points, a constant minimum likelihood value, e_{noise}, is used instead. Figure 5.5 shows the details of this calculation for one of the example edge points in Figure 5.4.

In addition to redefining the edge point density using point correspondence estimation, we approximate the probability of not detecting an edge point at a particular pixel (5.19) with constants. We use a pair of probability constants for detection responses that are missing an edge point or are image background with no edge point expected. Pixels not containing an edge point, but that have the same image location as a projected model point, contribute a constant factor, e_{miss}, to the likelihood. However, only the $K_i - 1$ furthest model points from the i^{th} non-edge pixel contribute this constant; one of the model contours is assumed to have generated an actual edge point. Model points not linked to any edge point that have a detection response of no edge also contribute this constant. For all other pixel locations with no detected edge points, we factor in a constant background probability, e_{bg}.

We combine the approximations of the edge and no-edge density functions to redefine the likelihood for the edge point generator. Since detections not containing an edge point have constant probability, it is unnecessary to know which of the model points are missing an edge point. We only need to know how many there are, which we can easily compute by adding the number of model points not linked to an edge point with $\sum_{i=1}^{N}(K_i - 1)^{\mathcal{E}_i}$. This enables us to approximate the generator likelihood (5.15) with

$$\prod_{i=1}^{N} f_\theta(x_i) \approx \left\{ \prod_{i=1}^{N} \widetilde{e}_\theta(x_i)^{\mathcal{E}_i} \right\} \; e_{\text{bg}}^{N_{\text{bg}}} \; e_{\text{miss}}^{N_{\text{miss}}} \, . \tag{5.22}$$

where N_{bg} and N_{miss} are the number of background and missing detection responses in the image, and $N_{\text{bg}} + N_{\text{miss}} = \sum_{i=1}^{N} 1 - \mathcal{E}_i$. This is a reformulation of the edge generator likelihood (4.5), where we introduced binary weights in a summation over edge detection likelihoods that select the type of detection at each pixel. In the generator likelihood (5.22), we instead use a product over pixel detection likelihoods and summation over binary exponents to select the detection type. We can then estimate, for example, the number of background pixels or the number of missed edge detections, and directly compute the probability of those pixels.

Our approach has some similarities to standard edge matching (Borgefors, 1988; Huttenlocher et al., 1993), but we explain the edge points as the result of a generative statistical process that accounts for both distance and gradient direction. Using the Hausdorff distance for edges in our approach, for example, would preclude our ability to link edge points to projected model contours for likelihood computation, since no correspondences would be computed.

While the assumption that a model point can be assigned to at most one image edge point may seem arbitrary, we have experimented with other assignment alternatives and found it to give the best results. We have also experimented with different weightings in the average computed over model points in (5.20) and found uniform weights to work the best in the most cases.

Implementation detail: Much of the model and edge point linkage is easily precomputed at program initialization for the input images. After detecting the edge points

in each image, we create a correspondence grid of potential model point distances and gradient angles with the same dimensions and indexing as the image. The kth index in the grid stores the computed d_{ik} and discretized set of ϕ_{ik} for each of the edge points x_i whose gradient traces through the kth point in the grid. A discretized set of ϕ_{ik} is computed because the orientation of the model contour is not yet known. During learning and recognition, when the model contours are computed and projected, we look-up the distance and gradient angle for each of the model points in the precomputed table.

Surface point generator

Surface points are the projected points of viewable surfaces in our object model and represent detected image foreground. We detect foreground pixels by applying k-means clustering on pixel intensities. Setting $k = 2$ works well as our training images were selected with the objects on a uniform background to minimize clutter and emphasize structure learning. Figure 5.3 shows an example foreground detection for an image.

Similar to the case of edge points, the surface detector gives a response at each pixel location. We also have density functions for surface and non-surface points. Thus, we define the surface generator likelihood as

$$\prod_{i=1}^{N} f_{\theta g}(x_{gi}) = \prod_{i=1}^{N} s_\theta(x_i)^{\mathcal{S}_i} \cdot s'_\theta(x_i)^{(1-\mathcal{S}_i)}. \tag{5.23}$$

The per pixel indicator $\mathcal{S}_i$ is 1 if the pixel contains a detected surface point in the foreground, otherwise it is 0 and considered part of the background.

We define the density functions in terms of constant likelihoods for surface and non-surface points. The decision for what type of constant to use is based on comparing the surface point detection response at a pixel in the observed image and the corresponding projected object model surface point in the same pixel location of a hypothesized model image.

We define the density function for detected surface points with two constants for foreground and noise. If the pixel contains a detected surface point and shares a location with a projected model surface point, then we say it is part of the foreground and contributes

s_{fg}. If the detected surface point has no projected model surface point over it, we label it as noise and factor in s_{noise}.

We define the density function for detected non-surface points also with two constants, but for background and missing points. If the pixel does not contain detected surface point and has no projected model surface points in the hypothesized image, then we say it is part of the background and contributes s_{bg}. If the pixel again does not contain a detected surface point, but has a projected model surface point over it in the hypothesized image, we label it as missing a surface point in the observed image with factor in s_{miss}. Thus, the surface point generator likelihood expands to

$$\prod_{i=1}^{N} f_\theta(x_i) = s_{\text{fg}}^{N_{\text{fg}}} \; s_{\text{bg}}^{N_{\text{bg}}} \; s_{\text{noise}}^{N_{\text{noise}}} \; s_{\text{miss}}^{N_{\text{miss}}} \, . \tag{5.24}$$

where $N_{\text{fg}} + N_{\text{bg}} + N_{\text{noise}} + N_{\text{miss}} = N$.

5.3 Learning

To learn a category model, we sample the posterior, $p\left(\mathbf{\Theta^{(n)}} \mid \mathbf{X}\right) \propto p\left(\mathbf{X}, \mathbf{\Theta^{(n)}}\right)$, to find good parameters shared by images of multiple object examples from the category. Given enough iterations, a good sampler converges to the target distribution and an optimal value would be readily discovered in the process. However, our posterior distribution is highly convoluted with many sharp, narrow ridges for close fits to the edge points and foreground. In our domain, as in many similar problems, standard sampling techniques tend to get trapped in these local extrema for long periods of time. Our strategy for inference is to combine a mixture of sampling techniques with different strengths in exploring the posterior distribution while still maintaining convergence conditions.

Our sampling space is over all category and instance parameters for a set of input images. We denote the space over an instance of the camera and object models with n parts as $\mathbf{C} \times \mathbf{S}^{(n)}$. Let $\mathbf{T}^{(n)}$ be the space over all topologies and $\mathbf{R_c^{(n)}} \times \mathbf{R_s^{(n)}}$ over all category statistics. The complete sampling space with m subcategories and D instances is then defined as

$$\Omega = \bigcup_{\mathbf{n} \in \mathbb{N}^m} \mathbf{C}^D \times \mathbf{S}^{(\mathbf{n})D} \times \mathbf{T}^{(\mathbf{n})} \times \mathbf{R}_{\mathbf{c}}^{(\mathbf{n})} \times \mathbf{R}_{\mathbf{s}}^{(\mathbf{n})} \times \mathbf{n}\,, \tag{5.25}$$

Our goal is to sample the posterior with $\mathbf{\Theta}^{(\mathbf{n})} \in \Omega$ such that we find the set of parameters that maximizes it.

We combine several sampler transition kernels to effectively explore the parameter space. Since the number of parameters in the sampling space is unknown, some of the transitions must change the model dimensions. To this end, we build a trans-dimensional kernel in the Metropolis-Hastings framework to explore the space of category topologies. For parameter changes within a topology, we follow the strategy of Chapter 4 and apply both standard Metropolis-Hastings sampling and stochastic dynamics. Although the latter reduces random walk behavior of the sampler, it introduces additional computation in gradient calculation, which we estimate with numerical differentiation. Moreover, the Hyperdynamics sampler we used in Chapter 4 requires many estimations of the gradient to generate a sample from the biased energy function. Compounding the complexity of numerical gradient estimation is the fact that our parameter space is much larger than the model in Chapter 4, i.e. the object model here has an unknown number of blocks. For these reasons we pursue an alternative stochastic dynamics algorithm that does not require as much numerical differentiation per sample. However, we continue following the general strategy of mixing Metropolis-Hastings with stochastic dynamics. Our hybrid kernel then cycles between a mixture of these transitions while maintaining the posterior as invariant (Tierney, 1994).

5.3.1 Sampling within topology

To sample instance and category parameters within a particular object topology, we follow a hybrid transition kernel comprising the Metropolis-Hastings (MH) and stochastic dynamics algorithms. This enables efficient exploration of the complicated, multi-modal posterior distribution. In our experience, MH sampling enables large jumps between many modes of the posterior; stochastic dynamics excels at following tight ridges in highly correlated regions of parameter space that might otherwise cause high rejection

rates under MH. We describe our approach for each algorithm in the following.

Metropolis-Hastings

The Metropolis-Hastings algorithm is an MCMC sampling technique to generate unbiased and representative samples from a target distribution (Metropolis et al., 1953; Hastings, 1970; Neal, 1993; Forsyth et al., 2001; Bishop, 2006). The central concept of the algorithm is to propose samples from a distribution $q(\tilde{\boldsymbol{\theta}} \mid \boldsymbol{\theta})$, which can be easily sampled, and accept or reject the samples with probability

$$\alpha\left(\tilde{\boldsymbol{\theta}}^{(\mathbf{n})}\right) = \min\left\{1, \frac{p(\tilde{\boldsymbol{\theta}}^{(\mathbf{n})} \mid \mathbf{X})\, q(\boldsymbol{\theta}^{(\mathbf{n})} \mid \tilde{\boldsymbol{\theta}}^{(\mathbf{n})})}{p(\boldsymbol{\theta}^{(\mathbf{n})} \mid \mathbf{X})\, q(\tilde{\boldsymbol{\theta}}^{(\mathbf{n})} \mid \boldsymbol{\theta}^{(\mathbf{n})})}\right\}, \tag{5.26}$$

where we have sampled an independent subset of model parameters in $\boldsymbol{\Theta}^{(\mathbf{n})}$, such as a set of instance parameters.

We constructed several such proposal distributions, or diffusion moves, that modify the instance and category parameters and diffuse across the parameter space. We found a multivariate Gaussian with covariance values on the diagonal to be a good proposal distribution for the instance parameters. Proposals for block size changes are done in one of two ways: scaling or shifting attached blocks. We found that both are useful for good exploration of the object structure parameter space. Category parameters were sampled by making proposals from the Gamma priors.

Stochastic dynamics

The MH diffusion moves exhibit a random walk behavior and can take extended periods of time with many rejections to converge and properly mix well in regions of high correlation in the target distribution. As an alternative, we occasionally follow a hybrid Markov chain based on stochastic dynamics (Neal, 1993; Bishop, 2006). However, rather than rely upon the combination of Langevin and Hyperdynamics from Chapter 4, we use the Verlet integration algorithm (Verlet, 1967, 1968), commonly referred to as the leap-frog algorithm. The primary reason for this switch is to balance the trade-off between good mixing and computational complexity.

As we have shown, cycling Langevin and Hyperdynamics samplers provide an excellent means to transition between regions of high probability in our posterior; the Langevin dynamics excel at rapidly moving to areas of high probability, and Hyperdynamics enables transitions to saddle points between these areas. Unfortunately, the computational burden of Hyperdynamics is too high for practically fitting category models to a set of images containing objects in an category. We observe that solely following Langevin dynamics focuses the sampler too much on regions of high probability (Figure 4.3) for long sampling runs. Thus we use another dynamics algorithm, Verlet (1967, 1968), that balances the traits of Langevin and Hyperdynamics, as we later illustrate in Figures 5.6 and 5.7.

We use ideas from molecular dynamics and simulate a physical system by representing our model parameters as a position in phase space with an introduced and hypothetical momentum, $\mathbf{r}$, under an energy function involving our posterior joint density. The dynamics in the system generate representative samples while reducing the random walk effect and drive the integration over phase space.

