Abstract

Motivated by problems in image processing involving segmentation and detection of multiple instances of complex objects, this dissertation explores the use of marked Poisson point processes within a Bayesian nonparametric framework. The Poisson point process underlies a wide range of combinatorial stochastic processes and as such has been a key object driving research in Bayesian nonparametrics. We explore Poisson point processes in combination with probabilistic shape and appearance priors for detection/segmentation of objects/patterns in 1D, 2D and 3D frameworks. This probabilistic formulation encompasses uncertainty in number, location, shape and appearance of the feature of interest, be it in images or time-series data (detection/segmentation of objects of interest). In images, this model can simultaneously detect and segment objects of interest.

The generative process of the model can be explained as sampling a random number of objects at random locations from a Poisson process. The shape of each object is sampled from a shape model. The appearance inside and outside the shape boundary is sampled from an appearance model with foreground and background parameters respectively.

The Poisson intensity parameter can either be homogeneous (uniform) or non-homogeneous. A non-homogeneous Poisson prior provides the flexibility to incorporate spatial context information regarding where the high or low concentration areas occur. We model the non-homogeneous Poisson intensity with a log-Gaussian Cox process. For shape, any probabilistic model can be used. We describe examples of both, parametric and complex shape priors. Appearance features can be simple intensity values of the image or higher level features such as texture.
Inference on the proposed model is complicated due to changing model order, use of non-conjugate priors, and a likelihood that depends on partitioning based on shape boundaries. Inference on such models typically involves a reversible-jump Markov chain Monte Carlo (RJMCMC). We introduce a simpler Gibbs sampling approach which can be accomplished by leveraging the finite nature of digital images. We demonstrate empirical results on 2D images. We also generalize and extend our model with Gibbs sampling on 1D and 3D data. We show case studies of our method on a diverse set of images: cell image segmentation, traffic surveillance, and 3D segmentation of the dermal-epidermal junction of reflectance confocal microscopy images of human skin to aid in cancer detection.

We also worked on the Versatile Onboard Traffic Embedded Roaming Sensors (VOTERS) project as a part of this dissertation. VOTERS aims to detect pavement quality using the data captured by several sensors mounted on a vehicle. The goal is to design a non-invasive technique that can assess the pavement quality without disrupting regular traffic. We have developed algorithms to detect surface defects (for e.g. cracks, along with their type, length, width, and area covered) from video data. These features form a key component in the determination of pavement condition by Civil Engineers. Data is acquired from the video camera using a software trigger developed to capture images at regular intervals of distance rather than time resulting in efficient use of hard-disk space. We present a quantified and thorough analysis using groundtruth data that will be made publicly available.
Acknowledgement

This dissertation was made possible with the help and support from professors, research staff, graduate students, friends, family and coffee shops of Boston and Cambridge.

I thank my advisor Jennifer Dy for her guidance and encouragement. Her high frequency of meetings, personal attention, suggestions and support have helped me immensely to mature as a graduate student. I thank her for giving me the opportunity to be a part of her talented research team and graduate students.

I thank Ralf Birken, Ming Wang, Sara Wadia-Fascetti, and the VOTERS project for accepting me into their program and mentoring me. The VOTERS project gave me the opportunity to peek into a wide range of research developments in other fields as well. I thank my co-advisor Ralf Birken for his guidance and patience in helping me through the graduate study.

I thank the BSPIRAL group and specifically professors, Dana Brooks and Deniz Erdogmus for encouraging us to interact and present a wide variety of materials to the group. It has helped me learn the art of digesting new material quickly and asking right questions. I thank Michael Jordan for his ideas and help in making this dissertation possible. I thank Milind Rajadhyaksha and Kivanc Kose for their support and help in providing me data and their insight. I thank Gunar Schirmer for his guidance and mentoring that helped me understand and develop the camera system.
in VOTERS project.

I thank my friends Rameez Ahmed, Foram Thakkar, Abhishek Dey and Shruti (also my room-
mate) for their support, and making my graduate life in Boston memorable. I thank my lab mates, 
Yale Chang, Junxiang Chen, Sarah Brown, Hanjiao Qiu, Salar Sahini Shamsabadi and Jamshid 
Sourati for their help and support. Seeing them work very hard all the time motivated and pushed 
me to do better. I thank Donglin Niu and Sila Kurugol for their help and discussions that helped 
me make progress.

Finally, I thank my parents G.G. Krishna and G.N.Malleswari, who encouraged and supported 
me in my pursuit for Ph.D. I especially thank my mother for believing in me and spending an im-
mense amount of time mentoring me in my childhood (when I was a terrible student), which helped 
me build a strong foundation, and pursue higher studies. I thank my father for his encouragement, 
support and the tremendous enthusiasm he showed in my research.
Contents

Abstract iii

Acknowledgement i

Notation 1

1 Introduction 2
  1.1 Contributions ................................................................. 4
  1.2 Thesis Organization .......................................................... 6

2 Review of Related Work 7
  2.1 Bayesian Nonparametrics ...................................................... 7
  2.2 Poisson Processes .............................................................. 9
  2.3 Object Detection and Segmentation ........................................... 11
  2.4 Inference ........................................................................ 12

3 Background 14
  3.1 Poisson Process ................................................................. 14
    3.1.1 Properties of a Poisson Process ........................................ 17
3.1.2 Inference on Intensity Parameter of Poisson Process .......................... 19
3.2 Poisson Process for Bayesian Nonparametrics ................................. 23
3.3 Sampling Techniques ........................................................... 24
    3.3.1 Reversible Jump Markov Chain Monte Carlo ......................... 26
    3.3.2 Gibbs Sampling ........................................................... 27

4 Latent Marked Poisson Process ................................................. 29
    4.1 Model formulation .......................................................... 29
        4.1.1 Spatial Poisson Process Prior ...................................... 31
        4.1.2 Shape Prior ............................................................ 33
        4.1.3 Feature/Appearance Model ......................................... 35

5 Inference ............................................................................. 36
    5.1 Reversible Jump Markov Chain Monte Carlo Inference ................. 37
        5.1.1 Inference on the Complete Model ................................ 38
    5.2 Gibbs Sampling ............................................................. 40
        5.2.1 Leveraging the Finite Nature of Image Pixels ..................... 41
        5.2.2 Complete Spatial Randomness ...................................... 42
    5.3 Inference of the Non-homogeneous Poisson Prior ....................... 49

6 Extensions to the Model .......................................................... 53
    6.1 Multiple Categories with Same Appearance Model. ....................... 53
    6.2 Multiple Categories with Different Appearance Model. ................ 56

7 2D case studies ................................................................. 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Case Study 1: Cell Image Segmentation</td>
<td>59</td>
</tr>
<tr>
<td>(in unsupervised mode).</td>
<td></td>
</tr>
<tr>
<td>7.2 Case Study 2: Traffic Surveillance</td>
<td>63</td>
</tr>
<tr>
<td>Dataset (in supervised mode).</td>
<td></td>
</tr>
<tr>
<td>8 3D Data Segmentation</td>
<td>67</td>
</tr>
<tr>
<td>8.1 Reflectance Confocal Microscopy (RCM)</td>
<td>67</td>
</tr>
<tr>
<td>8.1.1 Dermal-Epidermal Junction of Human</td>
<td>69</td>
</tr>
<tr>
<td>Skin</td>
<td></td>
</tr>
<tr>
<td>8.2 Literature Review</td>
<td>70</td>
</tr>
<tr>
<td>8.3 Model Formulation</td>
<td>72</td>
</tr>
<tr>
<td>8.3.1 Spatial Poisson Process</td>
<td>73</td>
</tr>
<tr>
<td>8.3.2 Shape Model</td>
<td>75</td>
</tr>
<tr>
<td>8.3.3 Appearance Model</td>
<td>78</td>
</tr>
<tr>
<td>8.4 Experiments and Results</td>
<td>80</td>
</tr>
<tr>
<td>9 Time-Series Data</td>
<td>87</td>
</tr>
<tr>
<td>9.1 Introduction</td>
<td>87</td>
</tr>
<tr>
<td>9.2 Model Formulation</td>
<td>89</td>
</tr>
<tr>
<td>9.2.1 Complex Shape Model</td>
<td>89</td>
</tr>
<tr>
<td>9.3 Inference</td>
<td>90</td>
</tr>
<tr>
<td>9.4 Experiments</td>
<td>91</td>
</tr>
<tr>
<td>10 VOTERS Project</td>
<td>95</td>
</tr>
<tr>
<td>10.1 Camera System (SVAD)</td>
<td>96</td>
</tr>
<tr>
<td>10.2 Literature Review</td>
<td>98</td>
</tr>
<tr>
<td>10.3 Algorithms</td>
<td>100</td>
</tr>
</tbody>
</table>
10.3.1 Multi-Class Logistic Classifier ................................................. 102
10.3.2 Multi-scale Hessian Based Detection ........................................ 109
10.3.3 Alligator Area Detection .......................................................... 112
10.3.4 Longitudinal and Transverse Crack Detection .............................. 113
10.4 Experiments ................................................................................. 115
  10.4.1 Determination of Optimal Parameters ...................................... 115
  10.4.2 Results ...................................................................................... 118
10.5 Conclusion .................................................................................... 122

11 Conclusion ...................................................................................... 127
  11.1 Future Work .................................................................................. 129
List of Figures

3.1 Nearest object occurrence from a random point. ........................................... 21

4.1 A graphical representation of the proposed model. ................................. 31

4.2 Outcome of a homogeneous Poisson process. ........................................... 32

4.3 Outcome of a non-homogeneous Poisson process. ..................................... 32

4.4 (a) Shape parameters of an ellipse. (b) Landmark points for shape prior of pedestrians. ......................................................................................... 33

4.5 (a) Shapes extracted from training images. (b) Synthetic images generated from the inferred shape prior. ................................................................. 35

5.1 A graphical representation of the proposed model. .................................... 42

5.2 (a) Location Set-1 and (b) Location Set-2 where the image is divided into 16 regions. (c) and (d) show the division of space into 4 regions. Value of Poisson log-likelihood for each configuration is reported below the corresponding figure. . . . 43

5.3 (a) Location Set-1 (b) Location Set-2 with $M = 4$ and (c), (d) show the same set with $M = 9$. Value of Poisson log-likelihood for each configuration is reported below the corresponding figure. ...................................................... 43

5.4 An example of inference of Poisson intensity given a set of observations. .......... 51
6.1 Results of our algorithm on synthetic images. Figure (a), (c) and (e) show results from the proposed model with spatial prior. Figure (b), (d) and (f) show results from the proposed model without spatial Poisson prior. 55

6.2 Results of our algorithm on synthetic images. 56

6.3 Templates used for training shape and appearance of maple leaves. 57

6.4 Results of our algorithm on dataset consisting of maple leaves with a different appearance. 58

7.1 Results of algorithms on a sample image. 61

7.2 Results of algorithms on a sample image. 62

7.3 Results on traffic images. (a) and (b) show example detection/segmentation results for pedestrians and cars. 64

7.4 Results on traffic images. (a) shows the plots of detection accuracy compared with other methods and (b) shows the contour plot of posterior Poisson intensity value inferred by the algorithm. 65

8.1 Histology images from vertical sections of skin. 68

8.2 Acquisition of slices using RCM. 69

8.3 Dermal-Epidermal junction of skin and RCM slices. 69

8.4 Graphical representation of the generative model. 74

8.5 Example 3D shape models. 74

8.6 Intensity along the z-axis varies according to $\Gamma(10,3)$. Uniform intensity prior across the x-y plane is assumed with the integral Poisson intensity value $\sum_{r} \lambda = 50$. 75
8.7 Dermal-Epidermal junction detected by our algorithm for dark skin type. Color indicates the error in DEJ detection compares to expert labels. 83

8.8 Dermal-Epidermal junction detected by our algorithm for fair skin type. Color indicates the error in DEJ detection compares to expert labels. 83

8.9 Dermis en-face segmentation results. 84

9.1 Two sample sub-sequences. 88

9.2 A part of the test sequence is plotted. Red indicates the sub-sequence of interest and blue indicates background. 93

9.3 Plots showing 3 classes of dataset ‘CBF’. 93

10.1 Distortion correction from calibration image 102

10.2 Figure 4: (a) Longitudinal crack template (b) Diagonal crack template of type 1 (c) Transverse crack template (d) Diagonal crack template of type 2 108

10.3 Results from Hessian decomposition per pixel on example image. 111

10.4 Results from different thresholds on Eigenvalues at different scales. 111

10.5 Analysis of scale parameter variation for crack detection. 117

10.6 Analysis of parameter variation for alligator area detection. 118

10.7 Example results of the algorithm. Green indicates alligator cracking, blue indicates longitudinal cracking, violet indicates transversal cracking and red indicates other random cracking. (a) and (e) are the original images, (b) and (f) are groundtruth labels, (c) and (g) are results of the ‘HMF’ algorithm, (d) and (h) show results of ‘MLR’. 119

10.8 Results of alligator on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset. 124
10.9 Results of longitudinal cracking on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset.

10.10 Results of transversal cracking on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset.
List of Tables

7.1 Segmentation results for cell data ................................................. 61
7.2 Detection results for cell data ....................................................... 62

8.1 Evaluation of the Performance of the Proposed Algorithm for Dark Skin Type.................................................. 85
8.2 Evaluation of the Performance of the Proposed Algorithm for Fair Skin Type .................................................. 85

9.1 Overall Precision Recall values for sub-sequence matching ................................................................. 92
9.2 Precision Recall values for sub-sequence matching of each class .................................................. 92

10.1 Specification for acquisition software of Camera ................................................................. 97
10.2 Pixel-wise segmentation accuracy and average segmentation accuracy values reported by algorithm with respect to ground-truth ................................................................. 121
10.3 Alligator area segmentation accuracy and average segmentation accuracy values reported by algorithm with respect to ground-truth ................................................................. 122
10.4 Area of alligator cracking reported by algorithm and groundtruth data in $m^2$ ................................................................. 122
10.5 Mean Square error in area of alligator cracking reported by ‘HMF’ and ‘MLR’ ($m^2$) ................................................................. 122
10.6 Length(m) and width(cm) of longitudinal cracks reported by algorithm and groundtruth data ................................................................. 123
10.7 Mean Square error in length of longitudinal cracking reported by ‘HMF’ and ‘MLR’ \( (m) \). \hspace{1cm} 123

10.8 Mean Square error in length of transversal cracking reported by ‘HMF’ and ‘MLR’ \( (m) \). \hspace{1cm} 124
Notation

 Scalars are represented by small letters, vectors by bold small letter and matrices are capitalized bold type. We do not distinguish between probabilities and probability densities.

\[ p(\cdot) \] probability density/mass or simply probability of the random variable
\[ \Pi \] A Poisson process
\[ \nu \] Measure of the Poisson process
\[ X \] Image matrix
\[ \lambda \] Poisson intensity parameter
\[ S \] Shape parameters of the model
\[ \zeta \] Shape hyper-parameters
\[ \alpha, \gamma \] Appearance parameters
\[ Z \] Indicator variable
\[ D \] Size of observations
\[ N \] Number of objects
Chapter 1

Introduction

We live in the era of information revolution where a large amount of information is generated every day in the form of images, documents, speech, etc. Automatic processing of this data to extract relevant information of interest has become a must. For example, one might search for cars and pedestrians in a traffic surveillance video, or companies with similar trends in the stock market, or objects of interest such as cells or tumors in medical images. Automatic processing is objective and leads to large savings in terms of expensive man hours. The processing that happens behind most such applications is a result of developments in Machine Learning literature. Very often, the data to be processed is the result of complex underlying processes which the machine learning algorithms try to imitate.

This dissertation explores the problem of simultaneously detecting and segmenting multiple occurrences of objects of interest in the image, inspired from developments in machine learning, statistics and computer vision literature. These objects could vary in shape and appearance. We provide a formulation that can be readily extensible to any number of dimensions: 1D, 2D, 3D. The problem of learning multiple occurrences of objects in an image is made more challenging
by the need to deal with the uncertainty associated with the number of occurrences of the object. Before delving into the details, a brief outline of the related literature is presented in the following paragraph. A detailed literature review is provided in Chapter 3.

This dissertation presents a nonparametric Bayesian model, and we start with an intuitive explanation of what it is. A statistical model explains the probability measures on a sample space $X$. One could use a set of parameters $\theta$ combined with a function to map each parameter point with a probability distribution $f : \theta \rightarrow p(X)$, where $f$ is injective. A model is called Bayesian if there is a component of the prior distribution on the parameter $\theta$ [102]. A Bayesian model is called non-parametric when there is an infinite dimensional parametric space. However, only a finite subset of the available parameter space is used based on the data being modeled. The process of determining the parameters that best explain the data using the assumed model is called inference. Bayesian nonparametric models have become popular in Machine Learning owing to their flexibility and adaptability with respect to the number of parameters. For example, a Bayesian non-parametric model can determine the number of clusters in the data along with their respective statistics [113]. This overcomes the problem of model selection, and the number of parameters is determined adaptively based on data. Inference on such models is not trivial and is an area of active research. Indian Buffet process (IBP) and Chinese Restaurant process (CRP) are a few examples of non-parametric Bayesian models on which efficient inference strategies have been designed and implemented successfully.

This paragraph briefly explains a Poisson process which is an integral part of the proposed model in this dissertation. Spatial point processes are statistical models that are used to study the patterns of points where their location represents the location of an object being studied [12]. The Poisson process is a special case of point processes where the occurrence of a point depends on
only one parameter, which is called the Poisson intensity. The variation in the intensity over a 2D plane is an indicator of the probability of occurrence of a point in it.

In this dissertation, we consider the elaboration of Poisson processes to *marked Poisson processes*, remaining within the context of Bayesian nonparametric modeling. Our motivating examples come from image processing and computer vision, where we often wish to infer the number of objects (of each type) in a scene, and where the shape of the object can be viewed as a “mark.” This “mark” is not fixed and is described by a shape prior model whose parameters also need to be inferred. This makes the “mark” a latent variable of the model and hence the model can be called a Latent Marked Poisson Process. Our formulation of the model can handle a variable number of points/occurrences at random locations with latent marks for a given data $X$.

**Thesis statement:** “A latent marked Poisson process is a Bayesian non-parametric model, that provides a natural framework to incorporate uncertainty in number, location, shape and appearance of flexible patterns in 1D, 2D and 3D data.”

### 1.1 Contributions

Building on developments from statistics, machine learning, image processing and computer vision, the contributions of this work are:

- We provide a nonparametric Bayesian probabilistic model for multiple objects that takes uncertainty in number, location, shape and appearance into account. Our formulation is quite flexible in that; it can incorporate any probabilistic spatial, shape and appearance prior. It can also detect more than one category in the image simultaneously and allow objects to
overlap.

- We show how to use a reversible jump Markov chain Monte-Carlo (RJMCMC) for inference and introduce a novel and more efficient, as compared to an RJMCMC, hybrid Gibbs sampler for a marked Poisson process.

- Our spatial Poisson process can have a homogeneous or a non-homogeneous intensity parameter. Utilizing a non-homogeneous intensity parameter enables incorporating spatial information to our model. The ability to provide spatial information is important because the probability of occurrence of objects in a scene depends on location (e.g., objects like cars will be found only on roads and the probability of finding them on a sidewalk or on a building is small). Learning the non-homogeneous intensity can show a probability map reflecting regions of where high (traffic) or low object concentration occur.

- Furthermore, we demonstrate our model on synthetic and real-world applications. We evaluate our algorithm in fully unsupervised mode (for the cell data), dermis-epidermis segmentation in reflectance confocal microscopy images (RCM) and in supervised mode for traffic surveillance data. We finally extend this model to subsequence matching in time-series data.

- We have formulated a multi-class logistic classifier for crack detection in images of road surfaces. Inspired from techniques in biomedical image processing, we propose to use a Hessian-based multi-scale filter to detect ridges in the form of cracks, at different scales. We present a thorough analysis and a comparison between the algorithms.

- We make available a public dataset consisting of pavement images, with groundtruth labels for cracks and their types. This will enable quantification of our results and also helps future
researchers to develop and improve automated detection of road surface defects from images.

1.2 Thesis Organization

The rest of this dissertation is organized as follows. In Chapter 2, we provide a literature review. In Chapter 3, we present the background theory on which our model is based. In Chapter 4, we describe our proposed model. In Chapter 5, we derive the inference steps necessary to learn the parameters and latent variables of our model. In Chapter 6, we present the possible extensions to our proposed model. In Chapters 7, 8, and 9, we report and discuss empirical results on 2D, 3D and 1D data respectively. In Chapter 10, we describe our work under the VOTERS project for crack detection in surface images. Finally, we provide conclusions and possible future directions for this work in Chapter 11.
Chapter 2

Review of Related Work

This chapter reviews the literature in statistics, machine learning and computer vision with respect to the proposed algorithms and the formulation of our nonparametric Bayesian marked Poisson process model introduced in this dissertation. A review of literature associated with a specific experiment/application of the model is presented in the chapter in which the application is described (e.g., the related work for dermal-epidermal junction detection in reflectance confocal microscopy is provided in Chapter 8). Details of key concepts necessary to understand the work in this dissertation are presented in Chapter 3.

2.1 Bayesian Nonparametrics

Bayesian style inference on a nonparametric model is made difficult by the fact that one has to define a prior distribution over possibly infinite variables. Dirichlet process was first proposed by [52] as a workable prior distribution for non-parametric formulations. Dirichlet process mixture model offers a natural mechanism to formulate flexible clustering algorithms while staying within
the non-parametric Bayesian framework. This has been adapted and extended for different applications [142, 141]. The Chinese Restaurant Process (CRP) is a popular closely related sequential process [119] based on Polya urn [20] scheme useful in clustering applications [22, 137]. Beta process was introduced by [72] as a much larger and flexible class of nonparametric prior distribution. It was shown that Dirichlet Processes can be transformed in a particular way to produce a special case of beta processes [72]. The Indian buffet process (IBP) [65] is a closely related process [143] that places a prior over infinite binary sparse matrices and is considered as a generalization of CRP. IBP is useful for defining nonparametric latent feature models [30, 153, 79]. A draw from a beta process consists of an infinite collection of weights that lie in the interval [0, 1]. A stick breaking formulation to construct of beta process was shown in [115]. The stick-breaking construction samples sequentially from a beta process, by breaking a stick of unit length into progressively smaller parts. Such a construction paved way to novel and simple sampling strategies for inference. The work in [116] shows that the stick-breaking construction of beta process can be obtained, from the characterization of the beta process as a Poisson process.

In images, one is often interested in segmenting and detecting unknown number of objects. Bayesian non-parametric models provide priors that enable automatically determining the order of the model. Methods that use Dirichlet process mixtures and hierarchical beta processes provide a prior distribution on partitions and binary features (respectively), and do not limit the cardinality of the representation. Bayesian non-parametrics have been used in the literature to solve vision problems [71]. Spatial smoothness associated with images is captured by coupling Markov random fields [58] with Dirichlet process for segmentation problems in [112]. Distance-dependent Chinese restaurant process [21] allows dependencies between samples and a logistic stick-breaking process
CHAPTER 2. REVIEW OF RELATED WORK

[127] takes spatial and temporal dependence into account. However, these models do not incorporate shape information. In contrast, this thesis introduces a model that unifies multiple object detection and segmentation with shape and appearance priors, within the Bayesian nonparametric framework.

There does not exist a unified model in Bayesian nonparametrics; that can incorporate a flexible shape information along with uncertainty in the number of objects, their location, and appearance. Even though a Poisson process underlies a wide range of nonparametric stochastic processes including Gamma, beta and Dirichlet processes [85], it is a less explored area in the context of solving segmentation problems in images. In the following sections, a review of the literature on spatial Poisson processes is presented.

2.2 Poisson Processes

Poisson processes have been widely studied in the statistics literature as an example of Levy processes [17]. Spatial Poisson processes and their properties were studied extensively in [85, 12]. Most of the previous literature has focused on inference of the Poisson intensity parameter given an outcome of the Poisson process [106, 82, 44, 14, 125]. Quadrat, distance, and kernel methods are the popular deterministic methods used for inference [38]. Using another stochastic process to model the underlying intensity was suggested in [36], as the underlying true functional form is unknown. It is popularly known as the Cox process. Utilizing a nonparametric formulation to model Poisson intensity parameter has several advantages. Such a formulation is very flexible and can represent all possible spatial patterns [8, 88, 150].

We propose a novel way to model natural images, using a generative probabilistic model, based
on a spatial Poisson process prior. In our model, a spatial Poisson process is used to model the unknown number of objects along with their location. In addition, a spatial Poisson process allows us to incorporate a context prior informing the model of high or low spatial concentration areas of object occurrence. Context in our case refers to the probability of finding an object based on the absolute location of the object in the image plane, unlike other models, where pairwise geometric relationship between different objects is formulated as context [70]. Then, a flexible shape model is associated with each location of the object. Once the number, location and shape of each object is known, appearance of pixels inside and outside the object is captured by an appearance model.

When the points of a Poisson process are marked with additional information, it is called a Marked Poisson Process (MPP). A Marked Poisson process (MPP) is a special case of marked point processes. There exists literature on marked point processes for detection of objects [42, 13]. However, these work only show examples on simple shapes and synthetic images. An attempt to use the marked point process in real world scenarios was made in [94, 129, 15]. Combining contour detection framework with marked point process to detect arbitrarily shaped objects was proposed in [89]. These methods discourage object overlap between objects, either using a hardcore process or a Strauss process. In contrast to these approaches, our model utilizes a marked Poisson process with no constraints on overlap between objects. Context information in real data is captured naturally by a spatial Poisson process. Often, one has domain knowledge about regions of the image that have a high/low probability of occurrence of an object. For example, in traffic surveillance images, cars will be found only on roads and their probability of occurrence in other regions of the image is very low. This information is captured naturally by the Poisson intensity parameter.

A typical problem associated with utilizing a Poisson process for object detection is its changing model order. Model order change is a natural outcome of using a Poisson process as each
CHAPTER 2. REVIEW OF RELATED WORK

random outcome from it can have different number of objects. This change in model order makes the problem of inference hard. Typically, reversible-jump Markov chain Monte Carlo (RJMCMC) sampling strategy is used to ameliorate this problem [56]. Our work, on the other-hand, introduces a novel inference strategy based on hybrid Gibbs sampler. A detailed explanation of non-homogeneous Poisson process is presented in Chapter 3.1.

2.3 Object Detection and Segmentation

Computer vision has a rich literature on segmentation algorithms that include several variations and combinations of clustering, graph cuts and level sets. Shape is a powerful descriptor of objects and segmentation performance is shown to improve by including complex shape, in addition to appearance in many computer vision applications [48, 77, 132, 26, 33, 76]. The incorporation of shape priors is also popular in level-set methods and formulations that optimize an energy functional [26, 37, 75]. There have also been proposals that incorporate shape priors in graph-based segmentation algorithms [90]. The original formulations are designed to discover one object (for example, segmentation of the heart in a computed tomography image). However, one may be interested in detecting multiple occurrences of similar objects or patterns in an image (for example, cells in an image) or in segmentation of multiple objects/patterns in an image that may overlap [144, 145]. A discriminative model that would incorporate shape, appearance and contextual information using conditional random fields was proposed in [132]. In contrast, our model uses a spatial Poisson process to model the unknown number of objects along with their location. Context is captured by the spatial Poisson process in terms of absolute location of the object, unlike other models, where pairwise geometric relationship between different objects is formulated as context [70].
Probabilistic models have been used for image segmentation, including the pioneering works of [58, 19] on Markov random fields applied to image segmentation [96]. A review of probabilistic models to level-set segmentation is provided in [37], where inference is based on maximum a posteriori (MAP) estimation leading to an energy minimization optimization. Similar to our work, these approaches integrate shape and appearance; however, contrary to our work, they do not incorporate a spatial prior.

2.4 Inference

Often, inference of the posterior distribution of unknown variables in the model is intractable and one resorts to approximate methods. This is especially true in the Bayesian setting when conjugate priors are not used. This approximation can be accomplished by resorting to sampling techniques based on Metropolis-Hastings algorithm [45] or analytical approximations popularly known as variational methods [46].

Metropolis-Hastings algorithm provides a framework to simulate samples from a probability distribution (posterior) of the variable of interest by making use of the joint distribution of the model and independent proposal distributions. This class of methods is popularly known as Markov Chain Monte Carlo (MCMC). Gibbs sampling is a special case of Metropolis-Hastings where every sample proposed is accepted. This method requires a known conditional distribution of the variable being sampled [109]. However, none of these sampling methods can handle change in dimensionality of the parameters. To overcome this limitation, Reversible Jump Markov Chain Monte Carlo (RJMCMC) was proposed by [63]. It has the advantage of being able to handle non-conjugate priors, non-exponential distributions and additional variables that might be introduced.
into the model during the process of sampling.

Inferring the occurrence of multiple unknown number of objects from different categories, having variations in shape and appearance, using a Bayesian approach, is challenging. Reversible Jump Markov Chain Monte Carlo (RJMCMC) offers a natural framework to overcome this problem [63]. However, RJMCMC requires choosing appropriate proposal distributions to ensure a good acceptance ratio (at least 10% acceptance is desirable). The total amount of time it takes for an algorithm to converge depends on the ability to choose an appropriate proposal distribution. To overcome this limitation, we design a more efficient Gibbs sampler for inference. A Gibbs sampler cannot deal with trans-dimensional jumps; hence, we propose an approximation to our formulation of the model which will allow Gibbs sampling and at the same time, provide a posterior distribution on a variable number of objects in the image.
Chapter 3

Background

In this chapter, a review of the concepts of Poisson processes, Bayesian nonparametrics and Bayesian inference using sampling techniques is presented.

3.1 Poisson Process

A Poisson process in general is used to model point data and is a special case of point processes. In a spatial setting, one realization of the Poisson process consists of a random number of events at random locations in the 2D plane. We define a point process, based upon which we present the formal definition of Poisson process as a special case of the point process.

Let there be a Euclidean d-space $\mathbb{S}$, where countable number of points are observed at random locations. Let $\{x_1, \ldots, x_k\}$ be the elements in the space $\mathbb{S}$ and $\mathcal{B}$ be the Borel $\sigma$-field on $\mathbb{S}$. The number of points $x_i$ belonging to $B \in \mathcal{B}$ is given by Equation 3.1 [126].

$$\left| \{x_i \in \mathbb{S} : i \in I \} \cap B \right| = \sum_{i \in I} 1_B(x_i) = \sum_{i \in I} \epsilon_{x_i}(B),$$  \hspace{1cm} (3.1)
where, \( \epsilon_i(B) := 1_B(x_i), B \in \mathcal{B} \), is the Dirac measure with mass 1 at \( x \).

A discrete measure \( \mu \) in Equation 3.2 is a representation of the observed points \( x_i, i \in I \). Let the space of point measures on \( \mathcal{B} \) be denoted by \( \mathcal{M} \equiv \mathcal{M}(S, \mathcal{B}) \). A point measure without identical points (multiple points of the same value are not observed) is called \textit{simple}. Using these definitions, one can define an atom.

\[
\mu = \sum_{i \in I} \epsilon_{x_i}
\]  
\[(3.2)\]

**Atom:** Given a measurable space \((\Omega, \mathcal{A})\) and a measure \( \mu \) on that space, a set \( A \in \mathcal{A} \) is an atom if:

- \( \mu(A) > 0 \);
- For every \( V \in \mathcal{A} \) with \( V \subseteq A \), one of \( V, A \setminus V \) (\( A \) not including \( V \)) is negligible.

In other words, an atom is the smallest measurable set in \( \mathcal{A} \), on which the measure is positive.

Let \( \mathbb{N} \) denote positive integers and \( \mathbb{N} = \mathbb{N} \cup \{0\} \cup \{\infty\} \). Consider the projection \( \pi_B : \mathcal{M} \rightarrow \mathbb{N} \). The smallest projection such that \( \pi_B \) is measurable, is represented by \( \mathcal{M} \equiv \mathcal{M}(S, \mathcal{B}) \). Based on these definitions, one can define a point process.

**Point process:** Let \((\Omega, \mathcal{A}, P)\) be the probability space on which the random variables are defined. A mapping

\[
N : \Omega \rightarrow \mathcal{M}(S, \mathcal{B})
\]  
\[(3.3)\]

is called a \textit{point process} on \((S, \mathcal{B})\) if it is measurable with respect to \( \mathcal{A} \) and \( \mathcal{M}(S, \mathcal{B}) \).
The intensity measure $\nu|\mathcal{B}$ of a point process $N$ is called a mean measure of the point process (expected number of points in the set $\mathcal{B}$) and is defined by Equation 3.4 and is itself a measure.

$$\nu(\mathcal{B}) = E[N(\mathcal{B})], \mathcal{B} \in \mathcal{B}$$  \hspace{1cm} (3.4)

A point process $N : \Omega \rightarrow \mathcal{M}(\mathcal{S}, \mathcal{B})$ is a Poisson process if the following two conditions are satisfied:

- $p(N(\mathcal{B})) = P_{\nu(\mathcal{B})}$, for every $\mathcal{B} \in \mathcal{B}$ and

- $N(\mathcal{B}_1), \ldots, N(\mathcal{B}_k)$ are independent for every $k \in \mathbb{N}$ pairwise disjoint sets $\mathcal{B}_1, \ldots, \mathcal{B}_k \in \mathcal{B}$.

where, $P_{\nu(\mathcal{B})}$ is the Poisson distribution with parameter $\nu(\mathcal{B}) \geq 0$ given by Equation 3.5.

$$P_{\nu(\mathcal{B})}(k) = \frac{[\nu(\mathcal{B})]^k}{k!} e^{-\nu(\mathcal{B})}, k \in \mathbb{N}_0$$  \hspace{1cm} (3.5)

where, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Here, $\nu$ can be defined in terms of the measurable function $\lambda$ on $\mathcal{S}$ by Equation 3.6.

$$\nu(\mathcal{B}) = \int_{\mathcal{B}} \lambda(x)dx$$  \hspace{1cm} (3.6)

where, $\lambda$ is called the rate when we deal with one-dimensional space and intensity when $S$ has a spatial interpretation.

As is evident from the above description, the probability of occurrence of a point at any location depends on only one parameter, the Poisson intensity $\lambda$. This parameter can be a constant value in which case the Poisson process is called \textit{homogeneous}. In other words, there is uniform probability of a point to occur at any location. When the intensity parameter varies along the space $S$, it is called a \textit{non-homogeneous} Poisson process.
3.1.1 Properties of a Poisson Process

In this part of the Chapter, we describe a few interesting properties of the Poisson process such as the superposition property and distance statistics. This helps understand the beauty and uniqueness of the Poisson process. Some of the properties come in handy when we formulate a Gibbs sampling strategy for inference of our proposed model. Let $\Pi$ be a Poisson process.