We define a potential energy over position from the joint density function (5.5), and a kinetic energy for the introduced momentum as

$$E^{(\mathbf{n})}(\boldsymbol{\theta}) = -\log p^{(\mathbf{n})}(\mathbf{X},\, \boldsymbol{\theta}) \tag{5.27}$$

$$K^{(\mathbf{n})}(\mathbf{r}) = \frac{1}{2}\sum_{i=1}^{N^{(\mathbf{n})}} r_i^2\,. \tag{5.28}$$

The total energy in phase space is then given by the Hamiltonian

$$H^{(\mathbf{n})}(\boldsymbol{\theta},\, \mathbf{r}) = E^{(\mathbf{n})}(\boldsymbol{\theta}) + K^{(\mathbf{n})}(\mathbf{r})\,, \tag{5.29}$$

which we use in the canonical distribution over position and momentum in phase space, $p^{(\mathbf{n})}(\boldsymbol{\theta},\, \mathbf{r}) = Z_H^{-1} \exp\left(-H\left(\boldsymbol{\theta},\, \mathbf{r}\right)\right)$. Finally, integrating the canonical distribution is accomplished by following the Hamiltonian dynamics over time τ,

$$\frac{d\theta_i}{d\tau} = r_i\,, \quad \frac{dr_i}{d\tau} = -\frac{\partial E}{\partial \theta_i}. \tag{5.30}$$

Integrating the dynamics exactly conserves total energy and volume in phase space, leaving the canonical distribution invariant (Neal, 1993). To actually follow the dynamics and generate samples from phase space, we discretize with the Verlet, or leap frog, algorithm, where we alternate updating the momenta and position after every half time step. For a small step-size, ϵ, we update according to

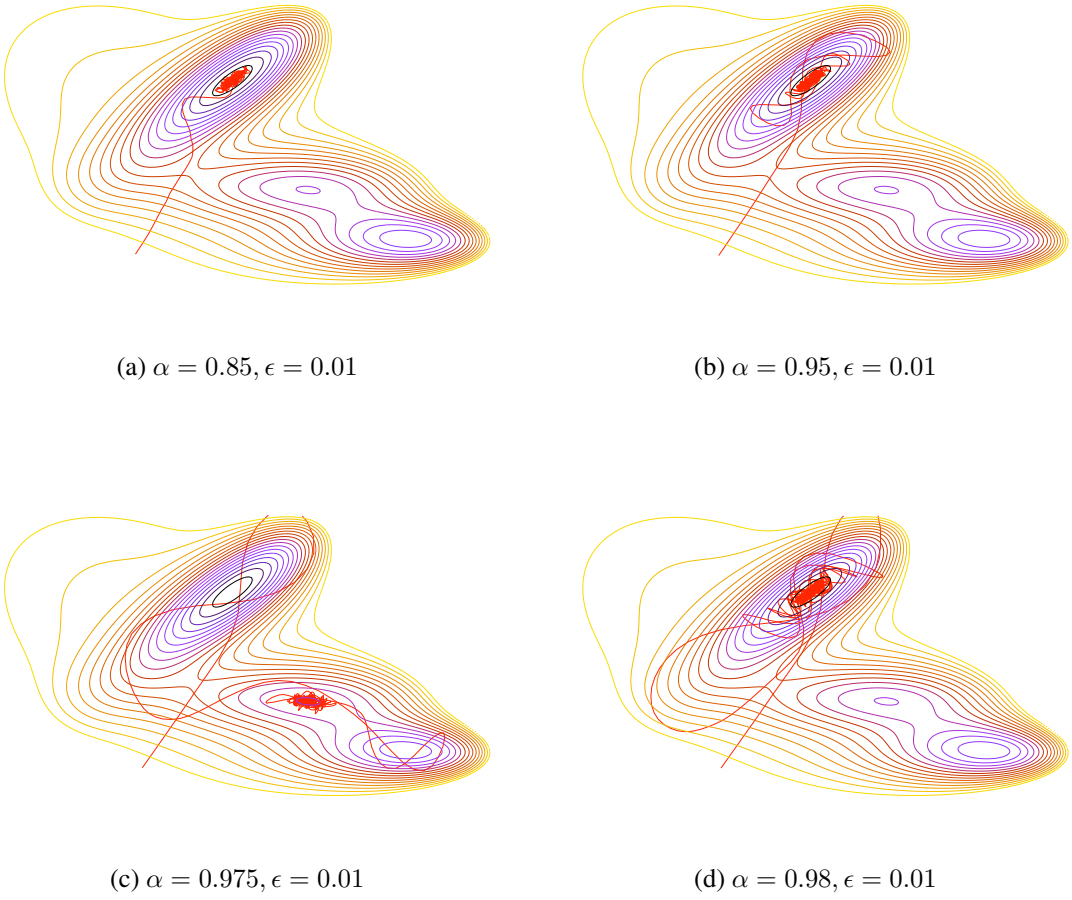
$$\tilde{r}_i\,(\tau + \epsilon/2) = r_i(\tau) - \frac{\epsilon}{2}\frac{\partial E\,(\theta_i(\tau))}{\partial \theta_i} \tag{5.31}$$

$$\tilde{\theta}_i(\tau + \epsilon) = \theta_i(\tau) + \epsilon\,\tilde{r}_i(\tau + \epsilon/2) \tag{5.32}$$

We further introduce a stochastic transitions after each step to ergodically sample from the from the canonical distribution in states of slightly different total energy. To accomplish this we use the stochastic transitions

$$\tilde{r}_i = \alpha\, r_i + (1 - \alpha^2)^{1/2}\,\eta_i\,, \tag{5.33}$$

where η_i is drawn from a standard normal. We choose α close to one to maintain nearly constant energy, enabling a transition back to Metropolis-Hastings sampling. In Figures 5.6 and 5.7 we show the effects of varying the values of α and ϵ in this algorithm on Müller's potential (Müller, 1980), which was defined in (4.13) of Section 4.3.1. We observe that choosing values of α close to one and relatively large values of ϵ reduces the amount of resistance in the dynamics and enables good mixing. Although there is some bias introduced in this hybrid approach, we are primarily interested in MAP estimation and have found it to work well in practice. The necessary derivative calculations of (5.5) are approximated using numerical differentiation.



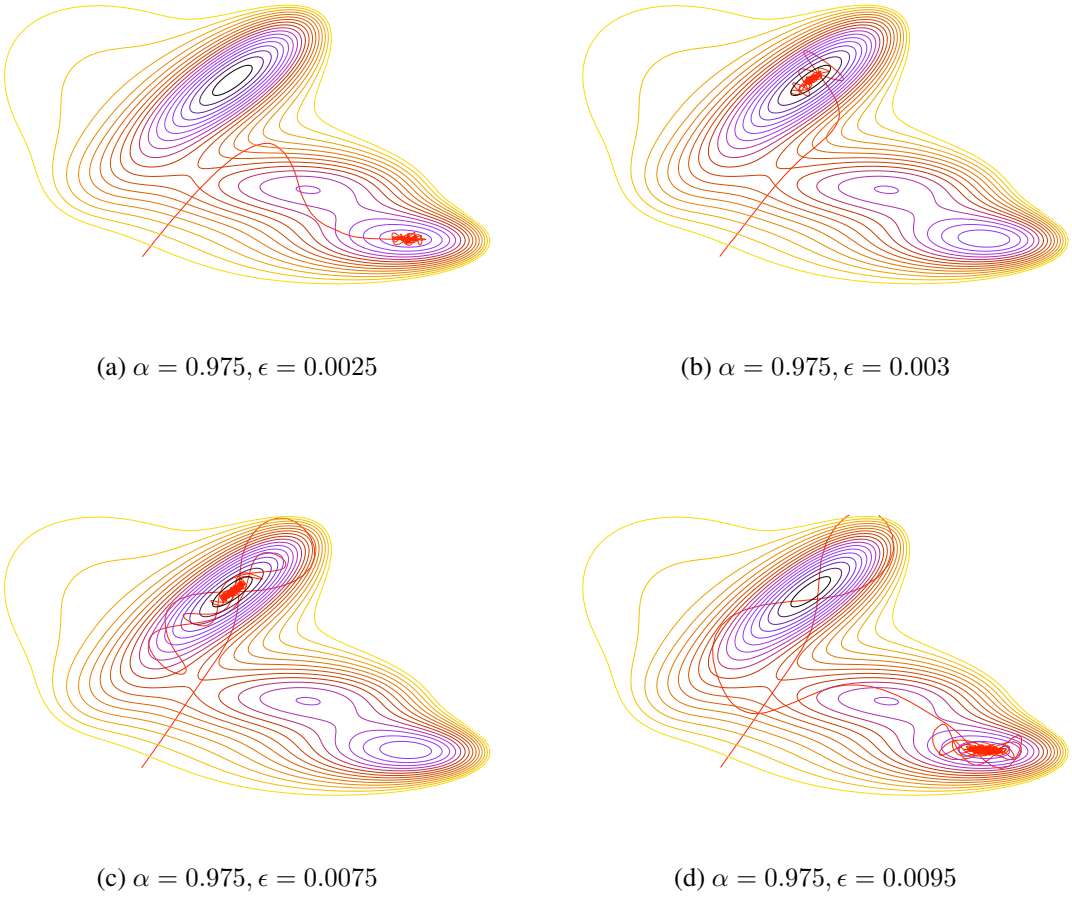
(a) $\alpha = 0.85, \epsilon = 0.01$

(b) $\alpha = 0.95, \epsilon = 0.01$

(c) $\alpha = 0.975, \epsilon = 0.01$

(d) $\alpha = 0.98, \epsilon = 0.01$

Figure 5.6: Effects of varying α during Verlet dynamics sampling on Müller potential (4.13) for 1000 iterations. As α approaches 1, the amount of resistance, or drag, in the sampler reduces and the transition rate is between states is increased.



(a) $\alpha = 0.975, \epsilon = 0.0025$

(b) $\alpha = 0.975, \epsilon = 0.003$

(c) $\alpha = 0.975, \epsilon = 0.0075$

(d) $\alpha = 0.975, \epsilon = 0.0095$

Figure 5.7: Effects of varying ϵ during Verlet dynamics sampling on Müller potential (4.13) for 1000 iterations. For small ϵ (a), the sampling step size is decreased and momentum rapidly decreases, leaving the sampler in a single state. As ϵ increases, more of the sampling space is explored before momentum is lost.

5.3.2 Sampling topologies

For changes to the object topology, we add or remove blocks by following the trans-dimensional sampling technique outlined by Green (1995) and refer to these changes as jump moves. For example, in the case of a block birth in the model, we modify the MH acceptance probability to

$$\alpha\left(\tilde{\boldsymbol{\theta}}^{(\mathbf{n}+1)}\right) = \min\left\{1,\ \frac{p(\tilde{\boldsymbol{\theta}}^{(\mathbf{n}+1)} \mid \mathbf{X})}{p(\boldsymbol{\theta}^{(\mathbf{n})} \mid \mathbf{X})\, q(\tilde{\mathbf{b}},\, \tilde{\mathbf{t}})} \frac{r_d}{r_b} \left| \frac{\partial(\tilde{\boldsymbol{\theta}}^{(\mathbf{n}+1)})}{\partial(\boldsymbol{\theta}^{(\mathbf{n})},\, \tilde{\mathbf{b}},\, \tilde{\mathbf{t}})} \right| \right\} \tag{5.34}$$

The proposal distribution generates a new block and attachment edge in the topology that are directly used in the proposed object model. Hence, the change of variable factor in the Jacobian reduces to 1. The probability of selecting a birth move versus a death move is given by the ratio of r_d/r_b, which we have also defined to be 1. The complimentary block death move is similar with the inverse ratio of posterior and proposal distributions.