**Superposition Theorem:** Let $\Pi_1, \Pi_2, \ldots$ be a countable collection of independent Poisson processes on $S$ and let $\Pi_n$ have mean measure $\nu_n$ for each $n$. Then, their superposition

$$\Pi = \bigcup_{n=1}^{\infty} \Pi_n$$

is a Poisson process with mean measure

$$\mu = \sum_{n=1}^{\infty} \mu_n$$

Details of the proof can be found in [85]. Intuitively, this theorem subsumes the additive property of the Poisson process, which states that sum of two independent Poisson variables with underlying intensity $\lambda_1$ and $\lambda_2$ follows a Poisson distribution with intensity $\lambda_1 + \lambda_2$. Another important property of the Poisson process is that when its state space is mapped into another space, the transformed random points also form a Poisson process. This is called the mapping theorem.

**Mapping Theorem:** Let $\Pi$ be a Poisson process with $\sigma$ finite mean measure $\nu$ on the state space $S$, and let $f : S \to T$ be a measurable function such that the induced measure has no atoms. Then $f(\Pi)$ is a Poisson process on $T$ having the induced measure $\nu^*$ as its mean measure.
A measure is called $\sigma$ finite when the set on which it is defined is the countable union of measurable sets with finite measure. The proof of mapping theorem can be found in [85]. We now proceed to show that a Bernoulli process can be derived from the Poisson process. The total number of points $N(S)$ from the Poisson process $\Pi$ have a Poisson distribution $P(\nu(S))$. Let the conditional distribution of $\Pi$ conditioned on the value of $N(S)$ be denoted by $P(\nu(S))$. Let $\mathcal{B}_1, \ldots, \mathcal{B}_k$ be disjoint subsets of $S$. Let $n = N(S)$ be the number of points and $n_0 = n - \sum^k_{j=1} n_j$ and $\mathcal{B}_0$ be the complement of $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_k$. Then,

$$
\mathbb{P}_n \{N(\mathcal{B}_1) = n_1, N(\mathcal{B}_2) = n_2, \ldots, N(\mathcal{B}_k) = n_k\}
$$

$$
= \mathbb{P}_n \{N(\mathcal{B}_0) = n_0, N(\mathcal{B}_1) = n_1, N(\mathcal{B}_2) = n_2, \ldots, N(\mathcal{B}_k) = n_k\} | P\{N(S) = n\}
$$

$$
= \frac{n!}{n_0! n_1! \ldots n_k!} \left(\frac{\nu(\mathcal{B}_0)}{\nu(S)}\right)^{n_0} \left(\frac{\nu(\mathcal{B}_1)}{\nu(S)}\right)^{n_1} \ldots \left(\frac{\nu(\mathcal{B}_k)}{\nu(S)}\right)^{n_k}
$$

$$
= \frac{n!}{n_0! n_1! \ldots n_k!} p(\mathcal{B}_0)^{n_0} p(\mathcal{B}_1)^{n_1} \ldots p(\mathcal{B}_k)^{n_k}
$$

is a Bernoulli process with mean measure $\mathbb{E}\{N(\mathcal{B})\} = np(\mathcal{B})$. Therefore, conditioning on $N(S) = n$ converts the Poisson process with finite mean measure $\nu$ into a Bernoulli process with parameters $n$ and $p(\cdot) = \nu(\cdot) / \nu(S)$. The existence theorem proves that for a given mean measure, there exists a Poisson process. We present the statement of the theorem and refer the reader to [85] for a detailed proof.

**Existence Theorem:** Let $\nu$ be a non-atomic measure on $S$ which can be expressed in the form

$$
\nu = \sum_{\mu=1}^{\infty} \nu_\mu, \quad \nu_\mu(S) < \infty,
$$

$$
(3.11)
$$
Then there exists a Poisson process on $S$ having $\nu$ as its mean measure. Imagine marking the outcome of a Poisson process with a feature that determines its type. For example, the positions of trees in a forest correspond to the outcome of a Poisson process and the marks are their heights. Care should be taken to formulate this idea precisely. Let $m_x$ be the random variable we associate with each point $X$ of the random set $\Pi$ taking values in some space $M$. The values of $m_x$ for different $X$ are independent and may depend on $X$. The pair $(X, m_x)$ can be regarded as a random point $X^*$ in the product space $S \times M$ forming a random countable subset $\Pi^* = \{(X, m_x); X \in \Pi\}$. As a result $\Pi^*$ is a Poisson process on the product space $S \times M$. Below is the formal definition of the marking theorem which can be found along with its proof in [85].

**Marking Theorem:** The random subset $\Pi^*$ is a Poisson process on $S \times M$ with mean measure $\nu^*$ given by

$$
\nu^*(C) = \int \int \nu(dx)p(x, dm).
$$

(3.12)

### 3.1.2 Inference on Intensity Parameter of Poisson Process

In this part of the chapter, different methods traditionally used for estimation of the intensity parameter in a Poisson process are presented. A review of quadrat methods, distance based methods, kernel estimators and cox processes is presented.

Estimation of the intensity parameter for a homogeneous Poisson process involves estimating a single positive number called the rate of occurrence. Estimating the intensity for a non-homogeneous process is challenging as one needs to determine the functional form of the parameter too.
**Quadrat sampling** is a technique used for determination of the number of occurrences of the desired object in the study region $\mathbb{B}$. Traditionally, these subsets are rectangular. Ideally, the frequency of occurrence of points in these quadrats should have a Poisson distribution. There is an extensive literature on selection of quadrat size with overlapping and non-overlapping quadrats. Let $z$ be a node of area $[a \times a]$. When the intensity does not vary much in the area $[a \times a]$, it can be represented by $\lambda(z)$. A set of quadrat counts per unit area ($N(z)/a^2$) estimates the intensity $\{\lambda(z)\}$. Here, let $N(\cdot)$ indicate the number of objects in a given area. This technique typically estimates the homogeneous Poisson intensity parameter.

**Distance methods** that measure the distance of closest occurrence of points from a given set of fixed points are another way to determine the intensity value. These methods take advantage of the complete spatial randomness property of a Poisson process. A random outcome from a Poisson process follows a certain distribution based on the underlying intensity and this information can be used in intensity estimation of the Poisson process. This method is demonstrated with an example to make it clear. Consider an image with $D$ pixels and uniform Poisson intensity parameter of $\lambda$ per unit area. Consider a point in the image, from which the occurrence of the nearest object is at a distance of $r$ as shown in Figure 3.1. The probability of there being no objects within the circle of radius $r$ is given by $\exp(-\pi r \lambda dr)$. The probability of occurrence of a point within $dr$ is given by $2\pi r \lambda dr$. These are independent events and the probability density function of the nearest object is given by $p(r) = 2\pi r \lambda \exp(-\pi r^2 \lambda))$. This is a well-known result whose details can be found in [134].

Now, assume a grid of $M$ points evenly distributed on the surface of the image. Let $r_m$ denote the distance from the $m^{th}$ grid point to the nearest object center in the image. The likelihood of sample distances $r_1, r_2, \ldots r_M$ is given by Equation 5.9 [120].
\[ p(R) = \prod_{m=1}^{M} \sum_{d=1}^{D} (2\pi r_m)(p_d \lambda_d) \exp(-\pi r_m^2 \lambda_d) = (2\pi I)^M \exp(-\pi I \sum_{m=1}^{M} r_m^2)(r_1 r_2 \ldots r_M) \]  

(3.13)

where, \( R = [r_1, \ldots, r_M] \) and \( p_d \) is the proportion of image with intensity \( \lambda_d \) and \( I = \sum_{d=1}^{D} \lambda_d \).

Figure 3.1: Nearest object occurrence from a random point.

The quadrat and distance methods described are parametric and are typically used to get a point estimate of the homogeneous Poisson intensity parameter. Kernel density estimation, on the other hand, is a nonparametric estimator of the Poisson intensity given a set of observations. Details of this method are presented in the following paragraph.

*Kernel density estimation* is a nonparametric technique used for determining the intensity parameter given a set of object locations as observations. Let \((x_1, x_2, \ldots, x_n)\) be the spatial locations of \( n = N(\mathcal{B}) \) in a bounded study region \( \mathcal{B} \). Consider an estimator \( \hat{\lambda}_h(z) \) of the form,

\[
\hat{\lambda}_h(z) \equiv \frac{1}{p_h(z)} \sum_{i=1}^{n} \kappa_h(z - z_i), \quad z \in \mathcal{B},
\]  

(3.14)

where, \( \kappa_h(\cdot) \) is a probability density (kernel) function symmetric about the origin, \( h > 0 \) determines the amount of smoothing and \( p_h(z) \equiv \int_{\mathcal{B}} \kappa_h \cdot (z - u)du \) is an edge correction [43, 38].

All the above methods give a point estimate of the Poisson intensity parameter. A Cox process on the hand provides a formulation to estimate the intensity in Bayesian setting. Details of this
CHAPTER 3. BACKGROUND

process are provided in the following paragraphs.

**Log-Gaussian Cox Process:** In a non-homogenous Poisson process, determining the value of \( \lambda \) as a function of location is challenging and the above methods (quadrat, kernel and distance methods) do not address this problem. Moreover, these methods have been used in maximum likelihood estimation, and the log Gaussian Cox process is inferred using Bayesian inference. Using another stochastic process to model the underlying intensity is as the underlying true functional form is unknown was proposed in [36]. There are many ways in which this can be modeled. A non-parametric modeling of intensity parameter will be very flexible and can represent all possible spatial patterns [8, 88, 150]. A log-Gaussian cox process is described here. A log-Gaussian Cox process assumes intensity function as the exponential of a random realization from the Gaussian process given in Equation 3.15, where \( g(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R} \) be a random scalar function having a Gaussian process prior.

\[
\lambda(u, v) = \exp(g(u, v)) \tag{3.15}
\]

\((u, v)\) is a continuous random variable that represents the location in the image). This implies that any discrete realization of the function will be a multivariate Gaussian distribution. This model makes no numerical assumptions, assumes no knowledge about the upper limit on the intensity value and does not suffer from boundary problems. A log-gaussian cox process can estimate the intensity values beyond the borders of the 2D plane, in cases where the camera might zoom out [105].
3.2 Poisson Process for Bayesian Nonparametrics

Nonparametric Bayesian models refer to formulations that have priors on an infinite dimensional parametric space [108]. Such probability distributions on infinite dimensional random variables are called stochastic processes [114]. Some examples include Gaussian processes for regression, Dirichlet processes for clustering and beta processes for feature selection.

A Poisson process underlies a wide range of stochastic processes including Gamma, beta and Dirichlet processes [85]. It is shown that the beta process can be characterized as a Poisson process in [116]. A beta process is an infinite collection of weighted atoms in the space \( \Omega \), where the weights lie in the interval \([0, 1]\). It consists of two parameters \((\alpha, \mu)\). A beta process is useful in providing parameters of the Bernoulli process. It is shown that a Poisson process underlies the construction of a Beta process in [81]. A brief overview is presented here. Consider a product space \( \Omega \times \mathbb{R} \) with a Poisson rate measure \( \lambda \). Consider defining this rate measure in terms of the product of an arbitrary sigma-finite measure \( B_0 \) on \( \Omega \) and an improper beta distribution on \((0, 1)\). An improper distribution refers to a distribution that integrates to infinity (improper priors were introduced in [80]). When the Poisson rate measure is defined in this way (Equation 3.16), a countably infinite number of random points are obtained as a draw from the Poisson process. The resulting completely random measure is called a beta process.

\[
\lambda(d\omega, dp) = cp^{-1}(1 - p)^{-1}dpB_0(d\omega)
\]  

(3.16)

Similarly, a gamma process can be obtained as a special case of a compound Poisson process [97] and an improper Gamma distribution is used instead of an improper beta distribution. Details can be found in [81]. Normalization of a sample from the gamma process results in a sample from
a Dirichlet process [52]. Therefore, Poisson process is at the core of much literature on Bayesian nonparametrics.

In this dissertation, use of the Poisson process prior directly to model the unknown number of objects and their locations is explored. Each location is associated with a mark. The mark (shape model parameters) is latent and is inferred as part of the model. This work presents a new class of Bayesian nonparametric modeling based on a marked spatial Poisson process prior.

### 3.3 Sampling Techniques

Sampling techniques offer a means to do Bayesian inference on models that do not require conjugate priors to be defined. Samples are generated from the posterior distribution of the variable being inferred. By generating enough samples, one can achieve the desired level of accuracy. Details of how to determine if ‘enough’ number of samples are generated and different methodologies to generate them are presented in detail in this section. First, Metropolis-Hastings algorithm that is a type of Monte Carlo Markov Chain (MCMC) method to sample from a distribution is described. Then, variations of this algorithm popularly known as reversible jump Markov chain Monte Carlo (RJMCMC) and Gibbs sampling are described.

The Markov Chain Monte Carlo (MCMC) sampling method constructs a state space \( \mathbb{X} \) whose stationary distribution is the target density \( p(x) \). A random walk on the state space is done such that the amount of time we spend in each state is proportional to the density \( p(x) \). A Markov transition kernel \( P(x, x') \) that is aperiodic and irreducible is defined such that the detailed balance given by Equation 3.17 is satisfied.
\[
\int_A \int_B p(x)P(x, x') dxdx' = \int_A \int_B p(x')P(x', x) dx'dx
\] (3.17)

This equation ensures equilibrium probability to move from \(A\) to \(B\) and \(B\) to \(A\), for all Borel sets \(A\) and \(B\) in \(C\), where \(C\) denotes the parameter space. A detailed balance equation ensures ergodicity and the correct limiting distribution of the target \(p(x)\).

In the Metropolis-Hastings algorithm, the random variable being sampled is first initialized to an initial value. Then, a proposal to jump from a given state \(x\) to \(x'\) is made from an arbitrary distribution \(q(x'|x)\). \(q\) is called the proposal distribution. Once a new state is proposed, it is either accepted or rejected based on a formula that ensures that the amount of time spent in each state is proportional to \(p(x)\). Let \(p(x) = \frac{1}{Z} \tilde{p}(x)\), where \(\tilde{p}(x)\) is the unnormalized distribution and \(Z\) is the normalization constant. The acceptance probability \(r\) for a sample \(x'\) is given by

\[
\chi(x, x') = \min\{1, A_m\}
\] (3.18)

\[
A_m = \frac{(\tilde{p}(x'))q(x|x')}{(\tilde{p}(x))q(x'|x)}
\] (3.19)

Note that this implies that one needs to know only the form of the probability density function. Samples can be drawn even when the normalization constant \(Z\) is unknown.

Consider the case when one of the unknowns in the model is the number of unknown random variables itself. In such a case, the above approach will not be able to sample from the target distribution \((p(x))\) as it requires a trans-dimensional jump. Such a trans-dimensional jump fails to satisfy the balance equation based on which the Metropolis-Hastings algorithm is designed. More
details on trans-dimensional jump and associated problems can be found in [63].

### 3.3.1 Reversible Jump Markov Chain Monte Carlo

A formulation that can handle the case of varying number of random variables is proposed in [63] and is called the Reversible Jump Markov Chain Monte Carlo (RJMCMC) method.

In the Metropolis-Hastings case, a new sample \( \mathbf{x} \) is proposed using the proposal distribution \( q \).

This formulation is modified slightly to derive the RJMCMC sampling framework. Let there be \( r \) random numbers \( \mathbf{u} \) with a probability density \( g \). The new state \( \mathbf{x}' \) be given by \( (\mathbf{x}', \mathbf{u}') = f(\mathbf{x}, \mathbf{u}) \), where \( f \) is a known deterministic function. Let the reverse transition be given by \( (\mathbf{x}, \mathbf{u}) = f'(\mathbf{x}', \mathbf{u}') \), where \( \mathbf{u}' \) is the random number with known density \( g' \), and \( f' \) is again a deterministic function.

The standard Metropolis-Hastings equation using the above notation is given by Equation 3.20.

\[
A_\mathbf{m}(\mathbf{x}, \mathbf{x}') = \frac{p(\mathbf{x}')g'(\mathbf{u}')}{p(\mathbf{x})g(\mathbf{u})} \tag{3.20}
\]

The modified detailed balance equation is given by Equation 3.21.

\[
\int_{(\mathbf{x}', \mathbf{x}) \in A \times B} p(\mathbf{x})g(\mathbf{u})A_\mathbf{m}(\mathbf{x}, \mathbf{x}')d\mathbf{x}d\mathbf{u} = \int_{(\mathbf{x}, \mathbf{x}') \in A \times B} p(\mathbf{x}')g'(\mathbf{u}')A_\mathbf{m}(\mathbf{x}', \mathbf{x})d\mathbf{x}'d\mathbf{u}' \tag{3.21}
\]

In a trans-dimensional case, the number of variables on the right-hand side is not equal to the number of variables on the left-hand side. Let the dimensions of the random variables \( \mathbf{x}, \mathbf{x}', \mathbf{u}, \mathbf{u}' \) be \( n, n', r, r' \) respectively. The functions that operate on this transformation are of the form:

\( f : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^{n'} \times \mathbb{R}^{r'} \) and \( f' : \mathbb{R}^{n'} \times \mathbb{R}^{r'} \rightarrow \mathbb{R}^n \times \mathbb{R}^r \). When the dimension does not change between states, \( n = n' = r = r' \). This separation in notation allows presenting a general case, which can handle all types of jumps. The equality criterion in the integral of the detailed balance
equation holds if

\[
p(x)g(u)\chi(x, x') = p(x')g'(u')\chi(x', x)|\frac{\partial(x', u')}{\partial(x, u)}|
\]  

(3.22)

Let \( J = \frac{\partial(x', u')}{\partial(x, u)} \) denote the Jacobian of the transformation from \((x, u)\) to \((x', u')\). The new acceptance ratio is then given by Equation 3.23.

\[
\chi(x, x') = \min\left\{ 1, \frac{(\tilde{p}(x'))g'(u')}{(\tilde{p}(x))g(u)} \times |J| \right\}.
\]

(3.23)

In the case where the dimension does not change between jumps, \( J = 1 \). Even in the case of a trans-dimensional jump, if the new dimension introduced or deleted is independent of the random variable in the other dimensions, \( J = 1 \). More details can be found in [63].