In order to obtain good mixing of the jump moves in our trans-dimensional sampler, we additionally define split and merge moves. These are essential moves in our case because the sampler often generates blocks with strong partial fits and proposing splitting them is often accepted. The acceptance probability for merge/split is the same as birth/death; we use parameters from a proposed block to deterministically split a block already in the model, with an analogous move for the merge.

5.4 Results

We evaluated our model and its inference with image sets of furniture categories, including tables, chairs, sofas, footstools, and desks. We have 30 images in each category containing a single arbitrary view of the object instance. Although our image model represents detected feature noise, we selected images that have the furniture object prominently in the foreground. This enables focusing on evaluating how well we learn 3-D structure models of objects.

Inference of the object and camera instances was done on detected edge and surface

points in the images. We applied a Canny-based detector for the edges in each image, using the same parametrization each time. Thus, the images contain some edge points considered noise or that are missing from obvious contours. To extract the foreground, we applied a dynamic-threshold discovered in each image with a k-means algorithm. Since the furniture objects in the images primarily occupy the image foreground, the detection is quite effective. Figure 5.3 shows examples of detected edges and foreground.

We learned the object structure for each category over a 15-image subset of our data for training purposes. We initialized each run of the sampler with a random draw of the category and instance parameters. This is accomplished by first sampling the prior for the object position, rotation and camera view; initially there are no structural elements in the model. We then sample the likelihoods for the instance parameters. The trans-dimensional moves in the sampler iteratively propose adding and removing object constructs to the model. Figure 5.9c illustrates which sampler moves are accepted and when by plotting the potential energy for each accepted move during 2K sampler iterations of the chair category parameters. Similarly, Figures 5.9a-b plot the potential energy for 2 of the 15 chair instances while they are simultaneously fit with the category parameters in Figure 5.9c. The mixture of moves in the sampler was 1-to-1 for jump and diffusion and very infrequently performing a stochastic dynamics chain. Figures 5.11, 5.12, 5.13 show examples of learned furniture categories and their instances to images after 100K iterations. We observe that the topology of the object structure is quickly established after roughly 10K iterations, this can be seen in Figure 5.8, which shows the simultaneous inference of two table instances through roughly 10K iterations. In addition to a good topology, we also learn the category structure statistics, which we have rendered using random category parameter samples in Figure 5.10. Finally, since the variation of structure within each of our object categories is quite small, we found that using a single cluster generally has the same result as learning multiple clusters.

We tested the recognition ability of the learned models on a held out 15-image subset of our data for each category. For each image, we drew a random sample from the category statistics and a topology and began the diffusion sampling process to fit it. The best overall fit according to the unnormalized joint density (5.5) is declared the predicted

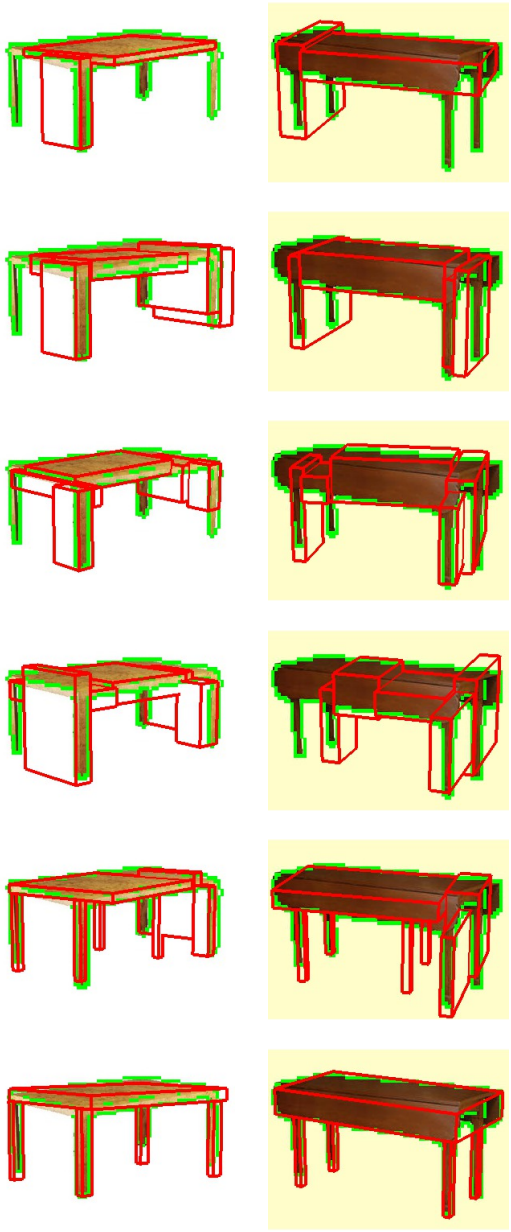
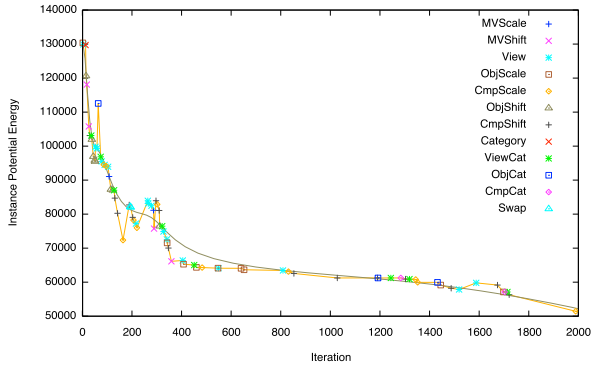
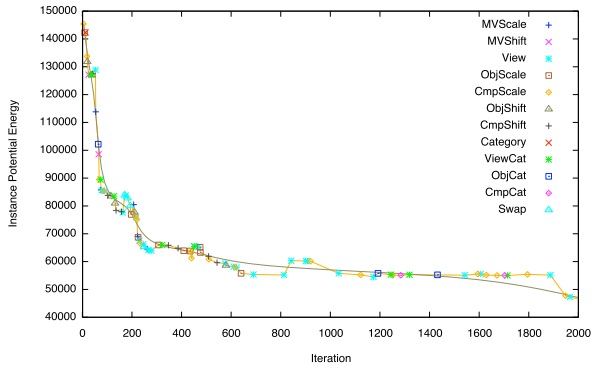


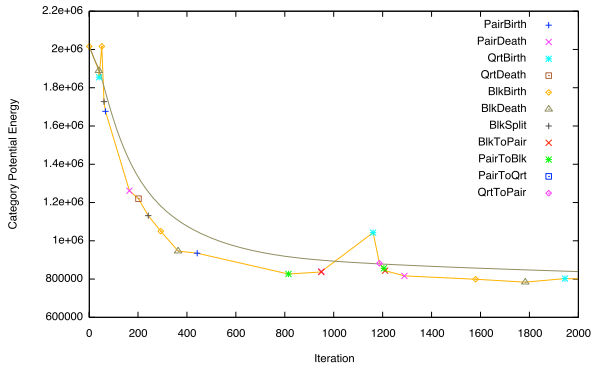
Figure 5.8: From left to right, successive random samples from 2 of 15 table instances, each after 2K iterations of model inference. The category topology and statistics are learned simultaneously from the set of images; the form of the structure is shared across instances.



(a) Instance 2



(b) Instance 7



(c) Category

Figure 5.9: Potential energy of accepted jump and diffusion moves during inference of the chair category and instance parameters after 2K iterations. (c) potential energy for each accepted category parameter move. (a)-(b) potential energy for 2 of the 15 chair instances while they are simultaneously fit with the category parameters in (c).

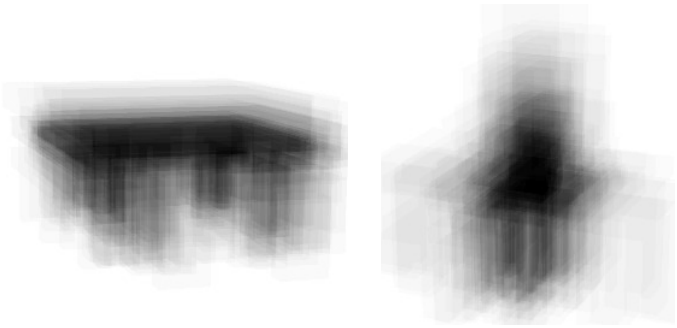


Figure 5.10: Generated samples of tables and chairs from the learned structure topology and statistical category parameters. The rotation angle for each category was fixed and 100 samples of instances from the table and chair categories were drawn from the topology and statistics.

	Actual				
Predicted	Table	Chair	Footstool	Sofa	Desk
Table	10	5	4	0	2
Chair	5	9	10	5	3
Footstool	0	0	1	3	1
Sofa	0	1	0	7	3
Desk	0	0	0	0	6

Table 5.1: Confusion matrix for object category recognition

category. The confusion matrix shown in Table 5.1 shows mixed results. Overall, recognition is substantively better than change (20%), but we expect that much better results are possible with our approach, and consider the current ones preliminary. We observe from these results that the learned chair topology shares much of its structure with the other categories and causes the most confusion. Estimating a normalization constant for the joint density under each category should help to improve these results by providing a more accurate comparison of category probability. We have yet to extensively experiment with larger training data sets, clustering category structure, and long run times to get better structure fits in the difficult training examples, each of which could help resolve this confusion.

Figure 5.11: Learning the structure of a table object. Model fitting is done jointly across fifteen training images using sets of contiguous blocks. The category topology and statistics are shared across all images, whereas the instance parameters (camera, block position and size) were fit across the exemplars. The location of the edge points in the image (shown in green) is only softly fit to the model edges (shown in red) to account for the deformation from a block (note the legs of some of the tables). While the quality of individual fits naturally varies across examples (these are relatively good results), in both cases the system learned, from the same starting point, a recognizable topology for the category represented by the collections of instances.

Figure 5.12: Learning the structure of a chair object. Model fitting is done jointly for the fifteen images in the training set using contiguous blocks to represent the structure. The fits for the training examples is shown by the blocks drawn in red. Detected edge points are shown in green.

Figure 5.13: Learning the structure of footstools, sofas, and desks. Sets of contiguous blocks were fit across each image data set. Model fitting is done jointly for the fifteen images of each set. The fits for the training examples is shown by the blocks drawn in red. Detected edge points are shown in green.

5.5 Discussion

The main goal of this work was to develop an approach for learning strong 3-D models, with unknown topologies, from single 2-D images. In particular, we are working with models that represent objects as 3-D assemblages of sub-structures that we assume to be effective at representing a range of objects for a variety of vision task. A key technical challenge for learning such models from single 2-D views was streamlining the inference so that model hypotheses can be explored relatively quickly. A second key challenge addressed in this work was to arrange an image model that statistically explains every image pixel to effectively mitigate against biases for more or less complex models, and explaining more or less of the image. Dealing with these two challenges allowed viable topologies to emerge that are consistent with multiple images of objects from the same category.

We have developed the approach in a relatively simple domain, but the methods can be extended to more general configurations (relaxing the right angle assumption), and a larger palate of more deformable parts. Further, once we expand the collection of categories to more complex objects, we can explore more deeply how our clustering approach interacts with learning and whether that can improve category recognition.

CHAPTER 6

Conclusion

6.1 Learning models of object structure

In this dissertation we presented an approach for representing object structure with 3-D geometric models and methods for inferring them from image data. We represented object structure as a collection of connected 3-D geometric primitives, such as blocks, cylinders, and ellipsoids, corresponding to object parts. We demonstrated our approach on biological structure and man-made objects. In both cases, *Alternaria* and furniture objects, we showed how a small set of production rules for piecing the parts together can build category specific topologies of structure.

In addition to modeling structure, we separated a representation of the imaging system away from the objects and their pose. This enabled a better understanding of the variation of 3-D object structure viewed in image data. In the context of *Alternaria*, we developed a model of the microscope capturing 3-D stacks of images. For furniture objects, we created a model of a standard camera taking single-view images constrained to reduce possible ambiguities. This abstraction allowed us to model object structure independent of how it was imaged.