### 3.3.2 Gibbs Sampling

Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm [109]. In Gibbs sampling, all the proposals are always accepted. However, one needs to know the form of the conditional distribution of the variable being sampled in order to perform Gibbs sampling. Consider proposals of the form \( q(x'|x) = p(x'|x_{-i})\mathbb{1}(x'_{-i} = x_{-i}) \), where \( \mathbb{1} \) is equal to one when \( x'_{-i} = x_{-i} \). Note that \( x'_{-i} = x_{-i} \) since it is left unchanged. The new acceptance ratio is shown in Equation 3.24.
\[ A_m = \frac{p(x')q(x|x')}{p(x)q(x'|x)} = \frac{p(x'_i|x_{-i})p(x_{-i})p(x|x')}{p(x)|x_{-i})p(x)|x_{-i})} p(x'|x_{-i}) \]
\[ = \frac{p(x'_i|x_{-i})p(x_{-i})p(x|x')}{p(x)|x_{-i})p(x)|x_{-i})} = 1 \] (3.25)

This implies that the proposal is always accepted. Gibbs, therefore, updates one variable at a time conditioning on the other variables in the model. A slight modification of this approach where more than one random variable can be updated at a time is called blocked Gibbs sampling algorithm [128]. A trans-dimensional case cannot be handled by Gibbs sampling [63].
Chapter 4

Latent Marked Poisson Process

In this chapter, we present a Bayesian nonparametric model for segmenting multiple objects/patterns in the image. We integrate spatial prior information with an object recognition model taking into account both shape prior and image feature information. We utilize a spatial Poisson process (which can be non-homogeneous) [12] as a nonparametric prior for the number of objects along with their locations. Each object has a corresponding shape generated from a shape prior model which provides a partitioning of the image. When locations and shapes of objects are determined, the pixels that are inside objects are assumed to be generated from an object image feature/appearance model and pixels outside objects are assumed to come from a background feature/appearance model.

4.1 Model formulation

An image is represented by a data matrix $X \in \mathbb{R}^{D_r \times D_c}$, where $D_r$ and $D_c$ are the number of rows and columns respectively (total number of pixels, $D = D_r \times D_c$). Each element in this matrix is a pixel,
represented by a scalar in the case of a gray-scale image, a vector in the case of RGB color images and in general a feature vector based on image characteristics (e.g., wavelet, Fourier coefficients, co-occurrence or histogram features). To simplify notation, we assume two-dimensional images. These concepts are extended to three-dimensional images in Chapter 8.

The generative process of the model can be explained as follows. $N$ objects at locations $L = [l_1, \ldots, l_N]$ are generated as a random outcome from a Poisson process with the Poisson intensity parameter $\lambda(\tau)$, where $\tau$ represents the 2D image plane. Here $l_n = [l_{n,r}, l_{n,c}]$ correspond to row and column positions respectively. A shape parameter for each object, $S = [s_1, \ldots, s_N]$, with origin/center at $L$, is generated from a shape prior distribution with hyper-parameter $\zeta$. The shape parameters define the object contour and thus a partitioning of the image: observation data (the feature representation of each pixel) inside and outside the contour are generated from foreground and background appearance models with parameters $\alpha$ and $\gamma$ respectively. Let all the hyper-parameters be represented by $\theta = [\lambda(\tau), \zeta, \alpha, \gamma]$.

The joint model is given by Equation 9.1. Likelihood of the image given all the parameters is given by Equation 9.2:

$$p(L, S, X|\theta) = \left[ p(L|\lambda(\tau)) \cdot p(S|L, \zeta) \right] \times p(X|L, S, \alpha, \gamma) \quad (4.1)$$

$$p(X|L, S, \alpha, \gamma) = \prod_{d=1}^{D} [p(x_d|\alpha)]^{I_d} [p(x_d|\gamma)]^{1-I_d}, \quad (4.2)$$

where $I_d$ is a random variable that takes a value 1 when the pixel $x_d$ belongs to an object and 0, when the pixel does not belong to an object (belongs to background). Note that this random variable is a deterministic function of $L$ and $S$. A proposal representation of the model is shown in
4.1.1 Spatial Poisson Process Prior

A sample from a spatial Poisson process consists of a random number of points at random locations on the 2D plane based on the underlying intensity function $\lambda(\tau)$. When $\lambda(\tau) = \lambda$, it is called homogeneous. When $\lambda(\tau)$ varies with $\tau$, we say it is non-homogeneous. A Poisson process in general is defined on a locally compact metric space $S$ with intensity measure $\Lambda$ (which is finite on every compact set and has no atoms), as a point process on $S$ such that, for every compact set $B \subset S$, the count $N(B)$ has a Poisson distribution with mean $\Lambda(B)$. If $B_1, \ldots, B_m$ are disjoint compact sets, then $N(B_1), \ldots, N(B_m)$ are independent [12]. In a spatial setting, $S = \mathbb{R}^2$ and $\Lambda(B) = \int_B \lambda(\tau) d\tau$.

An example of the outcome from a homogeneous Poisson prior is shown in Figure 4.2. The number of objects that can be expected in the 2D plane is given by the Poisson intensity parameter.
The distribution of these objects in the 2D plane is uniform. An example of the outcome from a non-homogeneous Poisson process is shown in Figure 4.3. The colormap of contours in Figure 4.3 represent the value of Poisson intensity at each point in the 2D plane. Red indicates a high Poisson intensity value and blue indicates a low Poisson intensity value. One should expect to see more object locations in the region containing red color compared to the regions that contain blue color.

![Figure 4.2: Outcome of a homogeneous Poisson process.](image)

(a) \( \Lambda(\mathcal{B}) = 10 \)  
(b) \( \Lambda(\mathcal{B}) = 30 \)

Figure 4.3: Outcome of a non-homogeneous Poisson process.

![Figure 4.3: Outcome of a non-homogeneous Poisson process.](image)
4.1.2 Shape Prior

We wish to incorporate the knowledge of distinct shape characteristics as a shape prior. In this work, we explore two approaches to building a shape model: in the first approach we take the shape model to be a known simple parametric form, and in the second approach we consider complex shape models.

We consider an ellipse to demonstrate a simple parametric model for shape. Ellipses can be described by three parameters: the major axis $r_a$, the minor axis $r_b$ and rotation angle $\rho$ (which implies $s = [r_a, r_b, \rho]$). In a given image, there can be objects with elliptical shapes with different parameter values of the minor axis, major axis, and rotation. One could assume uniform priors on these parameters (that are hyper-priors of the overall model) or any other appropriate distribution based on either domain knowledge or training data. Figure 4.4(a) shows the parameters of the ellipse shape model.

![Figure 4.4](image)

(a) Shape parameters of an ellipse. (b) Landmark points for shape prior of pedestrians.

To illustrate the use of a complex shape prior, we adapt the point distribution model for complex shapes proposed in [33]. In this approach, a complex shape is parameterized by landmark points
along the boundary of the object one wants to detect. Landmark points are representative points labeled by a domain expert or annotator for capturing shape constraints, such as points that are significant depending on the application, highest point of an object, curvature extrema, or points in between along the boundary. Figure 4.4(b) shows an example of a 16-point model of the boundary of a person. Each point on the object has a row and column element, giving rise to a 32-dimensional vector. Therefore, there are $m$ parameters for a $m/2$ point model. Each shape, $s$, is represented by a $m$-dimensional vector with elements comprising the row and column location of each landmark point with respect to the origin.

A prior shape model is constructed as follows. Given a set of $P$ training shape templates, the shapes are first aligned by adjusting the scale, translation and rotation that minimizes a weighted sum of squared distances between equivalent points on each shape template as described in [33]. Then, for each shape, $s_i$, where $i = [1, ..., P]$, we calculate its deviation, $d_{si}$, from the mean, $\bar{s}$:

$$d_{si} = s_i - \bar{s}$$

and the covariance matrix $C$, using

$$C = \frac{1}{P} \sum_{i=1}^{P} d_{si}d_{si}^T$$

(4.4)

One can sample directly from this mean and covariance matrix to generate a synthetic shape. If the application consists of shapes at different scales and orientations, additional variables that represent the scaling and rotation of the shape are introduced. When the number of landmark points is large, principal component analysis (PCA) can be used for dimensionality reduction as described in [33]. Note that this framework to represent complex shapes is chosen for illustration.
of an example; our framework does not restrict the form of shape model that one can deploy. In general, shape priors are parameterized models whose parameters are inferred during training from template images; for simple shapes, one can utilize domain knowledge prior belief or a broad prior.

### 4.1.3 Feature/Appearance Model

We assume that data (pixels), $x_d$, inside a shape contour is generated from an object feature/appearance model, $p(x_d|\alpha)$; and data outside any shape boundary are generated from a background model, $p(x_d|\gamma)$. As indicated earlier, an observation pixel, $x_d$, can be represented by its gray-scale value or color values. For textured objects, $x_d$ would be represented by a feature vector based on image characteristics (e.g., wavelet, Fourier, co-occurrence, histogram features). One can utilize a multinomial distribution over pixel intensities or a truncated Gaussian distribution based on the features used for the application. Challenges associated with inference on this model along with a solution to work around it is presented in Chapter 5.
Chapter 5

Inference

Finding the posterior distribution of the number of objects, their locations and corresponding shapes in a Bayesian framework for a continuous location parameter is challenging due to the need to compare different model orders. In addition, we must treat non-conjugate priors that result from the fact that the likelihood depends on partitioning based on shape boundaries. Two strategies to accomplish inference are presented in this Chapter: RJMCMC and Gibbs sampling.

A standard method to solve inference in a trans-dimensional case with complex likelihood and priors is to employ Metropolis-Hastings sampling within the RJMCMC framework. In this chapter, first, inference of parameters $[N, L, S]$ given the observation $X$ using RJMCMC is presented. This framework for inference is very flexible, can handle change in dimensions of parameters, any type of distribution, unknown normalization constant and non-conjugate priors. However, it is advantageous to use Gibbs sampling as it samples directly from the conditional posterior distribution, does not involve rejections. In addition, one does not have to worry about choosing an appropriate proposal distribution. The Gibbs sampling strategy cannot handle cross-dimensional jumps and hence cannot be used directly for nonparametric inference. We present a trick to overcome this
constraint and use Gibbs sampling for inference.

5.1 Reversible Jump Markov Chain Monte Carlo Inference

The Metropolis-Hastings equation for RJMCMC was previously described in 3.23. A slight modification of this Equation to include probability of proposing different transition types is presented in Equation 5.1.

\[
A_{m,t}(x, x') = \frac{\tilde{p}(x') j_t(x') g'(u')}{\tilde{p}(x) j_t(x) g(u)} |J| \tag{5.1}
\]

where, \( t \) denotes the type of move and \( j_t(x) \) denotes the probability that move of type \( t \) is attempted at state \( x \). Introducing \( j_t(x) \) improves the mixing of the chain and facilitates faster convergence. A ‘move type’ refers to the type of jump in states one wants to make in the Markov chain. Details are provided in the following paragraphs to make it clearer. The Jacobian term vanishes (equals to one) when the number of parameters does not change, in the proposed state, or the proposal is an independent distribution. As a first step, we describe the inference for a simple case where shape is assumed to be constant, and one only needs to infer the number of objects along with their locations. Later, we show how adding the shape variable is a simple additional sampling step in the RJMCMC framework.

Consider, inference of the parameters \([N, L]\) given the spatial Poisson prior \( \lambda(\tau) \). The possible transitions during Metropolis-Hastings sampling are (a) sampling the new location, (b) sampling the birth of an object and (c) sampling the death of an object. This implies that there are three types of moves.

Let the upper limit on the number of objects allowed in the image be denoted by \( N_{max} \). This
value could be as high as the number of pixels in the image. Therefore, all possible transitions can be made with a countable set of moves. These moves can be denoted by \( t = L, 0, 1, 2, \ldots \), where \( L \) represents change in location. \( n_o = 0, 1, 2, \ldots \) denotes the birth or death of an object where the number of objects are increased from \( n_o \) to \( n_o + 1 \) or decreased from \( n_o + 1 \) to \( n_o \). Each transition type is associated with a probability denoted by \( j_t \) as was shown in Equation 5.1. Let \( j^n_{L} \), \( j^n_{b} \) and \( j^n_{d} \) be the probability associated with the change in location, birth and death of an object respectively, subject to the constraint \( \sum j^n_{t} = 1 \). Naturally \( j^0_{L} = 0 \) and \( j^0_{d} = 0 \). One can achieve much better mixing by letting the probability \( j_t \) of each transition type to vary depending on the number of objects in the current state \( n_o \) as demonstrated in [63]. A similar approach is followed to arrive at suitable transition probabilities. The probability of making a move of type birth and death are chosen to be \( j^n_{b} = a \min[1, p(n_0 + 1)/p(n_o)] \) and \( j^n_{d} = a \min[1, p(n_o)/p(n_o + 1)] \) respectively, where \( a \) is a constant chosen to be as large as possible with the constraint \( j^n_{b} + j^n_{d} < 0.9 \) for \( n_o = 1, \ldots, n_{\text{max}} \).

This framework can be extended to multiple categories. When more than one category of objects exists, another type of move can be introduced. This move would propose the change of category of an existing object. Note that this move is equivalent to death of an object in one category along with birth of an object in the other category, simultaneously at the same location.

### 5.1.1 Inference on the Complete Model

Inference is discussed assuming two categories. Note that the same formulation can be extended to any number of categories. Initialize the number of objects to a random value \([N^0_1, N^0_2]\) which could be the maximum number of objects one wants to allow in the image. The value in the power of a
variable being sampled represents the iteration number. Initialize the locations \( L_1^0, L_2^0 \) to a random value by sampling from a uniform distribution. Now, one can decide the move type by sampling from transition probability \( j_t \).

1. If a move of type ‘\( L' \)’ is made, index of the object for which a new location is proposed is chosen at random, by sampling from a uniform distribution over all the locations in the current state. Let the index be denoted by \( i \). We consider a symmetric Gaussian proposal with a mean value given by current location and a constant variance. Equation 5.2 shows the Metropolis-Hastings step for sampling a new location of the \( i^{th} \) object from category \( c = 1 \).

Note that the proposal ratio is equal to one due to symmetry.

\[
A_{m,L}(L', L) = \frac{p(X'|L_{1,i}^t, \cdot)}{p(X|L_{1,i}^t, \cdot)} \cdot \frac{p(L'|\lambda_1)}{p(L|\lambda_1)}
\]

(5.2)

2. For a move of type birth (where a new object is added to the existing state), location of the new point is sampled from a uniform distribution \( l_{c,n,p} \sim U[0, D_r], l_{c,n,q} \sim U[0, D_c] \) respectively, where \( l_{c,n,p} \) and \( l_{c,n,q} \) denote row and column location of \( n^{th} \) point from category \( c \). Since the proposal for new location during the birth of a point is from an independent distribution, the Jacobian term equals to one. Equation 5.3 shows the sampling step for birth of an object of category \( c = 1 \).
\[ A_{m,b}([N+1, L'], [N, L]) = \frac{p(X|L'_{i,N_i+1})}{p(X)} \cdot \frac{1}{p(L_{i,N_i+1}|P, Q)} \cdot \frac{p(L'_{i,N_i+1})}{p(N_i+1|\lambda_i)} \cdot \frac{p(N_i|\lambda_i)}{p(N_i+1|\lambda_i)} \]  \tag{5.3}

Move of type death (d) of an object is the exact inverse of Equation 5.3.

3. Adding a random variable that represents the shape of an object translates to an additional move type (s) to the Metropolis Hastings step. The proposal distribution is assumed to be symmetric Gaussian with the mean value given by current shape and a constant variance.

The Metropolis-Hastings step is similar to that of Equation 5.2 and is given by Equation 5.4.

\[ A_{m,s}(S', S) = \frac{p(X|S'_{i,i})}{p(X|S_{i,i})} \cdot \frac{p(S'_{i,i}|\zeta)}{p(S_{i,i}|\zeta)} \]  \tag{5.4}

In the experiments, we stop sampling when the first and second order statistics of the parameters stop changing. We assume a constant value for burn in based on our observation of the Markov chain and use it for all the consecutive runs.

### 5.2 Gibbs Sampling

As mentioned before, it is advantageous to use Gibbs sampling instead of the RJMCMC framework. However, this sampling strategy cannot be used directly for models involving trans-dimensional jump. To make Gibbs sampling possible, one can take advantage of the fact that there are only a
finite number of pixels in an image and that we only care about location information up to a pixel resolution. For example, it is enough for us to know that an object center occurs in an area $d\tau$ covered by a pixel rather than the exact continuous value of the location. A Gibbs sampling strategy based on this approximation is presented in the following paragraphs.

### 5.2.1 Leveraging the Finite Nature of Image Pixels

We introduce a latent variable $Z = [z_1, \ldots, z_D]$, where each $z_d$ is a discrete variable representing the number of object centers at a particular pixel location $d$. Note that the number of object centers that can occur at any given pixel is unbounded.

According to the definition of a spatial Poisson process, for disjoint sets $B_1, \ldots, B_N$, the number of objects $N(B_1), \ldots, N(B_N)$ are independent random variables. In our case, the disjoint sets are individual pixels. This implies that the probability of occurrence of an object center at any given pixel is independent of all the other pixels in the image. This value depends only on the underlying Poisson intensity parameter in the area covered by the pixel. Let this value of Poisson intensity parameter at each pixel be given by $\lambda_d = \int_{p-1}^{q} \int_{q-1}^{p} \lambda(u, v)dudv$, where $p$ and $q$ are the indices of the pixel and $u$ and $v$ are continuous random variables that represent location in the 2D plane. $z_d$ has a Poisson distribution with parameter $\lambda_d$. The total number of objects $N$ in the image has a Poisson distribution with parameter $I$, where $I$ is the integral of the intensity parameter of the Poisson process for the entire image given by $I = \int_\tau \lambda(\tau)$.

Given the number of objects $N$ in the image, the joint distribution is a multinomial distribution as shown in Equation 5.5. The resulting model, after introduction of $Z$ is given by Equation 5.6 and the graphical model is shown in Figure 5.1. The random number of objects in the image can
be represented by $\sum_{d=1}^{D} z_d$. We have:

$$p(Z|N) = \frac{p(Z)}{p(N|I)} = \frac{\prod_{d=1}^{D} \frac{\exp(-\lambda_d)}{\lambda_d!} z_d^{\lambda_d}}{\frac{N!}{\prod_{d=1}^{D} \lambda_d!}} = \frac{N!}{z_1! \ldots, z_D!} \left(\frac{\lambda_1}{I}\right)^{z_1} \ldots \left(\frac{\lambda_D}{I}\right)^{z_D}$$  

(5.5)

$$p(Z,S,X|\theta) = p(Z|\lambda(\tau))p(S|L, \xi)p(X|Z, S, \alpha, \gamma).$$  

(5.6)

### 5.2.2 Complete Spatial Randomness

Inference for the proposed model involves calculating the posterior distribution over the variables $(Z, S)$ given the observation $X$. We develop a Gibbs sampling framework to perform this inference. A naive application of Gibbs sampling would compute the following conditional probability:

$$p(z_d|Z_{-d}, S, \theta) \propto p(z_d|Z_{-d}, \lambda(\tau))p(X|Z, S, \alpha, \gamma).$$  

(5.7)
One of the properties of the spatial Poisson process is that the distribution of points in the 2D plane is completely random and does not depend on the presence of other points. The occurrence of a point at any location depends only on the underlying Poisson intensity parameter. The Poisson process likelihood is given by:

\[
p(\{l\}_{n=1}^{N}|\lambda(\tau)) = exp\left(- \int \tau \lambda(\tau) \right) \prod_{n=1}^{N} \lambda(l_n) \tag{5.8}
\]

However, calculating likelihood after sampling \(z_d\) in this way does not capture the complete spatial randomness property of the Poisson process for a given set of object locations. The random variable \(Z\) has a distribution which is a functional of the spatial Poisson process. However, it is not possible to invert this functional by considering the values of \(z_d\) independently.

---

**Figure 5.2:** (a) Location Set-1 and (b) Location Set-2 where the image is divided into 16 regions. (c) and (d) show the division of space into 4 regions. Value of Poisson log-likelihood for each configuration is reported below the corresponding figure.

(a) \(-6.158\)  (b) \(-6.158\)  (c) \(-3.386\)  (d) \(-4.079\)

---

**Figure 5.3:** (a) Location Set-1 (b) Location Set-2 with \(M = 4\) and (c), (d) show the same set with \(M = 9\). Value of Poisson log-likelihood for each configuration is reported below the corresponding figure.

(a) \(-5.688\)  (b) \(-8.816\)  (c) \(-8.67\)  (d) \(-13.03\)
As an illustration, consider Figure 5.2(a) and 5.2(b) as two sets of observations and the goal of calculating the likelihood of these observations as the outcome of a uniform Poisson process with prior \( \int \lambda(\tau) = 2 \). The number of points in both sets is the same. Intuitively, one expects the likelihood value to be low when the observations are clustered and high when they are spread apart. When the image is discretized into 16 parts; both observations will have the same likelihood according to \( \log(\prod_{d=1}^{16} p(z_d|\lambda)) = -6.158 \). If instead the image is discretized into 4 pixels/parts, the grid in Figure 5.2(c) (−3.386) has higher likelihood compared to Figure 5.2(d) (−4.079). The relative likelihood value depends on the number of parts the image has been divided into and \( \prod_{d=1}^{D} p(z_d|\lambda) \) cannot capture this spatial randomness as every value of \( z_d \) is independent [134]. In our case, digital images are naturally discretized into pixels and introducing the latent variable \( z_d \) for the presence of an object center cannot distinguish clustered objects from uniformly distributed objects (assuming homogeneous Poisson intensity). A similar issue is encountered in maximum likelihood estimation of the Poisson intensity parameter where distance methods are proposed to ameliorate this problem [120].

Distance methods assume a grid of \( M \) points evenly distributed on the surface of the image (denoted by hollow red circles). The probability of finding the nearest object/observation point at a distance \( r_m \) from the \( m^{th} \) grid point is given by \( p(r_m) = (2\pi r_m \lambda)\exp(-\pi r_m^2 \lambda) \). Let the distances of nearest object occurrence from these grid of \( M \) points be denoted by \( r_1, r_2, \ldots r_M \). For example, Figures 5.3(a) and 5.3(b) assume four grid points in the image. We will describe how the number \( M \) can be chosen and its associated trade-offs later in this section. The likelihood of sample distances
$r_1, r_2, \ldots, r_M$ is given as follows [120]:

$$p(R) = \prod_{m=1}^{M} \sum_{d=1}^{D} (2\pi r_m)(p_d \lambda_d) \exp(-\pi r_m^2 \lambda_d), \quad (5.9)$$

where $R = [r_1, \ldots, r_M]$ and $p_d$ is the proportion of image with intensity $\lambda_d$. In a homogeneous case with $D$ pixels, this value is equal to $1/D$ and this equation reduces to:

$$p(R) = (2\pi \lambda)^D \exp(-\pi \lambda \sum_{d=1}^{D} r_m^2)(r_1 r_2 \ldots r_m) \quad (5.10)$$

The likelihood value in terms of the distances $R$ given in Equation 5.9 captures the spatial randomness for a given set of object locations and intensity value. For example, $p(R)$ in Figure 5.3(a) ($-5.688$) is higher compared to Figure 5.3(b) ($-8.816$). Even in a case where more number of grid points are chosen ($M = 9$), the likelihood of configuration in Figure 5.3(c) ($-8.67$) is higher than the configuration of Figure 5.3(d) ($-13.03$). This is a desirable property to determine the configurations that are more likely to occur than others.

Note that [120] introduced distance methods for point estimation of the Poisson intensity parameter, $\lambda(\tau)$, using maximum likelihood criterion, given observed Poisson locations. We, on the other hand, are interested in exploring the possible object locations $z_d$ and their likelihood given the intensity value. We view $R$ as an auxiliary variable (as shown in Equation 5.11) that captures the spatial randomness of the objects in the image. Based on this augmentation, we derive a Gibbs sampler as follows:

$$p(z_d, R|Z_{-d}, S, \theta) \propto p(X|Z, S, \theta)p(z_d, R|Z_{-d}, \lambda(\tau)) \quad (5.11)$$

Note that $R$ is deterministic given $Z$. Given that we are interested only in $z_d$ for the purposes of
inference, we discard the values of $R$ after they are used in the sampling process. We have:

$$p(z_d, R|Z_{-d}, \lambda(\tau)) = p(R|Z_{-d}, \lambda(\tau))p(z_d|Z_{-d}, R, \lambda(\tau)).$$  \hspace{2cm} (5.12)$$

Note that $z_d$ can take discrete values in the range $[0, \infty)$; i.e., we do not place any restrictions on the number of object centers that can appear in one pixel.

For Gibbs sampling, all the variables except the one being sampled, are assumed to be known and constant. This implies that one needs to sample the values of $R$ with only one unknown value $z_d$. The possible values that $R$ can take given $Z_{-d}$ reduces to two cases ($R_o$ and $R_\theta$, if $z_d = 0$ or $z_d > 0$ respectively) shown by Equation 5.13.

$$p(R|Z_{-d}, \lambda(\tau)) = \begin{cases} 
p(R_o), & \text{if } z_d = 0 \\
p(R_\theta), & \text{if } z_d > 0 
\end{cases} \hspace{2cm} (5.13)$$

This value for both cases can be calculated from Equation 5.9 as $R$ is deterministic given the values of $Z$. All the cases where $z_d > 0$ give the same $R = R_o$ value as the distance between a fixed point and its nearest object remains the same irrespective of the number of objects present in that pixel. This leads to the following conditional probabilities:
\begin{align*}
p(z_d = k, R|Z_{-d}, \lambda(\tau))_{k=0} &= (1/H)p(R_o)p(z_d = 0|Z_{-d}, R_o) \\
&= (1/H)p(R_o)e^{-\lambda_d} \quad (5.14) \\
p(z_d = k, R|Z_{-d}, \lambda(\tau))_{k \neq 0} &= (1/H)p(R_o)p(z_d = k|Z_{-d}, R_o) \\
&= (1/H)p(R_o)(e^{-\lambda_d} \lambda_d^k / k!) \quad (5.15)
\end{align*}

\begin{align*}
H &= \sum_{k=0}^{\infty} p(R)p(z_d = k, R_k|Z_{-d}) \\
&= p(R_o)e^{-\lambda_d} + p(R_o) \sum_{k=1}^{\infty} \frac{e^{-\lambda_d} \lambda_d^k}{k!} \\
&= p(R_o)e^{-\lambda_d} + p(R_o)(1 - e^{-\lambda_d}) \\
&= (5.16)
\end{align*}

To set the number of grid points \( M \) for calculating \( R \), we notice that if \( M \) is too sparse compared to the number of objects in the image, most values of \( z_d \) will have no influence on the value of \( R \) and \( p(z_d, R|Z_{-d}, \lambda(\tau)) = e^{\lambda_d} \lambda_d^d / z_d! \); this ceases to capture the spatial randomness of the objects. On the other hand, having the grid points that are too dense will result in increased computational expense. We chose \( M \) to be 10 pixels apart for our experiments.

Step 1 of Algorithm 9.3 provides the pseudocode for sampling \( z_d \) based on three cases: deleting of a point, remaining at the same state and adding a point. This involves calculating the likelihood for all the possible cases followed by normalization giving a discrete probability distribution. The
Algorithm 1: Gibbs Sampling for a Marked Poisson Process

**Initial:** Sample the variables \( \{z_d\}_1^D \) for every pixel and \( \{s_n\}_1^N \) for every object, where number of objects is given by \( N = \sum_1^D z_d \). Given a set of these values from the previous iteration, sample a new set as follows:

**Step 1:** \( Z = Z' \) and \( S = S' \). For \( d = 1, \ldots, D \), draw a new sample for \( z_d \) from the following probabilities:

\[
p(z_d = k, R|Z_{-d}, S, \theta) \propto \begin{cases} p(z_d = k, R|Z_{-d}, \lambda(\tau)) p(X|Z, S, \alpha, \gamma), & \text{if } k \leq k' \\ p(z_d = k, R|Z_{-d}, \lambda(\tau)) \int p(X|Z, S, \alpha, \gamma) p(s_{N+1}|\zeta) ds, & \text{if } k = k' + 1 \end{cases}
\]

if \( k = k' + 1 \), sample a new parameter \( s_{N+1} \) from \( p(s_{N+1}|\zeta)p(X|Z, S, \alpha, \gamma) \) and \( N = N + 1 \); if \( k < k' \), delete an object and reduce dimension of \( s, N = N - 1 \).

**Step 2:** For \( n = 1, \ldots, N \)
Sample the shape parameter of each object: \( p(s_n|Z, S_{-n}, \theta) \propto p(X|Z, S, \theta)p(s_n|\zeta) \)

**Step 3:** \( t = t + 1, Z' = Z \) and \( S' = S \)

**REPEAT** steps 1, 2 and 3 until convergence.

**Output:** We assume convergence when first order statistics varies below a pre-set threshold \( \epsilon \): save and report the mean values of all object’s location and shape.

The final value of \( z_d \) is sampled from this distribution. This is similar to the Gibbs sampling framework for Dirichlet Process Mixtures.

The shape variables are assumed to be independent of each other. Sampling the shape variable is given by Equation 5.17 and constitutes Step 2 of Algorithm 9.3:

\[
p(s_n|S_{-n}, \zeta, \xi) \propto p(X|Z, S, \theta)p(s_n|\zeta)
\]  

(5.17)

Because our prior is not conjugate to the observation likelihood, we use sampling-resampling to approximate a sample from the posterior [135]. We sample a fixed number (300) of shapes from the prior distribution and then calculate the corresponding likelihood for each case. We then sample a shape from these candidate shapes by sampling from a discrete distribution (for each shape model) whose weights are the normalized scores of Equation 5.17.
5.3 Inference of the Non-homogeneous Poisson Prior

Inference of the non-homogeneous Poisson intensity prior given a set of training examples is not straightforward. In this section, we present the details regarding learning a non-homogeneous Poisson intensity prior parameter.

Let the set of observations in a region \( \tau \) be denoted by \( \{l_n\}_{n=1}^N \). Given, these observations, one needs to infer the underlying Poisson intensity parameter \( \lambda(\tau) \). One does not know the functional form of the intensity parameter a priori. Therefore, a non-parametric formulation that can encompass all possible patterns is chosen to model the intensity. Such a construction involving another stochastic process to represent the Poisson intensity prior is called a doubly stochastic Poisson process or a Cox process. A Gaussian process can capture all the possible function forms, but the outcome has to be restricted to a positive value since the intensity parameter is always positive.

To accomplish this, we utilize a log Gaussian Cox process. This formulation has the following advantages: (a) no problem of edge effects, (b) intensity within a bounded window, given a realization of the process can be predicted using Bayesian methods, (c) higher-order properties take a simple expression and theoretical properties can be easily derived, and (d) parametric models can be interpreted easily. This framework models the intensity as an exponential function of a random realization from a Gaussian process: \( \lambda(\tau) = \exp(g(\tau)) \), where \( g(\tau) : \mathbb{R}^2 \rightarrow \mathbb{R} \) is a random scalar function having a Gaussian process prior.

The likelihood value of the Poisson process for a given set of observations is given by

\[
p(l_{n=1}^N | \lambda(\tau)) = \exp\{-\int_\tau d\lambda(1) \prod_{n=1}^N \lambda(l_n) \}
\]

where, \( \lambda(\tau) = \exp(g(\tau)) \). Inferring \( \lambda(\tau) \) given a set of observations requires inference of the
infinite-dimensional object corresponding to \( g(\tau) \) denoted by \( g \). The posterior distribution of \( g \) is given by

\[
l(\exp(g) | \{l_n\}_{n=1}^{N}) = \frac{\mathcal{GP}(g) \exp \left(- \int_{\tau} \exp(g) d\tau \right) \prod_n \exp g(l_n)}{\int dg \mathcal{GP}(g) \exp \left(- \int_{\tau} \exp(g) d\tau \right) \prod_n \exp g(l_n)}.
\]

We use the MCMC framework to accomplish this. The general Metropolis-Hastings step to jump from \((x)\) to \((x')\) is given by Equation 5.19, where, \( x \) is the current state of the parameters to be inferred and \( q \) is called the proposal distribution. The value \( \min\{1, A_m\} \) determines the probability with which the proposed move to jump to \( x' \) will be accepted (A detailed explanation is provided in Section 3.3).

\[
\chi(x, x') = \min\{1, A_m\} \quad (5.19)
\]
\[
A_m = \frac{\tilde{p}(x')q(x'|x)}{\tilde{p}(x)q(x|x')} \quad (5.20)
\]

In our case, the random variable being sampled is the Gaussian process \( g \). The posterior distribution is doubly-intractable (due to the presence of an integral over the Gaussian process in both numerator and denominator) and a standard MCMC method cannot be used directly for inference. The problem of intractability is overcome by using the trick of generating fantasy events [8, 105, 104], such that the integral is canceled out in both numerator and denominator making MCMC sampling feasible. Fantasy events \( \{e\} \) are generated from the current state and the proposed state of the random variable being inferred. In our case, that would be the locations of new
points \( \{e\} \) that we generate apart from the given observations \( \{l_n\}_{n=1}^{N} \) (based on the current and proposed states of \( g \)). As a result of plugging this into Equation 5.19, the intractable integral cancels out. The new Metropolis-Hastings step is given by Equation 5.21. We refer the reader to [8] for more details on the sampling process.

\[
A(x, x') = \frac{(\tilde{p}(x'))q(x'|x)\tilde{p}(e)}{(\tilde{p}(x))q(x'|x)\tilde{p}(e')}
\]

(5.21)

We use delayed rejection sampling [62] to minimize rejections resulting in faster convergence. An example problem is shown in Figure 5.4(a).