We defined a statistical model over the parameterized representations of both the object structure and imaging system. Conditioned on an instance of these, our likelihood for image data statistically generates observations. We then reverse this generative formulation in terms of Bayesian statistical inference with a posterior distribution. To accomplish the inference, we presented effective sampling algorithms that explore the varying dimensions of topology space and parameter ranges within a particular topology. We showed that both structure and imaging models can be inferred simultaneously given image data. We further showed in the case of *Alternaria* that data-driven sampling is an effective means to improve the convergence rate and inference of a posterior maximum. For the

case furniture we demonstrated that we could infer category level information such as topologies and shape statistics shared by objects across groups of images.

Understanding objects in 3-D enables more than just recognition in images. The models we learned can be used to extract quantitative information about structure. At the most basic level, we learned models for particular instances, which is especially useful in biological data, such as microscopic images of *Alternaria*; we can provide numerical information about structure that was previously described by humans qualitatively. Similarly, quantitative information is available for learned instances of furniture from single view images. In this case, however, we also learned quantitative descriptions of category-level structure; we extracted information about the topological definition for a category of objects and the statistics over their shape.

When most people look at an object they not only see its surface, texture and outline, but often understand its constituent parts. For example, we easily identify where a door is located on a model of car not seen before. The structure representation yielding these abilities goes even further, to the extent that we can predict where in 3-D unseen parts are located and what they might look like. This type of 3-D part representation would be extremely useful in situations where we want to build machines that can understand and interact with objects as they exist in the world under unseen views. In this thesis we have presented an approach that moves towards automatically learning such representations and being able to use them to understand object structure in images.

6.2 Contributions

We have made a number of contributions in this work towards understanding images of object structure.

- We propose learning both 3-D object structure and imaging models simultaneously from data. Inference of the structure is assisted by an estimate of the camera, and an improved camera estimate is possible by understanding the structure in the image.
- We present a parameterized point spread function model for a brightfield transmitted light microscope.

- We present a geometric primitive representation for biological structure composed of independent or recursively connected filament-like parts, such as those in *Alternaria*. The model can be extended to other specimens based on a set of rules for its growth.

- We describe a specific grammar for Alternaria and discuss L-systems as a general approach for biological structure representations.

- We present a generative model for the parameterized geometric structure of *Alternaria* comprising ellipsoids and cylinders.

- We present an image likelihood for 3-D stacks of microscopic image data based on geometric primitives in a biological structure model.

- We present a sampling algorithm that infers parameterized biological structure and microscope imaging models from data simultaneously.

- We present a mixture of reversible-jump and diffusion-based MCMC samplers for learning *Alternaria* structure models. The trans-dimensional sampler searches over *Alternaria* topologies by implementing rules in its grammar for growth. The diffusion moves search within the topology of an instance by sampling likely values from parameter ranges.

- We present a data-driven augmentation to the MCMC sampler that improves successful generation of substructure element proposals and increases sampler convergence rate.

- We present a surface point detection algorithm from 3-D stacks of microscope images. We describe several directions for developing a more complete surface model and using it to improve model fitting.

- We present a comparison of using different point spread functions and their effects on fitting structure in the *Alternaria* data.

- We demonstrate that the sampling algorithm successfully fits *Alternaria* structure and microscope point spread functions to 3-D stacks of images.
- We present a constrained camera model that can produce arbitrary single view images of 3-D object models and minimize parameter ambiguity.
- We present a parameterized, connected block structure model for tables and show how it can be fit to single view images.
- We present a generalized structure model that can describe man-made furniture objects comprising a topology of connected blocks.
- We present a generative category model for furniture objects that includes stochastic topology creation and structure shape statistics.
- We present an image likelihood for detected features in images of objects captured by a standard camera. This includes edge points and surface points, but can be extended to color, texture, and other interest points.
- We present an generative edge point distance measure for our likelihood model and a method to compute it quickly during each iteration of the inference process.
- We present a sampling strategy that mixes Metropolis-Hastings and stochastic dynamics. For cases of numerical differentiation on a small, fixed number of parameters, e.g. the table model, we mix Hyperdynamics, Langevin dynamics, and Metropolis-Hastings to quickly explore the highly correlated state space.
- We present a Metropolis-Hastings covariance scaled sampling extension to situations where the form of the posterior distributions is not completely known, and use it as a mechanism to make large jumps between modes of the posterior.
- We demonstrate how mixing the above samplers successfully fits 3-D structure and camera models to single view images of furniture objects with fixed topologies, such as a table.

- We present a sampling algorithm to learn topologies of general furniture objects using reversible-jump on the rules of the structure model. The algorithm also samples within topologies by using stochastic dynamics and Metropolis-Hastings.

- We present a stochastic dynamics sampler based on the Verlet algorithm as an alternative to the Hyperdynamics and Langevin mixture. The latter become prohibitively expensive to use with numerical differentiation as the number of parameters increases and the topology is unknown.

- We demonstrate how category models are successfully fit for types of furniture objects including tables, chairs, footstools, desks, and sofas.

- We demonstrate that the learned furniture category models can be used for recognition of objects in unseen images.

6.3 Future work

This work can be extended in many directions, with the most immediate being an increase in the number of learned furniture categories. Preliminary work suggests that creating cross-pieces between two blocks in our furniture model enables us to represent many more structures, for example bookshelves. In other words, by expanding the rule set for constructing topologies of blocks, we can represent more types of furniture objects. A challenge in this extension includes creating reversible-jump sampler moves that effectively transition with a reasonable acceptance rate between cross-piece blocks and other types of attachment.

Another direction we plan to explore is further investigating learning clusters of structure. There are currently two types of clustering possible with our model. We could collect images from different categories together in a single group and learn subcategories of object structure within that set. This would be an unsupervised approach to learning structure categories. Alternatively, for a set of images within the same category, we could focus on learning subcategory structure, such as chairs with armrests and those without. An idea for pursuing this line of clustering includes running the sampler for some time to

fit rough structure topologies, then allowing it to split categories into subcategories and fitting more detailed structure topologies within these clusters.

Both of our structure and image likelihood models for man-made objects could be improved in multiple ways. We could try fitting more sophisticated structure elements within our object model. For example, we could use shapes defined with splines or generalized cylinders to capture curvature. We could also relax the prior over block attachment angle to allow for topologies with more then just right angle attachments. The image likelihood model could be extended by representing other detectable image features, including object texture and color. We could also develop a generative approach for detectable corners in an image, such as those from a Harris corner detector.

For our stochastic dynamics sampling algorithms that rely on derivatives of the likelihood function, we currently use numerical differentiation. As the number of substructures or blocks are added to a model, this computation becomes extremely expensive. An alternative approach that would significantly speed up the sampler is an analytical derivative computation. The difficulties in this lie in the number of parameters relating to the geometric primitives and their transformations. For the case of Alternaria and 3-D stacks of images, we have formulated analytical derivatives with orthographic projection of ellipsoids and cylinders. An implementation of this formulation is still needed, however. For single view images from a standard camera, the problem is more difficult due to a perspective projection. Ideas for solving this problem have been previously addressed by Lowe (1991), indicating that an approximation is possible.

Other ideas for improving inference through sampling is to define more extensive data-driven proposals. This includes estimating the location of 3-D planes in images, block corners, or complete parameterizations of blocks. For stacks of 3-D microscopic images, we could also extend the surface reconstruction algorithm, as described in Appendix B, and fit our structure model to that instead of solely to image data. This could improve the fit of the model, and improve the surface reconstruction.

A parallel direction that we have already begun investigating with some success is fitting the 3-D scene orientation and camera viewing it to single view images. For images of indoor scenes where a majority of the edges are parallel to one of three primary axis,

we utilize our image likelihood model to estimate the orientation of the primary walls of the scene and the direction of the camera. A direct extension of this approach is to include our learned 3-D models of furniture objects as structure proposals in order to recognize and place them within the scene. By assuming that all objects in a scene are aligned with one of the three primary axis of the scene, we can infer the scene orientation and better estimate objects that are difficult to detect. We can also start to learn about relative sizes of objects in the scene.

Finally, we would like to take the approach for learning category parameters of structure models of man-made objects and apply that to biological structure, such as the *Alternaria* model. This would enable learning quantitative structure information that characterizes species or phenotypes of *Alternaria*. For example, we could quantify the average branching angles or spore count of one species versus another. This would further contribute towards the overall goal of building a high-throughput analysis system capable of automatic species classification.

APPENDIX A

Markov Chain Monte Carlo Sampling

A.1 Introduction

According to Sokal (1989), Monte Carlo methods are a bad approach and should only be used "when all alternative methods are worse." The primary justification for this is two-fold: (1) unlike analytical methods, numerical methods utilize algorithms to estimate their solution and have more potential for statistical error; and (2) compared with other numerical methods, Monte Carlo approaches are extremely inefficient. In fact, citing the central limit theorem, Sokal argues that the amount of error around any estimate achieved with Monte Carlo methods is of the order $1/\sqrt{n}$, where n is the number of samples used for estimation. If any other numerical methods are available with smaller error, they should be used instead. It is often the case, however, that many common real-world problems push the error of deterministic numerical methods past Monte Carlo error thresholds. Sokal uses the example of numerical integration with Simpson's rule. Under this common approach, when a function of d dimensions is broken into n intervals to approximate analytical integration, the error is $n^{-4/d}$. Thus, in high dimensions, e.g., $d > 8$, applying Simpson's rule for numerical integration has worse error than an estimate from Monte Carlo sampling. As in many real-world domains, problems in computer vision frequently have many degrees of freedom resulting in intractable analytical solutions. Thus we feel justified in turning to sampling methods.

Although the work presented in this thesis does not currently estimate expectations using Monte Carlo samples, it is desirable to express how much error is involved in such averages, and how long a Markov chain approach will take to converge to the stationary, or target, distribution. Having a quantitative way to estimate convergence would be useful in our application. Moreover, for projects that extend this work, where we might build a hierarchical Bayesian models, it could be useful to compute expected values of solu-

tions instead of maximum a posterior estimates. In this case, having an estimate of error involved would be informative.

For the remainder of this appendix, we describe the theoretical analysis of MCMC methods to provide reasoning for their usage in our application of learning structure models. We follow the presentation of Sokal (1989) and Neal (1993) to initially show under what conditions an MCMC sampling algorithm converges to a target distribution, which in our case is typically a posterior distribution. We further outline ways to estimate how long convergence could take and how to estimate the number of samples required to achieve a desired error rate. Finally, we summarize the Metropolis-Hastings algorithm, a specific type of MCMC sampler used throughout this dissertation and show how that algorithm satisfies the convergence requirements.

A.2 Markov chain Monte Carlo theory

Suppose we would like to generate random samples from a target probability density function π on a (discrete state) space $\mathcal{S}$. For large enough sample size N, we can produce expectation estimates using Monte Carlo integration

$$\mathbb{E}[f(X)] = \int_{\mathcal{S}} f(x)\,\pi(x)\,dx \tag{A.1}$$

$$\approx \sum_{n=1}^{N} f(x_n)\,\pi(x_n)\,. \tag{A.2}$$

Furthermore, if we can reliably sample from π, we can be sure to find states with maximal probability.

The method for generating samples described here produces a stochastic process, specifically a Markov chain, that converges to the target distribution π when started from an initial distribution α and an arbitrary point in $\mathcal{S}$. We further provide estimates for how long the convergence will take and the amount of error produced in estimates with the samples. Although our application is currently only to estimate extrema of a distribution, knowing the convergence rate would still be helpful.