Figure 5.4: An example of inference of Poisson intensity given a set of observations.

The points in the 2D plane are drawn from a Poisson process with intensity given by the contours shown in Figure 5.4(a). The inverse problem is to infer the intensity (represented by the underlying contours) given the observations (points in the figure); i.e., we need to calculate the posterior distribution of the underlying intensity. Figure 5.4(b) shows the inferred intensity given
the observations. Ideally, the inferred contours in Figure 5.4(b) and the actual (ground-truth) contours of Figure 5.4(a) should be the same. We infer the contours (intensity parameter) given only one sample (one set of observations/points drawn from a Poisson process), and the results are very close. This learned probability map provides us with a spatial context prior indicating the high-density and low-density spatial regions of object occurrence. Given a set of training images with object location annotations, we infer the intensity parameter \( \lambda(\tau) \) and utilize this learned probability map as the spatial prior for our model.
Chapter 6

Extensions to the Model

In this chapter, we present the possible extensions to the proposed model. We present two types of extensions, one involving multiple categories with the same appearance model and the other involving multiple categories with different appearance models.

6.1 Multiple Categories with Same Appearance Model.

In this section, we address the problem of having to detect multiple categories in an image. For example, one might want to detect the cars and pedestrians from a traffic surveillance video. Assuming there are $C$ categories of objects in the image, the joint likelihood is given by Equation 6.1, where each category has its own Poisson and shape prior parameters.

$$p(L, S, X|\theta) = \prod_{c=1}^{C} \left[ p(L_c|\lambda_c(\tau) \cdot p(S_c|L_c, \zeta_c) \times p(X|L, S, \alpha, \gamma) \right] (6.1)$$
where, $L_c = [l_1, \ldots, l_{N_c}]$ and $S_c = [s_1, \ldots, s_{N_c}]$. We demonstrate the advantage of using a non-homogeneous Poisson intensity as a prior in the case with multiple categories in the following experiments on synthetic data. The number of objects in each category is represented by $N_c$. However, the appearance prior for all the categories is still the same. An illustration of this scenario is presented in the second case study of the Chapter 7.

**Experiments on Synthetic data:**

We test the performance of the proposed method on synthetic images that contain two categories of objects: rectangles and squares. We compare the results with the algorithm having the exact framework to detect the objects, without spatial prior. The results obtained would demonstrate the advantage of using a spatial Poisson prior compared to a model without it.

The goal of the algorithm is to be able to infer the object location along with its category, given the appearance and spatial Poisson prior. We add different levels of Gaussian noise to these images to make the detection challenging. Notice that the appearance of objects in the image is significantly different from the background. In the absence of noise, both models can detect the objects easily. However, when noise is added, a Poisson prior helps to push the likelihood value towards the presence of an object as it favors a Poisson number of objects with mean given by the underlying intensity. This proves to be useful in the detection of objects under noisy conditions that are true in most real world scenarios. Five images are generated from one sample of the spatial Poisson process with varying levels of noise. No noise is added to the first image and the rest of the images have a Gaussian noise with parameters mean and variance given by $[-10, 30], [-20, 30], [-30, 30]$ and $[-40, 30]$ respectively followed by smoothing using a median filter of size $[5, 5]$. 50 different images were generated from different spatial Poisson priors. Each image has four additional noisy
images associated with it. The performance of the proposed algorithm on a total of 250 synthetic images is reported in Figure 6.2. A constant number of iterations (20000) were used for both the models to make a fair comparison. This number (of iterations) was determined to be the mean amount of time taken by the algorithms to converge when observed over long periods of time.

![Image (a) - Image without noise](image1)

(a) Image without noise  
(b) Image without noise  
(c) Image with noise parameters (-20,30)

(d) Image with noise parameters (-20,30)  
(e) Image with noise parameters (-40,30)  
(f) Image with noise parameters (-40,30)

Figure 6.1: Results of our algorithm on synthetic images. Figure (a), (c) and (e) show results from the proposed model with spatial prior. Figure (b), (d) and (f) show results from the proposed model without spatial Poisson prior.

Clearly, the model with spatial Poisson prior outperforms the model without the prior. These results are intuitive as the model with a prior will give a very low probability to a model with very few objects. This experiment was also performed on images where there were no objects and both models correctly detected no objects in every case.
6.2 Multiple Categories with Different Appearance Model.

In this section, we address the case where each category has its own unique appearance model. Let the appearance parameters for each category be represented by $\alpha_c$. The joint likelihood given by Equation 6.1 becomes problematic when there is overlap between objects. One needs to decide which object is on top in order to assign it to the appropriate appearance model. Based on this decision, the indicator variable $I_{d,c}$ is equal to one for one of the object categories $c$ or zero if it belongs to the background.

To overcome this issue, we introduce a random variable $b_{c,n}$ that represents the probability of the $n^{th}$ object of category $c$ to be on top. This random variable is assumed to be drawn from a uniform distribution $b \sim U[u_{\text{min}}, u_{\text{max}}]$, where $u_{\text{min}} = 0$ and $u_{\text{max}} = 1$. Let these parameters be denoted by $\varepsilon$ and $\mathbf{b} = [b_{1:N_1,1}, \ldots, b_{1:N_C,C}]$. In the generative framework of the model, this parameter is drawn for each location of the object of each category and when there is a overlap of shape boundaries, the object with higher $b$ value is assigned to be on top. The binary variable, $I_{d,c}$, is thus assigned according to Equation 6.3. The modified joint likelihood is given by Equation 6.2.
Note that given the locations, shape boundaries and the value of \( b_{n,c} \), \( I_{d,c} \) is deterministic.

\[
p(L_{1:C}, S_{1:C}, X(\theta)) = \prod_{c=1}^{C} \left[ p(L_c|\lambda_c(\tau) \cdot p(S_c|L_c, \zeta_c)p(b|\varepsilon) \right] \times p(X|L, S, \alpha_{1:C}, \gamma)
\]

(6.2)

\[
I_{d,c} = \begin{cases} 
\delta(c_n), & \text{if } b_n > b_{n'} \\
\delta(c_{n'}), & \text{if } b_{n'} > b_n 
\end{cases}
\]

(6.3)

Figure 6.2 shows images containing maple leaves. This data was collected to illustrate an example with multiple categories and different appearance. There are two types of leaves in the image based on the side on which it landed on the ground. Each has its own appearance model. Background consists of grass. In this case, overlap needs to be considered when detecting them. The shape was trained with 22 landmark points on 6 examples. Images used for training are shown in Figure 6.2. Results on ten images are shown in Figure 6.2. Red and blue color on the boundaries of the leaves indicate different categories. This experiment is conducted as a proof of concept for the extensions presented. The average segmentation accuracy of the experiments is 0.95 and the precision and recall rate for object detection is 1 and 0.94 respectively.

Figure 6.3: Templates used for training shape and appearance of maple leaves.
Figure 6.4: Results of our algorithm on dataset consisting of maple leaves with a different appearance.
Chapter 7

2D case studies

In this chapter, details of experiments on two real-world applications are provided. First application involves segmentation of cells from fluorescence microscopy images. Inference is accomplished in the unsupervised mode using a homogeneous spatial Poisson prior. Second application involves detection of cars and pedestrians (two categories) from traffic surveillance data. Inference is accomplished in the supervised mode, and a non-homogeneous Poisson prior is utilized.

7.1 Case Study 1: Cell Image Segmentation (in unsupervised mode).

The goal in this application is to segment and detect cell nuclei [32] in stained microscope images. The dataset contains two sets of images, ‘gnf’ and ‘ic100’, each containing 50 images. Resolution of each image is $[1024 \times 1344]$ pixels. The nuclei of the cell show variation in appearance, shape, and orientation. Shape of the cell nuclei is very close to an ellipse. Therefore, variation in shape is encompassed in a flexible shape prior that assumes an ellipse with a major axis ($r_{ma}$), minor
axis \((r_{mi})\) and angle \((r_{a})\) is chosen. Uniform priors are assumed on all the parameters as inference is performed in the unsupervised mode; \((r_{mi}, r_{ma}) \sim (U[0, \text{MAX}])\) where, \(\text{MAX} = 300\) and angle \((r_{a} \sim U[−\pi, \pi])\). A multinomial distribution is assumed over the intensity values (whose range is \([0, 255]\)) for the appearance prior. Since this is an unsupervised setting, appearance parameters for foreground and background are determined adaptively for each test image. To do this, the intensity of the image is divided into two clusters using the k-means algorithm. The resulting histograms for the two clusters are used to calculate the multinomial distribution parameters for the foreground (cell) and background appearance priors. Note that the cells are always brighter than the background. This domain knowledge is utilized to consider the cluster with a higher mean to belong to foreground.

Performance of the proposed algorithm is compared with other unsupervised algorithms. Bayesian nonparametric image segmentation (NBIS) [112] and graph-based algorithm (GB) [51] that can automatically determine the number of segments in an image. Spectral clustering (SC) [110] partitions the image into a constant number of regions specified by the user. This was set to 50 in our experiments. We also compare against level sets (LS) [47] and the algorithm (MINS) proposed in [99], both of which are capable of inferring segmentations and learning the number of objects. Code for implementation of all the competing algorithms mentioned is downloaded from the author’s website. The competing algorithms are sensitive to parameter tuning; hence, the parameters are tuned on a validation set of two images chosen randomly.

Table 7.1 reports the average segmentation results based on Rand index [123] (RandI) and average segmentation accuracy [49] (SA) on all the cell images. Rand index is the ratio of the number of pixels that have been classified correctly to the sum of correctly and incorrectly classified pixels [123]. Segmentation accuracy is defined as the ratio of true positives to the sum of true positives,
false positives, and false negatives [49]. We report Precision and Recall values for object detection. We follow the rules specified in [49] to calculate these values. Table 7.2 reports the detection results in terms of precision and recall. Note that our algorithm has a better performance compared to other algorithms in the experiments. Note that our algorithm has a better performance compared to other algorithms in the experiments. Graph based methods have very noisy detections as can be seen from Figure 7.1 and 7.2, showing high false positives and recall values but very poor precision as shown in Table 7.2.

Figure 7.1: Results of algorithms on a sample image.

<table>
<thead>
<tr>
<th>Method</th>
<th>RandI</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBIS</td>
<td>.80 ± .01</td>
<td>.54 ± .11</td>
</tr>
<tr>
<td>GB</td>
<td>.81 ± .008</td>
<td>.58 ± .1</td>
</tr>
<tr>
<td>SC</td>
<td>.82 ± .01</td>
<td>.63 ± .1</td>
</tr>
<tr>
<td>LS</td>
<td>.78 ± .01</td>
<td>.45 ± .09</td>
</tr>
<tr>
<td>MINS</td>
<td>.77 ± .008</td>
<td>.33 ± .09</td>
</tr>
<tr>
<td>Ours</td>
<td>.86 ± .006</td>
<td>.69 ± .03</td>
</tr>
</tbody>
</table>

Table 7.1: Segmentation results for cell data

Apart from (LS) [47] and (MINS) [99], all other algorithms used for comparison are primarily
Figure 7.2: Results of algorithms on a sample image.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gnf</td>
<td>ic100</td>
</tr>
<tr>
<td>SC</td>
<td>.55 ± .09</td>
<td>.29 ± .16</td>
</tr>
<tr>
<td>GB</td>
<td>.01 ± .005</td>
<td>.01 ± .002</td>
</tr>
<tr>
<td>NBIS</td>
<td>.27 ± .07</td>
<td>.16 ± .1</td>
</tr>
<tr>
<td>MINS</td>
<td>.67 ± .02</td>
<td>.13 ± .01</td>
</tr>
<tr>
<td>LS</td>
<td>.35 ± .14</td>
<td>.12 ± .01</td>
</tr>
<tr>
<td>Ours</td>
<td>.72 ± .01</td>
<td>.36 ± .06</td>
</tr>
</tbody>
</table>

Table 7.2: Detection results for cell data

designed for segmentation of images. This implies that one needs to post-process the results to determine which segments belong to the background and which ones belong to the foreground. This is accomplished using k-means. The intensity of the image is divided into two classes using the k-means algorithm and the resulting mean values are used to determine the clusters belonging to background and foreground. Clusters belonging to the category with low mean are assumed to be from the background.

The algorithm has been implemented in MATLAB and takes an average of 10 minutes per [1024 × 1344] image on a 12GB RAM, 2.4GHz computer. The NBIS and spectral clustering
(SC) algorithm needed the image to be scaled down to $[512 \times 512]$ and 20\% of the original size respectively making computation time comparison unfair. Graph based algorithm (GB) and MINS take 61 and 25 seconds per image respectively. However, their segmentation and object detection performance is much worse than our algorithm. Level sets method takes 15 minutes without the need to scale down the image. Parameters of mean and propagation weight were tuned for optimal performance. This was done once for each of the folders ‘gnf’ and ‘ic100’. The maximum number of iterations allowed was set to 500.

### 7.2 Case Study 2: Traffic Surveillance Dataset (in supervised mode).

The traffic surveillance dataset [151] contains the video of a static camera showing the traffic of pedestrians and vehicles. A spatial prior in this case is very useful to model the high probability of a car on the road compared to any other part of the image. The pedestrians in the dataset also show a spatial pattern that can be captured in a non-homogeneous spatial Poisson prior. Ground truth for training and testing provided by the dataset itself consisting of 350 training and 100 test images is used to train the model.

**Training: (Shape and Appearance Prior)** A complex shape prior with 16 landmark points shown in Section 4.1.2 is used to model the shape of pedestrians. A rectangular box whose length and width are determined by the outcome of a multinomial distribution with parameters $[\zeta_l, \zeta_w]$ is used for the shape prior on cars.

Pixel intensities of objects from different categories (cars and pedestrians) show huge variation
and modeling them directly is complex. Alternatively, one can model the difference in intensity value of each pixel from the observed mean background value; $e = |B_d - x_d|$, where $B_d$ is the mean background intensity value calculated from the training data. $P(e|\phi_{c,1}, \phi_{c,2})$ given in Equation 7.1, which represents a right truncated exponential distribution (truncated at $\phi_{c,2}$), where $c$ represents the category which is foreground and background in our case. $\alpha = [\alpha_1, \alpha_2] = [\phi_{f,1}, \phi_{f,2}]$ and $\gamma = [\gamma_1, \gamma_2] = [\phi_{b,1}, \phi_{b,2}]$ denote appearance parameters for foreground and background respectively. The hyper-parameter values for the shape and appearance priors are calculated using maximum likelihood on the training set.

$$P(e|\phi_{c,1}, \phi_{c,2}) = \frac{\phi_{c,1} \exp(-e\phi_{c,1})}{(1 - \exp(-\phi_{c,2}\phi_{c,1}))}$$

$$0 < e < \phi_{c,2} \quad (7.1)$$

**Testing:** The dataset was designed for object detection (pedestrians) and does not contain segmentation annotations in the training set. We, therefore, evaluate the results only with respect to
object detection accuracy. Results reported in [151] for different approaches are used for comparison. The approaches that we compare with are: a generic HOG+SVM detector trained on CUHK dataset; on the INRIA [39] and CUHK dataset; on an additional dataset [64]; and a scene-specific pedestrian detector [151]. We refer to the fourth approach that used adaptive context cues as SSPD [151]. Results are reported in Figure 7.4 where we see that our algorithm outperforms competing models for detecting multiple pedestrians and cars in this dataset.

**Implementation Details:** Inference using Gibbs sampling on this model is expensive due to the introduction of the latent variable Z, which is the size of the image. For this reason, we use a hybrid Gibbs sampler that combines a single site and a blocked Gibbs sampler. Blocks of pixels in the image are sampled for the value of Z at first. Blocks that show very low likelihood for the occurrence of an object in it are removed from further sampling. In other words, these are the pixels whose appearance is closer to the background than foreground model. This allows us to get rid of obvious non-object center locations. After this step, the rest of the image is sampled for Z at every pixel location. The algorithm has been implemented in MATLAB and takes an average of
10 minutes per $[1024 \times 1344]$ image on a 12GB RAM, 2.4GHz computer.

Additional benefits of the model include the ability to generate synthetic shapes from the inferred shape prior (Figure 4.5b) and the inference of the posterior distribution of the spatial Poisson process that can show the high stress/traffic areas (i.e., the areas of high probability of occurrence of objects), depicted by the contour plot in Figure 7.4(b). This can be inferred and analyzed for different times of the day as well.
Chapter 8

3D Data Segmentation

Segmenting objects of interest from 3D datasets is a common problem encountered in medical data. Optically subtle changes in intensity and low contrast make the task of segmentation difficult. In this chapter, extension of the proposed model for segmentation of dermal-epidermal junction in 3D reflectance confocal microscopy (RCM) images is presented.

8.1 Reflectance Confocal Microscopy (RCM)

Skin cancer is diagnosed in approximately 3.6 million people in the USA every year. This is more than the total number of all other cancer cases [133, 139]. Standard clinical screening and diagnosis of skin cancer is performed using dermoscopy and visual examination. But, based on clinician expertise and setting, this has a highly variable sensitivity of 70 – 90% and specificity of 32 – 90% in discriminating benign melanocytic lesions (moles) from malignant ones [68]. When an abnormal lesion is suspected, biopsy is carried out which is invasive, painful and leaves a scar. The appropriateness of biopsy referred by clinicians was reported to have sensitivity of only 58%
to 75% [27] and hence benign-to-malignant biopsy ratios vary from 4:1 up to 47:1 [136]. Several optical technologies are being developed to improve detection of melanoma. A vertical section of the histology image is shown in Figure 8.1. The DEJ has the shape of hills and valleys as can be seen from this image.

Reflectance confocal microscopy (RCM), is a non-invasive technique that enables imaging of the human skin to the depth of the papillary dermis or superficial reticular dermis (100 – 150µm) [61]. En-face slices in the form of mosaics (parallel to the surface) are captured sequentially for increasing depths of the skin and recorded as a z-stack of images as shown in Figure 8.2. RCM based clinician diagnosis of basal cell carcinoma can be achieved with sensitivity and specificity of 92 – 88% and 70 – 84% for melanomas and 92 – 100% and 85 – 97% for basal cell carcinoma respectively. Imaging offers significantly better specificity compared to dermoscopy especially for less or non-pigmented lesions [67]. This makes RCM an attractive technology to use for skin cancer diagnosis.
8.1.1 Dermal-Epidermal Junction of Human Skin

The dermal-epidermal junction (DEJ) separates the superficial epidermis from the underlying deeper dermis. It has a 3D structure that looks like multiple hills and valleys as shown in Figure 8.3. The presence and location of dermal-epidermal junction (DEJ) is a key attribute, as the majority of diagnostically important features appear here. Dermal-epidermal junction (DEJ) is identified by expert readers who manually and laboriously read RCM images in sequence through the z-stacks. Mosaics of RCM slices just below and above the junction are then identified and analyzed [11, 60]. Accurate and repeatable localization of the DEJ is important for both clinicians (when examining patients) and pathologists (when examining biopsy sections) because melanocytic lesions and cancers originate at this junction from which they may later spread laterally or invade in depth.

Identifying the DEJ in these RCM images is a difficult task due to subtle optical variations that
are hard to recognize. This results in subjectivity and high variability. The goal of this method is to present an algorithm that automatically detects the DEJ in RCM image z-stacks to assist clinicians in reading images and offer a quantitative aid for noninvasive screening and diagnosis.

8.2 Literature Review

The 3D structure of DEJ is marked by multiple hills and valleys. This structure is flexible and subject to inter and intra patient variability [103]. Variable intensity contrast in RCM images makes the task of automatic DEJ detection challenging. The complexity of the problem varies with skin type – with dark skin type being easier than fair ones because dark skin type shows high contrast due to the presence of melanin. In fair skin, the contrast is relatively low. The current state of the art system [91, 92], applies multiple classification heuristics to tackle this challenging segmentation problem. They applied two different strategies for dealing with dark skin and fair skin types. In both cases, they divided each layer of the RCM stack into disjoint tiles. For dark skin, they applied an approach that finds the peaks in the median intensity profile combined with texture template matching to localize the DEJ. For fair skin, they first grouped consecutive tiles into intervals along the depth direction and then used these tile candidates as input to a locally smooth support vector machine [146] that assumes smoothness in decision among neighboring tiles (in 2D). Here, rather than combine several heuristics to handle this difficult segmentation task, we utilize domain knowledge to design an unsupervised generative model that takes shape structure and appearance into account.

There exists a long and rich literature on general unsupervised methods for segmentation; that includes several variations and combinations of clustering [163], graph cuts [28] and deformable
models [69]. We present a brief review of the relevant work that exists on medical image segmentation followed by the motivation for our model.

General unsupervised methods such as k-means, spectral clustering, and graph cuts can be used to perform segmentation. However, images are structured. Thus, variants of clustering that take into account local information are proposed. Examples include [87, 23], that uses max-flow/min-cut formulation for minimizing the energy in a Markov Random Fields (MRF) framework and [160] that further extends this formulation to 3D using a fast continuous max-flow method (CMF). These models work well for segmentation problems where appearance information is discriminative enough. However, for challenging segmentation problems, such as our DEJ detection problem, where image contrast is poor, appearance information is no longer sufficient.

In difficult segmentation problems, incorporation of the knowledge about the shape of an object combined with image feature information often improves segmentation performance [34, 157, 18]. Many techniques exist in the literature for estimation of shape [41, 73]. Incorporation of shape in 3D medical data is popular in contour evolution formulations that optimize energy functionals [16]. There have also been proposals that incorporate shape priors by combining atlas-based segmentation with active contours [54, 31]. [117] extended the deformable parts based model [50] to 3D object detection and trained the objective function using latent variable structured SVM. However, most formulations are typically designed to segment/detect one object at a time [161, 164, 2]. However, the segmentation problem we are dealing with has a complex flexible shape that does not lend itself to having landmark points or a pre-defined structure, unlike typical anatomical structures. Also, one may be interested in detecting multiple occurrences of similar objects/components, such as the hills and valleys in DEJ.
In this chapter, extension of the model to 3D data for segmentation of dermal-epidermal junction is presented. The model can automatically detect and segment multiple occurrences of similar objects and automatically determine the number and location of these objects. Moreover, the model allows incorporation of domain knowledge information regarding the general shape of objects, yet allowing shape parameter variability among objects. This is a unique and new direction of modeling strategy in the realm of probabilistic models, for segmentation in 3D. Performance of the model is demonstrated for segmenting the DEJ for both dark and fair skin RCM image stacks with an average error of 5.41\(\mu\)m and 6.55\(\mu\)m respectively. Note that only one approach is utilized unlike previous solutions that needed different approaches based on skin type.

### 8.3 Model Formulation

Stacks of images of human skin *in vivo* are collected with a reflectance confocal microscope. Each stack from a single skin site consists of 45 – 70 image slices, where each slice is 1000 \(\times\) 1000 pixels, with a pixel resolution of 0.5\(\mu\)m. The z step (depth) resolution is 1\(\mu\)m, although the true optical sectioning thickness of the imaging system is 3\(\mu\)m.

Let the 3D RCM image be represented by a data tensor \(X \in \mathbb{R}^{D_r \times D_c \times D_z}\), where \(D_r\), \(D_c\) and \(D_z\) are the number of rows, columns and slices of the image respectively. Each element in this tensor is a voxel, represented by a scalar (e.g., gray scale intensity value) or a vector (e.g., texture features). Let the total number of voxels in the image be denoted by \(D\).

The notation introduced in Chapter 4 is used to formulate the model for 3D data. The number of points generated from the Poisson process be denoted by \(N\) and the location of these \(M\) points be denoted by \((L = [l_1, \ldots, l_M])\). Let the intensity parameter of the Poisson process be denoted
by $\lambda(\tau)$, where $\tau$ represents the 3D space of the RCM stack. For each location, shape variables $(S = [s_1, \ldots, s_M])$ are drawn from a flexible shape prior model (a probabilistic model of hills in this case). One random sample from the shape prior is shown in Figure 8.5(a). The hills (shape) are marks associated with the outcome of a spatial Poisson process and hence the model is called a marked spatial Poisson process. Because hills (shape) are unknown, the marks are latent and thus we have a latent marked Poisson process. One random outcome of a spatial Poisson point process in combination with random realization of the shape model is shown in Figure 8.5(b). Given the location and shape of each hill defining a boundary (DEJ): voxels above the boundary are drawn from an epidermis (background) appearance model and voxels below the boundary are drawn from a dermis (foreground) appearance model. Note that $N$ is random and open-ended making it possible to generate a flexible model of the DEJ. The generative process as follows:

1. Generate $N$ points along with their location from a spatial Poisson Process.

2. For each object, given its location, generate object boundaries from the shape model.

3. Given the location of objects and its shape boundary, data below the object region (dermis) are generated from an object feature/appearance model and data above the objects (epidermis) are generated from a background model.

A pictorial representation of this process is shown in Figure 8.4 In the following subsections, each of the three components of the model is described in detail.

### 8.3.1 Spatial Poisson Process

For this data, $\tau$ represents a 3D volume where a sample from a spatial Poisson process consists of a random number of points at random locations in the 3D volume, based on an underlying
intensity function $\lambda(\tau)$. The dermal-epidermal junction starts in the range $50 - 100\mu m$, which is where we expect to see the hill peaks. The functional form of $\lambda(\tau)$ should be such that it gives a higher probability to the RCM slices in this range (which is approximately 25-50th slice in depth). Therefore, we choose to use the shape of a Gamma distribution ($\lambda(\tau_z) \sim \Gamma(10, 3)$) with values 10, 3 which correspond to a shape which gives maximum probability to slices in this range. Alternately, one can choose any other shape that can capture this domain knowledge.

Since we do not have prior knowledge about the regions in each slice where DEJ has high/low probability of occurrence of hill peaks, we choose a uniform Poisson intensity prior in the $x - y$ plane. This implies that two voxels belonging to different layers will have different Poisson
intensity values while the voxels belonging to the same layer have a uniform value. The integral intensity value $\sum_{\tau} \lambda$ for the entire 3D volume is assumed to be 50. This is an approximate value chosen using domain knowledge [103] and the fact that the thickness of a basal cell (cells close to the DEJ) is approximately 10 microns. Figure 8.6 shows one random outcome of the Poisson process with the chosen priors.

![Figure 8.6: Intensity along the z-axis varies according to $\Gamma(10, 3)$. Uniform intensity prior across the x-y plane is assumed with the integral Poisson intensity value $\sum_{\tau} \lambda = 50$.](image)

### 8.3.2 Shape Model

Each location (determined from the outcome of a Poisson process) is associated with a shape. In this case, the dermal-epidermal junction (DEJ) is composed of hills and valleys (see Figure 8.3). One can model the shape prior to be a hill. The intersection of the various slopes from these hills constitutes the valleys of DEJ. One could alternatively model the DEJ as a combination of valleys (inverted hills) too. However, the dip of these inverted hills (valleys) is not always present in the RCM stacks, as the depth of the stack is sometimes not enough to include the dip. This makes
modeling and inference harder and hence the choice of modeling the hills. The height, orientation and size of the hill is variable and this has to be captured in the shape prior of the model.

On the \textit{en-face} \((x - y)\) plane, the hill looks like an ellipse. The major and minor axis are variable and also change as a function of depth (increase with depth in \(z\)-direction). We propose the following 3D shape model for each hill to capture these properties, where each hill peak location is denoted by \([l_x, l_y, l_z]\) (which is the outcome of a Poisson process). An ellipse has a center \((\{l_x, l_y, l_z\})\), a major axis \((\rho_{ma})\), a minor axis \((\rho_{mi})\), and an orientation \((\rho_o)\). Let \(\rho_{ma,o}, \rho_{mi,o}\) represent the major and minor axis value at the peak of the hill. As we go down the hill along the \(z\)-stack, the major and minor axes parameters increase as a function of depth. Mathematically, this increase is captured independently for minor and major axis by introducing two additional parameters \(w_{ma}\) and \(w_{mi}\).

The value of major \((\rho_{ma,z})\) and minor \((\rho_{mi,z})\) axis at each depth \(z\) of the RCM stack, where the center of hill peak is \([l_x, l_y, l_z]\), is given by Equation 8.1 as a function of these two parameters \((w_{ma}, w_{mi})\).

\[
\rho_{ma,z} = \rho_{ma,o} + \log((l_z - z)^{w_{ma}}), 1 \leq z \leq l_z
\]  

\[
\rho_{mi,z} = \rho_{mi,o} + \log((l_z - z)^{w_{mi}}), 1 \leq z \leq l_z
\]  

Let us represent all the shape parameters of each hill by \(s = [\rho_{ma,o}, \rho_{mi,o}, \rho_o, w_{ma}, w_{mi}]\). After the parametric form of the shape prior has been determined, one needs to determine the distribution of these parameters that can capture the prior knowledge of all possible shapes.

In this data, the approximate thickness of a basal cell is 10\(\mu m\). Basal cells correspond to the basal layer which is the innermost layer of the epidermis, present just above the dermis. We assume a Gamma distribution with the hyper-prior values 5 and 5 as this ensures that the highest probability is given to the value 10 and the parameter is always a positive value. Alternately, any
other distribution that can capture these properties can be used.

Parameters of the shape model, minor ($\rho_{mi}$) and major axis ($\rho_{ma}$) are assumed to be generated from a Gamma distribution.

$$
\rho_{ma} \sim \Gamma(\lambda_{ma}, \eta_{ma}) \quad (8.2)
$$

$$
\rho_{mi} \sim \Gamma(\lambda_{mi}, \eta_{mi})
$$

There is no obvious distribution of orientations and rate of increase of hills. Orientation ($\rho_{o}$) and rate of increase of major and minor axes ($w_{ma}, w_{mi}$) are, therefore, assumed to be generated from a uniform distribution.

$$
\rho_{o} \sim U[0, \pi] \quad (8.3)
$$

$$
w_{ma} \sim U[w_{min}, w_{max}]
$$

$$
w_{mi} \sim U[w_{min}, w_{max}]
$$

Thus, each hill can vary in shape, size and orientation. An example random sample drawn from one hill shape model is shown in Figure 8.5(a). Note that the shape of DEJ is modeled as a combination of several hills. This is consistent with the natural structure of DEJ. Since the number of such hills and their location is known a priori, we model it using the spatial Poisson process. This is a novel unique way to model the shape that places no restrictions on the possible combinations and hence is flexible enough to encompass the true shape of the DEJ. The number of such hills present in any RCM stack will grow naturally with the size of the stack (en-face area)
and the upper limit on the number of possible hill peaks is captured in a probabilistic framework based on domain knowledge. An example random sample from the latent shape marked spatial Poisson point process, where each hill is the hidden mark is shown in Figure 8.5(b).

### 8.3.3 Appearance Model

An appearance model captures the likelihood of a given voxel/pixel to belong to any given category. One can either use the intensity values directly or higher level features such as texture to calculate this likelihood value. In the experiments, appearance parameters are determined adaptively for every image stack as inference is performed in the unsupervised setting. For this, we first extract higher level features from the RCM voxel intensity values and use them to infer the appearance parameters. This is done because fair skin images have lower intensity contrast than dark skin, making detection difficult. Clinicians rely on the variation in texture of the voxels corresponding to dermis and epidermis to locate DEJ visually. The texture patterns in fair skin require information from an area of at least $25\mu m \times 25\mu m$ to capture the texture information at each site. Thus, a sliding window size of 50 is used for feature extraction. We extract histogram, co-occurrence, wavelet and Gabor features as they are known to be good representatives of texture. This results in a feature vector of size 148 for each voxel.

In order to determine the appearance parameters, we perform K-means clustering on the voxels from the bottom five slices. The reason for this choice is that slices at the bottom have both dermis and epidermis voxels. Therefore, we obtain one cluster which belongs to dermis and the other that belongs to epidermis. The cluster whose features are close (in terms of Euclidean distance) to the features in the top five slices is identified as the class belonging to epidermis since the top slices
are known to contain only epidermis voxels. Once the clusters have been identified, parameters of the appearance model are calculated adaptively based on the feature values of each cluster. As mentioned earlier, the RCM voxels are characterized by noise and variations, which make the appearance based segmentation difficult. In this case, even though k-means clustering is used on the bottom slices to calculate the appearance parameters adaptively, a combination of the powerful shape model along with the spatial Poisson process make the model robust to noise in the raw RCM voxels.

However, 148 is a high value for the number of features, and all of them might not be discriminative enough. To reduce the dimension of the feature space, we employ a simple technique. Features whose standard deviation is more than three times the distance between their means, are removed as they are not considered to contain enough information to distinguish dermis and epidermis voxels. Since this is an unsupervised method, it results in different number of features for different stacks.

In dark skin, because there is better contrast, intensity features were sufficient for differentiating dermis from epidermis voxels. Again, K-means clustering is employed on the intensity values of the voxels adaptively for every stack.

Let $x_n$ represent the feature vector of each voxel. We assume the probability of $x_n$ above the hill (i.e., epidermis) be drawn from a Gaussian distribution with mean $\mu_b$ and variance $\sigma_b^2$ and the probability of voxels $x_n$ below (inside) the hill (i.e., dermis) be drawn from a Gaussian with mean $\mu_f$ and variance $\sigma_f^2$. Let all the appearance parameters be denoted by $\alpha_f = [\mu_f, \sigma_f]$ and $\alpha_b = [\mu_b, \sigma_b]$. Alternately, one could assume a different distribution that can capture the likelihood of the voxels to belong to each category.

Once the location and shape of each hill is known, a boundary/partitioning of the 3D space is
CHAPTER 8. 3D DATA SEGMENTATION

defined. Let all the appearance parameters be denoted by \( \alpha = [\alpha_f, \alpha_b] \).

\[
p(X|N, L, S, \alpha) = \prod_{n=1}^{N} [p(x_n|\alpha_f)]^{I_n} [p(x_n|\alpha_b)]^{1-I_n}
\]

(8.4)

where, likelihood of the image given all the parameters is given by Equation 9.2. \( I_n \) is a random variable that takes a value 1 when the pixel \( n \) is below/inside the contour and 0 otherwise. Note that this random variable is a deterministic function of \( L \) and \( S \).