We first review Markov chains as a stochastic process for generating samples over a state space. Each transition in a Markov chain is independent from all others by having its state depend only on the previous. In other words, for an initial distribution $\alpha_x = \mathbb{P}(X = x)$ on $\mathcal{S}$ and transition probability matrix P with elements $p_{xy} = \mathbb{P}(X_{t+1} = y \,|\, X_t = x)$, the probability of n successive states is

$$\mathbb{P}(x_1, \ldots, x_n) = \alpha_{x_1} \, p_{x_1 x_2} \, p_{x_2 x_3} \, \cdots \, p_{x_{n-1} x_n} \, . \tag{A.3}$$

The matrix P satisfies $p_{xy} > 0$ for all x, y and $\sum_y p_{xy} = 1$ for all x. It is also useful to define a n-step transition probability

$$p_{xy}^{(n)} = \mathbb{P}(X_{t+n} = y \,|\, X_t = x) \, . \tag{A.4}$$

It can be shown that the matrix P^n has elements $p_{xy}^{(n)}$.

We next define two useful attributes of a Markov chain P: irreducibility and stationarity. A chain is *irreducible* if there is a non-zero probability to get from a state x to any other state y with zero or more transitions, i.e. $p_{xy}^{(n)} > 0$ for all $x, y \in \mathcal{S}$ and some $n \geq 0$. A distribution, such as π, is *stationary* for a chain if

$$\sum_x \pi_x \, p_{xy} = \pi_y \, , \tag{A.5}$$

for all $y \in \mathcal{S}$. A sufficient, but not necessary, condition for guaranteeing a probability distribution π is stationary for a chain P is the detailed balance condition

$$\pi_x \, p_{xy} = \pi_y \, p_{yx} \, , \tag{A.6}$$

for every pair of states $x, y \in \mathcal{S}$. Most MCMC methods make use of this condition to obtain stationarity.

If an aperiodic Markov chain is irreducible and has a stationary distribution (not all chains must), then it can be proved that

$$\lim_{n \to \infty} p_{xy}^{(n)} = \pi_y \, . \tag{A.7}$$

A similar result holds if the chain has period $d > 1$. This theorem implies that the chain converges to the stationary distribution π as its length goes to infinity, regardless of the starting distribution α. An important result of the above theorem, and the strong law of large numbers, is that simulating the Markov chain P is a suitable means to generate samples from π and perform Monte Carlo average estimations, as in (A.2). Further, by applying the central limit theorem, it can be shown that the amount of error of any such statistical estimates will be proportional to $n^{-1/2}$.

A.3 Convergence rate and sample correlation

We now turn to defining bounds for how long the convergence will take and how much correlation exists between generated successive states. For a Markov Chain P with stationary distribution π, a function $f_t = f(X_t)$ on the state space $\mathcal{S}$ has stationary mean

$$\mu_f = \langle f_t \rangle = \sum_x \pi_x \, f(x) \, . \tag{A.8}$$

Both convergence time and correlation are defined in terms of the autocorrelation

$$C_{ff}(t) = \langle f_s f_{s+t} \rangle - \mu_f^2 \tag{A.9}$$

$$= \sum_{x,y} f(x) \left[\pi_x p_{xy}^{(t)} - \pi_x \pi_y \right] f(y) \, , \tag{A.10}$$

which is normalized and denoted $\rho_{ff}(t) = C_{ff}(t)/C_{ff}(0)$. It is assumed that the normalized autocorrelation decays exponentially with the length of the chain. So the exponential autocorrelation time is defined as

$$\tau_{exp} = \sup_f \left\{ \limsup_{t \to \infty} \frac{t}{-\log |\rho_{ff}(t)|} \right\} \, . \tag{A.11}$$

The time is the number of steps taken in the chain for convergence to an upper bound of the autocorrelation under the slowest converging mode (f). This can also be interpreted as the number of samples to throw away at the beginning of the Markov chain run before

equilibrium is reached, so called burn-in time.

For measuring the amount of error involved in an average estimation of f, we define an integrated autocorrelation time

$$\tau_{int,f} = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{ff}(t)\,, \tag{A.12}$$

and say that a sample mean $\bar{f} \approx \langle f \rangle$ has variance

$$\mathbb{V}(\bar{f}) = \frac{1}{n^2} \sum_{r,s=1}^{n} C_{ff}(r-s) \tag{A.13}$$

$$\approx \frac{1}{n} \left(2\,\tau_{int,f}\right) C_{ff}(0)\,. \tag{A.14}$$

The main idea behind this statement is that the variance of $\bar{f}$ is a factor of $2\,\tau_{int,f}$ larger than it would be if all the samples were completely independent. Another interesting point we can extract from this is that the number of independent samples during a run of the Markov chain of length n is approximately $n/2\,\tau_{int,f}$.

Estimating the values of τ_{exp} and $\tau_{int,f}$ for a Markov chain is not always a straightforward process. Sokal (1989) describes several empirical techniques that could be used, with most involving estimating the autocorrelation function $C_{ff}(t)$. In any case, an estimate should be made for the the number of samples to discard at the beginning of the run, τ, before equilibrium is reached. We can than use this value to approximate the order of the statistical errors as $(\tau/n)^{1/2}$. Further, since τ_{exp} and $\tau_{int,f}$ are typically of the same magnitude, we can use τ to estimate the number of samples needed for accurate averages. For example, if 1% error is desired, we should run the chain for about 10000τ iterations.

A.4 Metropolis-Hastings algorithm

Finally, we review how the Metropolis-Hastings (MH) algorithm satisfies irreducibility and stationarity and thus generates samples that can be used in Monte Carlo averaging. Irreducibility can be maintained in the chain by making sure there are no zero probability

transitions. As previously mentioned, a sufficient condition for stationarity is to show detailed balance (that the chain is reversible). The MH algorithm maintains this condition, as we show.

Let $p_{xy}^{(0)}$ be the proposal probability in the algorithm from state x to y. This proposed transition is either accepted or rejected with probability a_{xy}. Then the transition matrix in the Markov chain P has non-diagonal elements of accepted moves

$$p_{xy} = p_{xy}^{(0)}\, a_{xy}\,. \tag{A.15}$$

The diagonal elements of P are the probability of rejection $(1 - a_{xy})$, i.e., the probability of staying in the same state

$$p_{xx} = p_{xx}^{(0)} + \sum_{y \neq x} p_{xy}^{(0)}\,(1 - a_{xy})\,. \tag{A.16}$$

This can be thought of as the probability of proposing to stay in the same state or proposing a transition to any other state and rejecting. As long as $p_{xx} > 0$ and $p_{xy} > 0$ for all elements of P, the chain will be irreducible.

To ensure the Markov chain has a π as its stationary distribution, it is sufficient to show the detailed balance condition (A.6) holds. The chain satisfies this condition if and only if for all $x \neq y$,

$$\frac{a_{xy}}{a_{yx}} = \frac{\pi_y\, p_{yx}^{(0)}}{\pi_x\, p_{xy}^{(0)}}\,. \tag{A.17}$$

The Metropolis-Hastings algorithm for MCMC sampling satisfies (A.17) by setting the acceptance probability to

$$a_{xy} = \min\left\{1, \frac{\pi_y\, p_{yx}^{(0)}}{\pi_x\, p_{xy}^{(0)}}\right\}\,. \tag{A.18}$$

Thus, by repeatedly proposing a state y from $p_{xy}^{(0)}$ given the current state x, and accepting it as the next state with probability a_{xy}, we are guaranteed to generate samples from the distribution π once the chain converges.

APPENDIX B

Surface Reconstruction

B.1 Introduction

In this appendix we present our ideas for building an improved surface reconstruction and how it can utilized to improve 3-D structure learning and recognition. For our work on fitting 3-D structure models to biological images (Chapters 2 and 3), visualization of extracted surface points was not the primary focus, so we did not pursue a sophisticated algorithm for surface reconstruction. Instead, we created the visualization for *Alternaria* in Figure 2.3 with a simple approach that utilized gradient information from our surface point detector of Section 2.6.1. Starting with the detected surface points, we create independent, fixed-sized polygons centered at each point. The normal for the polygon is defined by gradient information extracted from the detector. While this produced a reasonable visualization sufficient for low-resolution applications, it does not approach state of the art reconstructions and can be improved significantly.

Automatic surface reconstruction from three-dimensional point clouds has been studied for some time yielding impressive results (Hoppe et al., 1992, 1994; Amenta et al., 1998; Amenta and Bern, 1999; Levin, 2003; Amenta and Kil, 2004; Fleishman et al., 2005). The ideas presented in this appendix go beyond simply applying one of these algorithms. We explain how these approaches can be combined with information from our 3-D structure model to improve both model inference and surface reconstruction. The primary purpose of detailing this information is to provide a guide for improving surface reconstruction when using our 3-D model and for improving 3-D structure learning when we have an estimate of the reconstructed surface.

B.2 Surface reconstruction as data

We first explore arguments for and against fitting a surface to the detected point cloud and using that as the data to infer *Alternaria* structure from. Specifically, we consider two approaches:

1. Fitting geometric primitives directly to the reconstructed surface, without using Bayesian inference of a model;

2. Using the reconstructed surface and fit geometric primitives as a bottom-up, data-driven guide to the sampling approach for model inference.

We begin by considering reconstructing the surface of *Alternaria* and then trying to fit geometric primitives to the surface, not necessarily in a Bayesian way, but perhaps statistically in some sense. From the reconstruction we could extract a skeletal structure, for example the medial axis (Amenta and Bern, 1999; Amenta et al., 1998), which would tell us where the branches occur in the structure, a type of topology. This in itself would be extremely informative. For example, when we start fitting geometric primitives to the images of *Alternaria*, like cylinders and ellipsoids, we would have a good estimate for the topology and basic shape; the branches would be in the approximately correct location. So we could detect the skeleton of the surface, then populate it with cylinders and ellipsoids. This would be an extremely valuable estimate of topology and basic shape. The ability to hypothesize a skeleton of the structure would significantly improve the speed and accuracy of inference by reducing the search space over topologies.

Reconstructing the surface also provides a means to estimate curvature of the structure at arbitrary locations. In Alternaria a high curvature is indicative of a spore in the structure. So where there is a particular, possibly learned, level of curvature on the surface, we could bias the geometric primitive selection to more likely be an ellipsoid. This would increase the chances of hypothesizing the correct primitive at a point in the skeleton of the structure.

Instead of analyzing curvature, we could do something similar in spirit to a Moving Least-Squares (MLS) approach for reconstruction (Amenta and Kil, 2004; Fleishman

et al., 2005; Levin, 2003). The least-squares residuals in MLS reconstruction indicate how well the surface point cloud fits our hypothesis for the surface manifold. We could try an MLS fit of geometric primitives embedded at a specific point on the skeleton. If the fit of a particular geometric primitive has low residuals, we know that it is a good primitive to use; we could decide whether a particular piece of the surface is modeled better by a cylinder or ellipsoid shape. This would provide just as much information as curvature analysis, but now we would have a statistically sound estimate of a good shape at a point on the skeleton. A generalization of this idea would be to integrate the shape-based MLS approach outlined above with the surface recognition MLS into a single algorithm.

A drawback of this approach is that, in order to fit structure, we rely heavily on our ability to generate an accurate surface reconstruction. In our current Bayesian inference process, we fit a structure and imaging model to the whole data set—the image pixels. But for the approach above, if part of the surface reconstruction is incorrect, the fit would inevitably be wrong. Whereas with Bayesian inference of our generative model for the data, we are better able to accommodate noise, e.g. excessive blur from the optics, and hallucinate missing data. Further, if we fit a Bayesian model, we can use it for higher-level inference on questions like which species the structure belongs to; statistical inference across several data-sets could be done to learn the structural form of different species.

Another issue with fitting structure directly to the surface reconstruction is the dependency on accurate point detections. Significant blurring exists in the microscopic images of *Alternaria*, so there is a lot of ambiguity as to where the actual surface points are located within each image in the stack. It is an unrealistic assumption to make that we could extract a very accurate set of points for surface reconstruction. This is particularly true when two or more pieces of the structure are close in the depth direction; the image resolution is much lower in this dimension.

The surface reconstruction process completely disregards the image formation process. Blur in microscopic images is viewed as an obstacle to extracting good surface points from the data for reconstruction. This is in contrast to a generative model and inference approach where we model both the structure and imaging system. We view the image blur as structural information that has been misplaced, i.e. diffracted, by the

imaging system. If we can understand how that blurring happened, we can use it as information about the structure. This concept of using the blur in the image as information does not exist in the surface reconstruction approach.

We could further improve upon the ideas above by utilizing the geometric structure extracted from the surface reconstruction as a data-driven component in the sampler for statistical inference. The basic idea is to apply the approach previously mentioned as a system of proposals for a sampler in a Bayesian framework. This would have all the advantages mentioned above, but without the drawbacks because they are only proposals in a larger statistical inference process. The issues here would be how to define a probability distribution over the extracted structure from the surface reconstruction. But assuming this could be solved in a reasonable way, such a proposal distribution would increase the convergence rate of our sampler significantly by eliminating most of the costly search over topologies.

As discussed below, once we start fitting the model to the data with our Bayesian inference process and the surface reconstruction based proposal distribution, we could update the surface reconstruction with the fit model. This should further improve overall inference, but at the cost of extra computation time to continuously re-estimate the surface manifold.

B.3 Improving surface reconstruction with a 3-D model

In this section, we explore situations where the detected surface points are not sufficient for a good reconstruction, but were enough for an estimate of 3-D model structure. For example, we have an inferred 3-D structure model and a cloud of points detected in the data, but the points are not sampled reliably and/or densely enough to reconstruct an acceptable surface. In this case we can use the model to improve the reconstructed surface from the point cloud.

By utilizing the fit structure model, we could improve the quality of the detected surface points in the data, leading to an improved surface reconstruction. We are currently using gradient thresholding to detect surface points in the data; the detector is a 3-D Canny

surface point detection algorithm. As usual for this algorithm, the gradient threshold value we choose is constant for the entire set of pixels in the stack of images under analysis. But if we have an estimate of where the structure is in the images, then we could adaptively change the gradient threshold during the detection algorithm when near a region close to the "crust" or shell of the structure model. So even though the gradient detection may not have hardly any change in gradient at a particular point, we could change the gradient threshold at that point if the model is located nearby. In other words, we are making the surface point detection more sensitive in areas nearby the structure model.

A second idea for improving surface detection focuses more on the reconstruction algorithm, possibly utilizing the improved point cloud from the previous idea. The structure model gives an estimate of a manifold representing a simplification of the true surface manifold in the data. Using the detected surface points, we could combine them with the model to better estimate the true manifold. In the following description, we assume an estimate for our 3-D structure model is available.

The reconstructed surface is a 2-D manifold embedded in $\mathbb{R}^3$. In the MLS approach we select a surface point s_i from the point cloud and find the best plane, in the least-squared sense, that fits the data, with an orthonormal coordinate system centered at the projection of s_i. We then project all the near-by surface points onto this plane and find the best 2-D polynomial that minimizes the distance between the projected 2-D points evaluated at the polynomial and the distance from the plane to the actual surface points, again in the least-squared sense. In our case, we not only know the detected surface points and their normals, but we already have an estimate of a manifold that is probably nearby the actual surface manifold. The structure model comprises ellipsoids and spores, so we could utilize the analytical form of these objects to propose, or give heavy weighting to, the form of the 2-D polynomial (from the quadrics) we fit to the data in the MLS approach. This should greatly reduce computation time and error in finding the moving least-squares fit of a polynomial to the data.

It is possible that parts of the structure model are fit where there is missing data. Indeed, this is one of the major benefits of inferring a model from the data. So another improvement would be to hallucinate reconstructed surface where there isn't even data,

assuming a correct fit of the model. We could do this generatively from the structure model in areas of the point cloud with low density. This would definitely improve the overall surface reconstruction; we would be reconstructing surface from what appears noise or nothing at all.

Finally, having the model should enable heavy smoothing of the data without worrying about losing fine detail; the model is there to enforce sharp edges and detail. We could easily improve upon the work of Fleishman et al. (2005) to detect where polynomial-based reconstructions should be broken apart to make sharp edges. In essence, what we would be able to do better than other techniques is resolve the ambiguity between noise in the data and fine detail in the structure.

APPENDIX C

Likelihood Weight Estimation

C.1 Introduction

For the image likelihood of Chapter 4, we introduce a set of binary weights to classify pixels based on output from an edge detector. Each pixel is classified according to whether it contains an edge point from the model, if it has an edge from noise, or is missing a detected edge point that the structure model suggests should be detected. Chapter 5 continues the approach of making binary assignments for edge detection type made at a pixel, although in a different formulation. In this appendix, we show how the problem of estimating weight parameters in inverse sequence alignment is similar to finding continuous weights in a pixel alignment problem for likelihood computation. The ideas presented here are intended to provide the groundwork for improving pixel classification and weight estimation in our image likelihood models of Sections 4.2.3 and 5.2.3.

C.2 Inverse alignment background

Given a collection of aligned sequences, the goal of inverse parametric sequence alignment is to to learn parameters that optimize a linear scoring function of those alignments. Although it may not be possible to find a set of parameters that yields an optimal score for all collections of aligned sequences, the problem and solution are formulated in such a way that the parameters closest to optimality, within some ϵ, are recovered.

The algorithm presented in Kececioglu and Kim (2006) and Kim and Kececioglu (2008) for learning the scoring function parameters is quite general. They show that if the objective function of an optimization problem is linear in its parameters, then a linear program that has exponentially many constraints (inequalities) can be solved in polynomial time. This opens up the possibility of applying the algorithm to many problems with similar parametrization and constraint sets, like pixel classification, as we will describe.

For the problem of aligning a set of sequential objects $\mathcal{S}$, like two strings, learning the alignment scoring function parameters through linear programming is formalized as follows. Let the scoring function for the alignment $\mathcal{A}$ of the sequences in $\mathcal{S}$ be defined as

$$f(\mathcal{A}) = f_0(\mathcal{A}) + f_1(\mathcal{A})w_1 + \cdots + f_p(\mathcal{A})w_p\,, \tag{C.1}$$

where p is the number of parameters, and the sub-scoring functions $f_i(\cdot)$ measure features of the alignment, like substitutions; the w_i are the parameters, or weights, we are interested in learning. Now suppose the alignment $\mathcal{A}$ is the optimal alignment for the sequences in $\mathcal{S}$. Then for every other alignment $\mathcal{B}$ of $\mathcal{S}$, the following inequality holds

$$f(\mathcal{A}) \geq f(\mathcal{B})\,, \tag{C.2}$$

assuming optimality translates to maximizing the scoring function. Since the inequality for each $\mathcal{B}$ is linear in its parameters, we can formulate finding the parameters as a linear program, given some objective function, that should also be linear in its parameters.

Depending on the type of alignment operations defined between the sequences in $\mathcal{S}$, it is possible to have an exponential number of other non-optimal alignments and, hence, an exponential number of inequalities. For the case of the sequences in $\mathcal{S}$ being two strings with substitution and gap differences, there are in fact $\Omega(4^n)$ inequalities (Kececioglu and Kim, 2006). It turns out, however, that by making clever use of the Separation Theorem and designing a cutting plane algorithm, this linear program can still be solved in polynomial time, under certain conditions, such as a linear objective function.

The constraints in the linear program are a set of intersecting half-spaces that define a polyhedron $\mathcal{P}$ (a potentially irrational and unbounded polyhedron, but suppose for simplicity it is not). For linear programming with an objective function $\max\{\mathbf{c}\cdot\mathbf{x}\}$, we want to find the point $\mathbf{x} \in \mathbb{R}^d$ that has maximal projection on the vector $\mathbf{c}$ and is in the polyhedron $\mathcal{P}$. This is an optimization problem, and according to the Separation Theorem, is equivalent to the separation problem. The latter is defined as follows. For a point $\mathbf{y} \in \mathbb{R}^d$, decide whether $\mathbf{y}$ is inside $\mathcal{P}$; if it isn't, find the hyperplane separating $\mathbf{y}$ from all the points that are in $\mathcal{P}$. The equivalence is highlighted by the fact that if one of the problems

can be solved in polynomial time, the other can as well.

Under several constraints, including that the alignment scoring function is linear in its parameters, the inverse alignment problem can be solved in polynomial time by giving a separation algorithm to solve the separation problem in polynomial time (Kececioglu and Kim, 2006). Specifically, a cutting plane algorithm exists that solves the separation problem for a polyhedron in, for most practical purposes, polynomial time.

One detail not explicitly summarized yet is how to handle over-fitting parameter values to each set of sequences. As was described, for each set of sequences $\mathcal{S}_i$ there is an optimal alignment $\mathcal{A}_i^*$. The parameters learned for the scoring function under this alignment, however, may not be shared for the optimal alignment of some other sequence. To ameliorate this, an alignment is said to be near optimal and considered acceptable if its score is within some ϵ of the optimal alignment

$$f(\mathcal{A}_i) \geq (1-\epsilon) f(\mathcal{A}_i^*) \ , \tag{C.3}$$

where ϵ is chosen as close to zero as possible and fixed for all sets of sequences $\mathcal{S}_i$. This is referred to an ϵ-optimal alignment and enables learning a set of parameters that are a best-fit across a whole collection of sequences. It has been shown that this relaxed linear programming problem can also be solved in polynomial running-time.

The results in Kececioglu and Kim (2006) are convincing and show that the algorithm can find an ϵ-optimal set of parameters for a data set in a reasonable amount of time. For each training data set, a convex combination of the parameters at the extremes of the optimization function gave a good estimate of the parameters found in the test set. In fact, the midpoint of the convex combination in the training sets seems to generalize well to the test sets.

Kim and Kececioglu (2008) builds upon the previous work and shows that including accommodations for noise and missing sequence data increases performance further. A more useful error measure is also introduced in the paper and used in the testing framework. Finally, by including cross-validation in the evaluation, it is apparent that the problem of over-fitting is not a significant issue.

C.3 Pixel alignment problem

Our research is focused on developing efficient inference algorithms for fitting three-dimensional object models to single view images. Since a 3-D object model can be projected into an image under widely varying views, we fit a constrained camera model as well. In fact, we fit the models simultaneously. The details of our object and camera model can be found in Chapter 4 and are omitted here. We begin by stating that we have a 3-D object model, e.g. a table, that has a large set of parameters. We also have some concept of a perspective projective view, e.g. a camera, of that model that has a set of parameters, such as focal length, position, and orientation. We can use this camera model to project a view of the 3-D object model into a 2-D image. This process comprises our generative model for image data and is what we use to fit the models to images using Bayesian statistical inference. Our object models are wire-frame and we try to fit them to detected edge points in the images.

The overall goals of this research are: (1) develop efficient inference algorithms to fit object and camera model parameters to a given image, and (2) use this inference process to learn 3-D object models from a statistical representation of geometry that we develop by inferring shared 3-D structure.

Let us summarize Section 4.2.3 and be a bit more specific about the inference problem we are trying to solve. Our generative process for an image is pixel based; we assume each pixel in the projected model image independently generates a pixel value in the data. More precisely, a data pixel value results from one of four processes: an edge in the object model, image background, noise from clutter, or missing data (the edge detector and object model are not perfect). Given an image $\mathcal{I}$ and object and camera model parameters $\boldsymbol{\theta}$, we can construct a projected model image $\mathcal{I}_\theta$ from the wire-frame of the object under the camera, with hidden lines removed. We then model detected edge points in the data image as generated by the lines in $\mathcal{I}_\theta$. Please see Figures 4.1 and 4.2 for an illustration.

We measure the goodness of fit of a model to data with a likelihood function. Our likelihood function for a given image $\mathcal{I}$ and varying model $\boldsymbol{\theta}$ can be written as

$$L(\mathcal{I} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{i=1}^{4} p_i(\mathcal{I}_n \mid \boldsymbol{\theta})\, w_i \ , \tag{C.4}$$

where N is the number of pixels in the input image; the density functions p_i and fixed weights w_i correspond to each component of the generative process for a pixel. The likelihood is not a probability distribution, but it is a density function that fits into our Bayesian statistical inference framework. What we would like to learn from some training data are good estimates of the weights w_i, which we will call parameters of the likelihood function.

One thing to notice about our likelihood function is that it is not linear in its parameters. Even if we take the log of the function and reduce the product term to a sum, the parameters are trapped inside the summed logarithm functions. So our idea is pretty simple, we are going to learn weights under a linear objective and scoring function that are semantically similar to the ones in the likelihood, but not exactly the same. To do this, we rearranged the problem a bit to make it possible to solve with the inverse parametric alignment approach.

To formulate our (alternative) pixel classification problem we will use similar notation and ideas from the above referenced papers. For example, the concept of sequence alignment is extended to aligning pixels in images. Conceptually, the alignment between a data image $\mathcal{I}$ and a model image $\mathcal{I}_\theta$ is pixel-to-pixel [1]. But what is more important is knowing which of the four types of generative processes links each pixel-to-pixel alignment, e.g. a pixel in $\mathcal{I}$ could be linked to an edge, background, noise, or missing data pixel in $\mathcal{I}_\theta$.

Let $\mathcal{A}$ be a pixel alignment between the data and model images $\mathcal{I}$ and $\mathcal{I}_\theta$. One idea for the score of a pixel alignment comprises a linear weighting of the sub-scoring functions

$f_1(\mathcal{A})$ — the number of edge pixels in $\mathcal{I}$ generated by model edges in $\mathcal{I}_\theta$,

$f_2(\mathcal{A})$ — the number of background pixels in $\mathcal{I}$ generated by background in $\mathcal{I}_\theta$,

$f_3(\mathcal{A})$ — the number of noise pixels in $\mathcal{I}$,

[1] Note that the pixel-to-pixel alignment referred to here is not limited to homogeneous indexes between the data and model images. Rather, an edge point pixel in the data could be aligned with a nearby edge point pixel in the model.

$f_4(\mathcal{A})$ — the number of pixels in $\mathcal{I}$ generated by unobserved (missing) edges in $\mathcal{I}_\theta$.

The functions f_3 and f_4 are actually a type of penalty against the overall score; a large amount of noise or missing data is most likely negative evidence for a good fit. Then the complete scoring function for an alignment $\mathcal{A}$ is

$$f(\mathcal{A}) = f_1(\mathcal{A})\, w_1 + f_2(\mathcal{A})\, w_2 - f_3(\mathcal{A})\, w_3 - f_4(\mathcal{A})\, w_4 \ . \tag{C.5}$$

We acknowledge that this is perhaps not the best set of scoring functions that could be used in this situation, but it seems like they will map fairly well to our likelihood function.

Using the scoring function (C.5), we can define the set of constraints for our parameter optimization problem. For some maximum-score alignment $\mathcal{A}$ between data and model images $\mathcal{I}$ and $\mathcal{I}_\theta$, every other alignment $\mathcal{B}$ obeys the inequality

$$f(\mathcal{A}) \geq f(\mathcal{B}) \ . \tag{C.6}$$

Since there are four types of alignment per pixel, there will be an exponential number of other alignments $\mathcal{B}$ and inequalities like (C.6). Indeed, there are exactly 4^N of them, where N is the number of pixels in the data and model images. For some clever choice(s) of the parameters w_i, the alignment $\mathcal{A}$ maximizes the scoring function, subject to all of these inequalities. As with sequence alignment, we can use Separation Theorem to solve a linear program in polynomial time to find a good choice of parameters.

Given an optimal alignment $\mathcal{A}$, the scoring function (C.5) and exponential number of inequalities like (C.6), we can formulate the problem of finding the parameters w_i in the scoring function as a linear program. This is because the inequalities are actually linear in their parameters, making them the constraints in the linear program. We will also need to develop an objective function to maximize under the linear program.

The problem with solving this linear program, though, is the exponential number of constraints. But as was shown in the papers and previously discussed, as long as certain conditions are met, such as a linear objective function, we can solve this in polynomial time. It is not clear exactly what the best objective function would be in the case of pixel alignment, but perhaps a good function would be $\max\{w_1 + w_2 - w_3 - w_4\}$.

The linear programming optimization problem is solvable in polynomial by applying the Separation Theorem. As long as we can solve an equivalent separation problem in polynomial time, the Separation Theorem essentially says that we can solve the optimization problem in the same time. We can design a separation algorithm to solve this problem, a cutting-plane algorithm, and apply that. Since our formulation of the problem for images is so close to the sequence alignment problem, we believe that the cutting plane algorithm implementation should be very similar.

For a whole set of images with given object and camera models that fit them best, we could use our cutting-plane algorithm to solve for the optimization problem and find the best set of parameters w_i. For reasons previously discussed, the parameters would not be optimal for all images in the set, but ϵ-optimal. Our idea then is to take those learned parameters and use them in the likelihood computation (C.4).

It is unfortunate that the ϵ-optimal weights recovered in our linear programming application do not have a direct link to the weights in the likelihood function (C.4). They are semantically similar, however, and would provide good estimates and an interesting alternative to the hard assignments made in Section 4.2.3.

REFERENCES

Al-Awadhi, F. (2001). *Statistical image analysis and confocal microscopy*. Ph.D. thesis, Department of Mathematical Sciences, University of Bath, Bath, UK.

Al-Awadhi, F., C. Jennison, and M. Hurn (2004). Statistical image analysis for a confocal microscopy two-dimensional section of cartilage growth. *Journal of the Royal Statistical Society: Applied Statistics*, **53**(1), pp. 31–49.

Al-Kofahi, K. A., A. Can, S. Lasek, D. H. Szarowski, N. Dowell-Mesfin, W. Shain, J. N. Turner, and B. Roysam (2003). Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *IEEE Transactions on Information Technology in Biomedicine*, **7**(4), pp. 302–317.

Al-Kofahi, K. A., S. Lasek, D. H. Szarowski, C. J. Pace, G. Nage, J. N. Turner, and B. Roysam (2002). Rapid automated 3D tracing of neurons from confocal image stacks. *IEEE Transactions on Information Technology in Biomedicine*, **6**(2), pp. 171–187.

Allen, M. T., P. Prusinkiewicz, and T. M. DeJong (2005). Using L-systems for modeling sourceink interactions, architecture and physiology of growing trees: the L-PEACH model. *New Phytologist*, **166**(3), pp. 869 – 880.

Amenta, N. and M. Bern (1999). Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, **22**, pp. 481–504.

Amenta, N., M. Bern, and M. Kamvysselis (1998). A new Voronoi-based surface reconstruction algorithm. In *SIGGRAPH 1998: Proceedings of the conference on computer graphics and interactive techniques*, pp. 415–421.

Amenta, N. and Y. J. Kil (2004). Defining point-set surfaces. In *SIGGRAPH 2004: Proceedings of the conference on computer graphics and interactive techniques*, volume 23, pp. 264–270.

Andrieu, C., N. de Freitas, A. Doucet, and M. I. Jordan (2001). An introduction to MCMC for machine learning. *Machine Learning*, **50**(1), pp. 5–43.

Belongie, S. and J. Malik (2001). Matching shapes. In *International Conference on Computer Vision*, pp. 454–461.

Berg, A. C. and J. Malik (2001). Geometric blur for template matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 607–615.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, **94**(2), pp. 115–147.

Binford, T. O. (1971). Visual perception by computer. In *IEEE Systems Science and Cybernetics Conference*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, **3**, pp. 993–1022.

Borgefors, G. (1988). Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(6), pp. 849–865.

Boxall, E. S., N. S. White, and G. S. Benham (1994). The processing of three-dimensional confocal data sets. In Cheng, P. C., T. H. Lin, W. L. Wu, and J. L. Wu (eds.) *Multidimensional microscopy*. Springer-Verlag.

Brooks, R. A. (1981). Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, **17**, pp. 285–348.

Bussi, G. and M. Parrinello (2007). Accurate sampling using Langevin dynamics. *Physical Review E*, **75**, p. 056707.

Can, A., H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam (1999). Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. *IEEE Transactions on Information Technology in Biomedicine*, **3**(2), pp. 125–138.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, pp. 679–698.

Carasso, A. S. (2001). Direct Blind Deconvolution. *SIAM Journal on Applied Mathematics*, **61**(6), pp. 1980–2007.

Chen, H., J. R. Swedlow, M. Grote, J. W. Sedat, and D. A. Agard (1995). The collection, processing, and display of digital three-dimensional images of biological specimens. In Pawley, J. B. (ed.) *Handbook of biological confocal microscopy*, pp. 197–210. Plenum Press, New York, NY.

Cheng, P. C., T. H. Lin, W. L. Wu, and J. L. Wu (eds.) (1994). *Multidimensional microscopy*. Springer-Verlag.

Clowes, M. B. (1971). On seeing things. *Artificial Intelligence*, **2**(1), pp. 79–116.

Conchello, J. (1998). Superresolution and convergence properties of the expectation-maximization algorithm for maximum-likelihood deconvolution of incoherent images. *Journal of Optical Society of America A*, **15**(10), pp. 2609–2619.

Conchello, J., J. J. Kim, and E. W. Hanson (1994). Enhanced three-dimensional reconstruction from confocal scanning microscope images. II: Depth discrimination versus signal-to-noise ratio in partially confocal images. *Applied Optics*, **33**(17), pp. 3740–3750.

Conn, P. M. (ed.) (1999). *Confocal microscopy*, volume 307 of *Methods in enzymology*. Academic Press, San Diego, CA.

Crandall, D. J. and D. P. Huttenlocher (2006). Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, pp. 16–29.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893.

Deussen, O., P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz (1998). Realistic modeling and rendering of plant ecosystems. In *SIGGRAPH 1998: Proceedings of the conference on computer graphics and interactive techniques*, pp. 275–286. ACM Press, New York, NY, USA. ISBN 0-89791-999-8. doi:http://doi.acm.org/10.1145/280814.280898.

Escuela, G., G. Ochoa, and N. Krasnogor (2005). *Evolving L-Systems to capture protein structure native conformations*, volume Volume 3447/2005 of *Lecture Notes in Computer Science*, pp. 74–84. Springer-Verlag, New York NY, 1 edition.

Fei-Fei, L., R. Fergus, and P. Perona (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *International Conference on Computer Vision*, volume 2, pp. 1134–1141.

Fei-Fei, L., R. Fergus, and P. Perona (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 178–178.

Fergus, R., P. Perona, and A. Zisserman (2003). Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 264–271.

Ferrari, V., L. Fevrier, F. Jurie, and C. Schmid (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(1), pp. 36–51.

Ferrari, V., F. Jurie, and C. Schmid (2009). From images to shape models for object detection. *International Journal of Computer Vision*, pp. 1–20.

Fleishman, S., D. Cohen-Or, and C. T. Silva (2005). Robust Moving Least-squares Fitting with Sharp Features. In *SIGGRAPH 2005: Proceedings of the conference on computer graphics and interactive techniques*, volume 24, pp. 544–552.

Forsyth, D. A., J. Haddon, and S. Ioffe (2001). The Joy of Sampling. *International Journal of Computer Vision*, **41**(1-2), pp. 109–134.

Forsyth, D. A. and J. Ponce (2002). *Computer vision*, chapter 8, pp. 165–188. Prentice Hall.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**(4), pp. 711–732.

Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo. In *Highly Structured Stochastic Systems*. Oxford University Press.

Grenander, U. and M. I. Miller (1994). Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B*, **56**(4), pp. 549–603.

Gui-Ting, L., Q. Yu-Zhen, Z. Peng, D. Wei-Hua, Q. Yuan-Ming, and G. Hong-Tao (1992). Etiological role of *Alternaria* alternata in human esophageal cancer. *Chin Medical Journal*, **105**(5), pp. 394–400.

Hammel, M., P. Prusinkiewicz, W. Remphrey, and C. Davidson (1995). Simulating the development of Fraxinus pennsylvanica shoots using L-systems. In *Sixth Western Computer Graphics Symposium*, pp. 49–58.

Han, F. and S.-C. Zhu (2005). Bottom-up/top-down image parsing by attribute graph grammar. In *International Conference on Computer Vision*, volume 2, pp. 1778–1785.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, pp. 97–109.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, **42**, pp. 177–196.

Hoiem, D., A. A. Efros, and M. Hebert (2005). Geometric context from a single image. In *International Conference on Computer Vision*, pp. 654–661.

Hoiem, D., A. A. Efros, and M. Hebert (2006). Putting objects in perspective. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2137–2144.

Hoiem, D., C. Rother, and J. Winn (2007). 3D layoutCRF for multi-view object class recognition and segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Holmes, T. J. (1989). Expectation-maximization restoration of band-limited, truncated point-process intensities with application in microscopy. *Journal of Optical Society of America A*, **6**(7), pp. 1006–1014.

Holmes, T. J. (1992). Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach. *Journal of Optical Society of America A*, **9**, pp. 1052–1061.

Hoppe, H., T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle (1994). Piecewise smooth surface reconstruction. In *SIGGRAPH 1994: Proceedings of the conference on computer graphics and interactive techniques*, pp. 295–302.

Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle (1992). Surface reconstruction from unorganized points. In *SIGGRAPH 1992: Proceedings of the conference on computer graphics and interactive techniques*, pp. 71–78.

Huttenlocher, D. P., G. A. Klanderman, and W. J. Rucklidge (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(9).

Huttenlocher, D. P. and S. Ullman (1990). Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, **5**(2), pp. 195–212.

Kadir, T., A. Zisserman, and M. Brady (2004). An affine invariant salient region detector. In *European Conference on Computer Vision*, pp. 228–241.

Kaess, M., R. Zboinski, and F. Dellaert (2004). MCMC-based multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points. In *Lecture Notes in Computer Science*, volume 3023, pp. 329–341.

Kececioglu, J. and E. Kim (2006). Simple and fast inverse alignment. In *Research in Computational Molecular Biology*, pp. 441–455.

Kemp, C. and J. B. Tenenbaum (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*, **105**(31), pp. 10687–10692.

Kim, E. and J. Kececioglu (2008). Learning parameters for sequence alignment from examples with missing data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **5**(4), pp. 546–556.

Kong, S. K., S. Ko, C. Y. Lee, and P. Y. Lui (1999). Practical considerations in acquiring biological signals from confocal microscope. In Conn, P. M. (ed.) *Confocal microscopy*, volume 307 of *Methods in enzymology*, pp. 20–26. Academic Press, San Diego, CA.

Kushal, A., C. Schmid, and J. Ponce (2007). Flexible object models for category-level 3D object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Leibe, B., A. Leonardis, and B. Schiele (2004). Combined object categorization and segmentation with an implicit shape model. In *European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision*.

Leordeanu, M., M. Hebert, and R. Sukthankar (2007). Beyond local appearance: category recognition from pairwise interactions of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Levin, D. (2003). Mesh-independent surface interpolation. In Brunnett, G., B. Hamann, H. Muller, and L. Linsen (eds.) *Geometric modeling for scientific visualization*, pp. 37–49. Spring-Verlag.

Liebelt, J., C. Schmid, and K. Schertler (2008). Viewpoint-independent object class detection using 3D feature maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, **18**(3), pp. 280–299, 300–315.

Lindenmayer, A. (1975). Developmental algorithms for multicellular organisms: A survey of L-systems. *Journal of Theoretical Biology*, **54**(1), pp. 3–22.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag.

Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, **31**(3), pp. 355–395.

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, pp. 441–450.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**(2), pp. 91–110.

Manning, C. D. and H. Schutze (1999). *Foundations of statistical natural language processing*, chapter 11, pp. 381–405. The MIT Press.

Markham, J. and J. Conchello (2001). Fast maximum-likelihood image-restoration algorithms for threedimensional fluorescence microscopy. *Journal of Optical Society of America A*, **18**(5), pp. 1062–1071.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, pp. 1087–1092.

Müller, K. (1980). Reaction paths on multidimensional energy hypersurfaces. *Angewandte Chemie*, **19**(1), pp. 1–13.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto.

Opelt, A., A. Pinz, and A. Zisserman (2008). Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, **80**(1), pp. 16–44.

Oppenheimer, P. E. (1986). Real time design and animation of fractal plants and trees. In *SIGGRAPH 1986: Proceedings of the conference on computer graphics and interactive techniques*, pp. 55–64. ACM Press, New York, NY, USA. ISBN 0-89791-196-2. doi: http://doi.acm.org/10.1145/15922.15892.

Pawley, J. B. (ed.) (1995). *Handbook of biological confocal microscopy*. Plenum Press, New York, NY.

Pentland, A. P. (1987). Recognition by parts. In *International Conference on Computer Vision*, pp. 612–620.

Pentland, A. P. (1990). Automatic extraction of deformable part models. *International Journal of Computer Vision*, **4**(2), pp. 107–126.

Pope, A. R. and D. G. Lowe (1996). Learning object recognition models from images. In Nayar, S. and T. Poggio (eds.) *Early Visual Learning*, pp. 67–97. Oxford University Press.

Preza, C. and J. Conchello (2004). Depth-variant maximum-likelihood restoration for three-dimensional fluorescence microscopy. *Journal of Optical Society of America A*, **21**(9), pp. 1593–1601.

Prusinkiewicz, P. and A. Lindenmayer (1990). *The algorithmic beauty of plants*. Springer-Verlag.

Prusinkiewicz, P., A. Lindenmayer, and J. Hanan (1988). Development models of herbaceous plants for computer imagery purposes. In *SIGGRAPH 1988: Proceedings of the conference on computer graphics and interactive techniques*, pp. 141–150. ACM

Press, New York, NY, USA. ISBN 0-89791-275-6. doi:http://doi.acm.org/10.1145/54852.378503.

Samal, A., B. Peterson, and D. J. Holliday (2002). Recognition of plants using a stochastic L-system model. *Journal of Electronic Imaging*, **11**(1), pp. 50–58.

Samarabandu, J. K., R. Acharya, and P.-c. Cheng (1994). Analysis and presentation of three dimensional data sets. In Cheng, P. C., T. H. Lin, W. L. Wu, and J. L. Wu (eds.) *Multidimensional microscopy*. Springer-Verlag.

Savarese, S. and L. Fei-Fei (2007). 3D generic object categorization, localization and pose estimation. In *International Conference on Computer Vision*.

Savarese, S. and L. Fei-Fei (2008). View synthesis for recognizing unseen poses of object classes. In *European Conference on Computer Vision*, pp. 602–615.

Saxena, A., S. H. Chung, and A. Ng (2005). Learning depth from single monocular images. In Weiss, Y., B. Schölkopf, and J. Platt (eds.) *Advances in Neural Information Processing Systems*, pp. 1161–1168. MIT Press, Cambridge, MA.

Schlecht, J. and K. Barnard (2009a). Learning models of object structure. In *Advances in Neural Information Processing Systems*.

Schlecht, J. and K. Barnard (2009b). Learning models of object structure. Technical report, University of Arizona.

Schlecht, J., K. Barnard, and B. Pryor (2006). Statistical inference of biological structure and point spread functions in 3D microscopy. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 373–380.

Schlecht, J., K. Barnard, E. Spriggs, and B. Pryor (2007). Inferring grammar-based structure models in 3D microscopy data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Shaw, P. J. and D. J. Rawlins (1991). The point-spread function of a confocal microscope: its measurement and use in deconvolution of 3-D data. *Journal of Microscopy*, **163**(2), pp. 151–165.

Shotton, J., A. Blake, and R. Cipolla (2005). Contour-based learning for object detection. In *International Conference on Computer Vision*, pp. 503–510.

Simmons, E. (1999). Alternaria themes and variations (287-304): Species on caryophyllaceae. *Mycotaxon*, **70**, pp. 325–369.

Sivic, J., B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman (2005). Discovering objects and their location in images. In *International Conference on Computer Vision*, pp. 370–377.

Sminchisescu, C. and B. Triggs (2001). Covariance scaled sampling for monocular 3D body tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 447–454.

Sminchisescu, C. and B. Triggs (2002). Hyperdynamics importance sampling. In *European Conference on Computer Vision*, pp. 769–783.

Sminchisescu, C. and B. Triggs (2003). Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, **22**(6), pp. 371–391.

Sokal, A. D. (1989). Monte Carlo methods in statistical mechanics: Foundations and new algorithms. Lecture notes.

Song, M., R. M. Haralick, F. H. Sheehan, and R. K. Johnson (2002). Integrated surface model optimization for freehand three-dimensional echocardiography. *IEEE Transactions on Medical Imaging*, **21**(9), pp. 1077–1090.

Spriggs, E. (2007). 3D Fungus Generator. Available through the world-wide-web at http://vision.cs.arizona.edu/taralove/lsystem.html.

Spriggs, E., J. Schlecht, K. Barnard, and B. Pryor (2007). dimensional structure in Alternaria and applications to morphometric analysis. In *The Annual Meeting of the Mycologicial Society of America*.

Sudderth, E. B., A. Torralba, W. T. Freeman, and A. S. Willsky (2005). Learning hierarchical models of scenes, objects, and parts. In *International Conference on Computer Vision*, volume 2, pp. 1331–1338.

Sugihara, K. (1984). A necessary and sufficient condition for a picture to represent a polyhedral scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(5), pp. 578–586.

Tenenbaum, J. B., T. L. Griffiths, and C. Kemp (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, **10**(7), pp. 309–318.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, **22**(4), pp. 1701–1762.

Torralba, A., K. P. Murphy, and W. T. Freeman (2004). Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–769.

Tu, Z., X. Chen, A. L. Yuille, and S.-C. Zhu (2005). Image parsing: unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, **63**(2), pp. 113–140.

Tu, Z., S.-C. Zhu, and H. Shum (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5), pp. 657–673.

Verlet, L. (1967). Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, **159**(98).

Verlet, L. (1968). Computer "experiments" on classical fluids. II. Equilibrium correlation functions. *Physical Review*, **165**(201).

Voter, A. F. (1997a). Hyperdynamics: accelerated molecular dynamics of infrequent events. *Physical Review Letters*, **78**(20), pp. 3908–3911.

Voter, A. F. (1997b). A method for accelerating the molecular dynamics simulation of infrequent events. *Journal of Chemical Physics*, **106**(11), pp. 4665–4677.

Webb, R. H. (1999). Theoretical basis of confocal microscopy. In Conn, P. M. (ed.) *Confocal microscopy*, volume 307 of *Methods in enzymology*, pp. 3–20. Academic Press, San Diego, CA.

Weber, J. and J. Penn (1995). Creation and rendering of realistic trees. In *SIGGRAPH 1995: Proceedings of the conference on computer graphics and interactive techniques*, pp. 119–128. ACM Press, New York, NY, USA. ISBN 0-89791-701-4.

Wilken-Jensen, K. and S. Gravesen (1984). *Atlas of moulds in Europe causing respiratory allergy*. ASK Publishing, Copenhagen, Denmark.

Wilson, C. and M. Wisniewski (1994). *Biological control of postharvest diseases: theory and practice*. CRC Press Inc., Boca Raton, Fla.

Winston, P. H. (1975). Learning structural descriptions from examples. In Winston, P. H. (ed.) *The psychology of computer vision*, pp. 157–209. McGraw-Hill.

Zhu, L., Y. Chen, and A. Yuille (2006). Unsupervised learning of a probabilistic grammar for object detection and parsing. In *Advances in Neural Information Processing Systems*, 19, pp. 1617–1624.

Zhu, S.-C. and D. Mumford (2006). A Stochastic Grammar of Images. *Foundations and Trends in Computer Graphics and Vision*, **4**(2), pp. 259–362.

Zhu, S.-C., R. Zhang, and Z. Tu (2000). Integrating top-down/bottom-up for object recognition by data driven Markov chain Monte Carlo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.