**Overall Model:** Recall that we denoted the object (hill) locations by \( L = [l_x, l_y, l_z] \). Let all the \( N \) objects be represented by \( L = [l_1, \ldots, l_M] \) and shape parameters for all the objects by \( S = [s_1, \ldots, s_N] \). Let the hyper-parameters of the shape prior distribution be denoted by \( \zeta \) and \( \theta = [\lambda(\tau), \zeta, \alpha] \). The joint likelihood of the model is given by Equation 9.1.

\[
p(N, L, S, X|\theta) = p(N, L|\lambda(\tau))p(S|N, L, \zeta)p(X|N, L, S, \alpha)
\]

(8.5)

Inference on the model is accomplished using Gibbs sampling framework described in Chapter 5.

**8.4 Experiments and Results**

We validate our proposed model, MPP-skin, on nine stacks of dark skin and four stacks of fair skin RCM images for which we have ground truth (expert labeling) of the DEJ. Performance of the algorithm is measured in terms of the error in localization of the dermal-epidermal junction
(DEJ) along the z-axis for each voxel compared to ground-truth. Ground-truth annotations were
given by three experts, who discussed and provided one label for each stack. We report the mean
and variance of the distance between DEJ detected by our algorithm and the expert labels. We
compare the proposed MPP-skin algorithm against the current state of the art approach on this
problem [93, 92] (Other) and a general 3D segmentation approach based on continuous max-
flow [160] (CMF). A description of these methods is provided in the following paragraphs.

Other: For dark skin type, [93] divides each en-face (x-y) plane into tiles of 16µm × 16µm
resulting in a stack of en-face image tiles. This width corresponds to the width of 1 – 2 basal cells.
In dark skin, basal cells show high contrast. Median intensity for the stack of tiles is computed
as a function of depth followed by Gaussian filtering. Peaks in intensity along the depth of the
calculated intensity profile are detected as basal cells. For profiles having multiple peaks, tem-
plates of texture extracted from profiles with a single peak, are used for comparison and the most
similar peak is detected as the basal layer. In fair skin, basal cells do not exhibit contrast due to
lack of melanin. Each stack is divided into disjoint tiles of 25µm × 25µm as patterns of texture
that distinguish epidermis from dermis are much larger than the basal cell thickness. Tiles that
are similar in texture are grouped into intervals along the depth. Dynamic programming [35] is
used to determine the number and location of such groupings. This grouping is passed through
a locally smooth support vector machine (LV-SVM) [146] as it takes into account smoothness in
decision among neighboring tiles. This method is designed specifically to solve the problem of
DEJ localization and contains a combination of heuristics.

CMF: Segmentation problems can be cast in terms of energy minimization in the min-cut sce-
nario, where the optimal cut in the graph is solved using graphcuts. Each pixel in the image cor-
responds to a node in the graph and smoothness is incorporated between neighboring pixels using
a boundary term that penalizes dissimilar neighbors [87]. [160] reformulate this as a continuous
min-cut problem resulting in a global binary solution to the original convex partition problem. This
has been extended to the 3D scenario, and we utilize the code provided by the authors to determine
the DEJ in the RCM stacks. This is an unsupervised method designed for segmentation of objects
in 3D.

Table 10.2 report the results of CMF [160], Other [93] and our proposed algorithm, MPP-skin,
for dark skin RCM image stacks and Table 10.3 present the results on the fair skin RCM image
stacks. Results from Tables 10.2 and 10.3 show that the proposed MPP-skin outperforms the other
methods. Note that CMF performs very poorly as it is unable to capture the shape constraints
of the DEJ. [93] designed a supervised algorithm that combined several heuristics to address this
problem specifically, yet our proposed unsupervised generative model MPP-skin is able to perform
as well as, and on average better than Other [93]. [93] does not capture the smoothness/shape
constraints across the z-axis, whereas our model incorporates the shape and smoothness naturally
in our formulation. Also, the algorithm addresses the dark skin type and light skin type stacks with
different algorithms.

Figure 8.9 shows an example dermis boundary (a) marked by an expert and (b) automatically
detected by our algorithm, MPP-skin on an en-face slice of image stack 1. The results show a
reasonable match with expert labels. In addition, the shape of the DEJ in 2D is captured very well
by the proposed method as it explicitly takes into account the general shape of a DEJ, resulting in
a smooth contour.

Figure 8.8 shows a surface 3D plot of the DEJ detected by our algorithm on a sample dark skin
stack. Color indicates how far the DEJ is from the actual value. Blue indicates a low error and
red indicates high error between the expert labels and the results of the proposed algorithm. From
CHAPTER 8. 3D DATA SEGMENTATION

Figure 8.7: Dermal-Epidermal junction detected by our algorithm for dark skin type. Color indicates the error in DEJ detection compares to expert labels.

Figure 8.8: Dermal-Epidermal junction detected by our algorithm for fair skin type. Color indicates the error in DEJ detection compares to expert labels.
these figures, it can be seen that the color map of our detected DEJ is mostly blue indicating that the DEJ is very close to the DEJ marked by an expert. The values in Table 10.2 and 10.3 also confirm that our algorithm performs very well in detecting the DEJ. Our method performs well because we incorporate both shape and appearance into our model; whereas, [93] and CMF only consider appearance. Furthermore, our approach also takes uncertainty in location into account.

Each stack is a 3D image with 45-75 image slices, where each slice has 1000x1000 pixels. Thus, the average and standard deviation are over 1000x1000 items and the error is the error along the z(depth) direction from the manually labeled boundary where the true optical sectioning thickness of the imaging system is 3 microns (resolution). The thickness of a basal cell (cells close to the DEJ) is approximately 10 microns. Hence, error performance of 5-6 microns is quite good, and error bars of 5 microns is still good (within 10 microns).

Our ‘MPP-skin’ offers several advantages over ‘other’. First, ‘other’ actually needs two separate methods, one to deal with dark skin and the other for fair skin because these show very different appearance patterns. However, MPP-skin only needs one model to address these skin
Table 8.1: Evaluation of the Performance of the Proposed Algorithm for Dark Skin Type.

<table>
<thead>
<tr>
<th>Test set</th>
<th>CMF ↓</th>
<th>Other (block) ↓</th>
<th>MPP-skin (Ours) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack 1</td>
<td>17.71 ± 12.3</td>
<td>3.15 ± 3.26</td>
<td>3.6 ± 3.4</td>
</tr>
<tr>
<td>Stack 2</td>
<td>3.6 ± 14.0</td>
<td>7.13 ± 6.36</td>
<td>10.1 ± 5.3</td>
</tr>
<tr>
<td>Stack 3</td>
<td>8.7 ± 11.1</td>
<td>5.89 ± 4.58</td>
<td>4.2 ± 3.1</td>
</tr>
<tr>
<td>Stack 4</td>
<td>47.6 ± 5.3</td>
<td>3.49 ± 3.28</td>
<td>6.0 ± 5.5</td>
</tr>
<tr>
<td>Stack 5</td>
<td>6.7 ± 7.9</td>
<td>14.45 ± 7.45</td>
<td>4.9 ± 3.6</td>
</tr>
<tr>
<td>Stack 6</td>
<td>34.18 ± 6.2</td>
<td>2.99 ± 2.65</td>
<td>2.1 ± 3.0</td>
</tr>
<tr>
<td>Stack 7</td>
<td>22.3 ± 14.5</td>
<td>7.73 ± 6.7</td>
<td>6.0 ± 5.4</td>
</tr>
<tr>
<td>Stack 8</td>
<td>17.4 ± 8.1</td>
<td>7.04 ± 5.98</td>
<td>7.0 ± 5.3</td>
</tr>
<tr>
<td>Stack 9</td>
<td>18 ± 13</td>
<td>9.95 ± 7.61</td>
<td>4.8 ± 4.8</td>
</tr>
<tr>
<td>Average</td>
<td>19.5</td>
<td>6.86</td>
<td><strong>5.41</strong></td>
</tr>
</tbody>
</table>

Table 8.2: Evaluation of the Performance of the Proposed Algorithm for Fair Skin Type.

<table>
<thead>
<tr>
<th>Test set</th>
<th>CMF ↓</th>
<th>Other (block) ↓</th>
<th>MPP-skin (Ours) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack 1</td>
<td>20.94 ± 13.62</td>
<td>13.65 ± 7.34</td>
<td>6.13 ± 5.65</td>
</tr>
<tr>
<td>Stack 2</td>
<td>16.08 ± 3.81</td>
<td>9.62 ± 7.92</td>
<td>4.15 ± 3.37</td>
</tr>
<tr>
<td>Stack 3</td>
<td>9.3 ± 7.7</td>
<td>2.24 ± 1.77</td>
<td>5.41 ± 3.33</td>
</tr>
<tr>
<td>Stack 4</td>
<td>17.5 ± 9.9</td>
<td>4.83 ± 4.79</td>
<td>4.31 ± 5.14</td>
</tr>
<tr>
<td>Average</td>
<td>15.93</td>
<td>7.14</td>
<td><strong>6.55</strong></td>
</tr>
</tbody>
</table>
types. Second, ‘other’ involves several heuristics to make it work on this challenging problem; whereas, ‘MPP-skin’ offers a natural model of images. Third, ‘other’ is a supervised method (involving several SVM components for the fair skin model); whereas, ‘MPP-skin’ is unsupervised despite which it was able to obtain lower errors. The advantage of having an unsupervised method perform better is that, there is greater hope of being able to adapt to other patient image stacks that will be obtained in the future for this highly variable and challenging problem.

Segmenting objects of interest in medical data is often challenging due to high intra and inter patient variability, and the complexity of the boundary structure. Incorporation of shape information about the object of interest can often help improve segmentation performance. In addition, the detection problem usually involves locating multiple occurrences of similar objects. Here, we address the problem of automatically detecting the dermal-epidermal junction in 3D reflectance confocal microscopy images of human skin. RCM imaging allows clinicians to view nuclear detail in the skin without resorting to biopsy. Moreover, detection of the DEJ is important because this is where skin cancer usually starts. DEJ detection in RCM images is challenging due to low contrast and the DEJ structure is highly variable and complex. However, we noticed that the DEJ is a 3D structure consisting of multiple hills and valleys. We proposed an unsupervised probabilistic model that incorporates shape and appearance using a marked spatial Poisson process. Inference on the algorithm is accomplished using Gibbs sampling and experiments are presented on two skin types: dark and fair. We show that this model is capable of detecting DEJ for both skin types with an average error of 5.41µm and 6.55µm for dark and fair respectively, compared to expert labeling.
Chapter 9

Time-Series Data

In this chapter, we extend the model proposed in Chapter 4 to 1D (time-series) data. Experimental results are provided and compared with the state of the art techniques.

9.1 Introduction

Identifying subsequences from a time-series dataset has several applications such as discovering stocks with similar price patterns, finding customers with similar tastes in products or any characteristic signature/shape associated with the activity of interest. We first present a brief literature review, followed by the model and inference.

A similarity measure for comparing two sequences along with fast search techniques was proposed in [9] and matching sub-sequences after transforming them into the frequency domain was proposed in [107]. A powerful way to retrieve basic shapes was proposed in [10], which can detect blurry shapes as it only cares about the overall shape which is captured by a set of user specified rules. A more sophisticated approach popularly known as 1−NN with dynamic time warping
(DTW), is a technique that does non-linear mapping of one signal on another to minimize the distance between them. The mapping can be done for multiple window sizes and is the current state of the art in this domain [124] for subsequence classification. Experimental results using this technique with a sliding window is provided in Section 9.4. Another interesting facet of time-series analysis deals with the detection of sub-sequence of interest from an observation. Discovering shapelets in a given dataset that would be most discriminatory of classes was proposed in [158]. An algorithm to speed up the discovery of shapelets was proposed in [84]. Once the shapelets are discovered, they would be tested on new data for their discriminatory power of various classes which is again, sub-sequence matching.

We propose a probabilistic generative model that can identify subsequences at different scales and lengths. For example, consider Figure 9.1(a) and Figure 9.1(b). Both have a similar shape, but different length and scale. We refer to the amplitude as scale. Our model is capable of identifying multiple occurrences of sequences of different scales and lengths in a probabilistic fashion.

![Figure 9.1: Two sample sub-sequences.](image-url)
9.2 Model Formulation

Let the time series sequence containing the subsequences of our interest be denoted by \( t = [t_1, \ldots, t_D] \).

The generative model can be described as follows. Let the locations of the starting points of the sub-sequence of interest be denoted by variables \( l = [l_1, \ldots, l_N] \), where \( N \) is the total number of points. We assume that these position variables are generated from a Poisson process. We then sample the shape \( S = [s_1, \ldots, s_N] \) of the sub-sequence from our shape prior. The shape prior is flexible in length and scale. Once the shape of all the subsequences is generated, values of the rest of the sequence is generated from the probability distribution of background. Let the parameters of Poisson intensity, shape and background be denoted by \( \lambda(t) \), \( \zeta \) and \( \gamma \) respectively. Let all the hyper-parameters be represented by \( \theta \).

\[
p(l, S, t|\theta) = \left[ p(l|\lambda(t))p(S_{1:N}|l, \zeta) \right] \times p(t|l, S, \gamma) \tag{9.1}
\]

\[
p(t|l, S, \theta) = \left[ \prod_{n=1}^{N} p(t_{n:(l_n+s_n)}, l, S) \right] \prod_{d=1}^{D} [p(t_d|\gamma)]^{1-I_d} \tag{9.2}
\]

where, \( I_d \) is a binary value that is equal to one when it falls inside a shape in the sequence.

9.2.1 Complex Shape Model

We propose a complex shape model for the shape of the subsequence using a Gaussian process. In a supervised setting, parameters of the Gaussian process can be determined from training examples.

We introduce a random variable \( \sigma_s \) to normalize the amplitude (e.g., to a value between 0 and 1).

This parameter also has a Gaussian hyper-prior whose value can be determined during training. We introduce another random variable \( \sigma_l \) to incorporate the uncertainty in length of the subsequence.
This random variable is also drawn from a Gaussian distribution and represents the value of length of the sequence. Any sub-sequence, irrespective of the original length, can be normalized to a fixed length (e.g., length = 10). This can be done in a systematic way keeping in mind the Nyquist criterion to ensure that there is no loss of information. Two random variables $\sigma_s, \sigma_l$ are introduced to normalize the scale and size of the sequence before calculating its likelihood to belong to a shape. The shape prior in this case consists of $s = [\mu, \Sigma, \sigma_s, \sigma_l]$, where $\mu$ represents the mean value at a fixed number of locations, $\Sigma$ indicates the covariance matrix. The hyper-priors on these shape parameters are represented by $\zeta$.

### 9.3 Inference

Inference involves calculating the posterior over the parameters $L$ and $S$. The pseudo-code for the algorithm is shown in Table 1. For a homogeneous Poisson process with intensity parameter $\lambda$, the probability of occurrence of a subsequence at $d$ after occurrence of the subsequence at the previous location $l_{n-1}$ is given by $(\lambda \times (d - l_{n-1})^3)^{\frac{e^{-\lambda d} - e^{-\lambda l_{n-1}}}{1}}$. The probability that no subsequence occurs within the interval $d - l_{n-1}$ is given by $e^{-\lambda (d - l_{n-1})}$. We use this property to calculate the probability of occurrence of a subsequence at any location in the time series. We also sample from all possible shapes at each subsequence location which gives us the posterior of shape prior as shown in Step 2 of the algorithm in Table 1.

One might have to detect sub-sequences of different categories from a given time series. The above algorithm can be extended to detect different categories of sub-sequences. Let the number of categories be denoted by $C$. Let the locations and shapes of the subsequence are denoted by $I_c$ and $S_c$, where $c$ indicates category. The new formulation is given by Equation 9.3. Let the
Algorithm 2 Gibbs sampling for sub-sequence discovery in a time-series

**Initial:** Initialize the number of subsequences present in the time-series being analyzed to zero, \( l = \{\}; S = \{\}; N = 0; m = 1 \) and start from the first point in the time-series. Let the iteration number \( i = 1 \)

**Step 1:** For \( d = 1 \ldots D \). Calculate the probability of occurrence of a sub-sequence at the location \( n \)
\[
p(l_n = t_d|\cdot) \propto (\lambda \times (d - (l_{n-1} + s_{n-1}))) e^{-\lambda \times (d - (l_{n-1} + s_{n-1}))} \int p(X|\cdot)p(s_n|\zeta)ds
\]
\[
p(l_n > t_d|\cdot) \propto (\lambda \times (d - (l_{n-1} + s_{n-1}))) e^{-\lambda \times (d - (l_{n-1} + s_{n-1}))} p(X|\cdot)
\]
Sample from the above distribution to determine the value of \( l_n \).
If \( l_n = d \), \( N=n, n=n+1 \).

**Step 2:** For \( n = 1, \ldots, N \)
Sample the shape parameter of each object from the following distribution
\[
p(s_n) \propto p(s_n|\zeta)p(X|\cdot)
\]

**Step 3:** \( i = i + 1, l_i = l \) and \( S_i = S \)

**REPEAT** steps 1, 2 and 3 until convergence.

**Output:** Mean of the states where first order statistics vary below a pre-set threshold \( \epsilon \): all object’s location, shape.

Corresponding shape priors for each category be given by \( \zeta_c \).

\[
p(t|l, S, \theta) = \left[ \prod_{n=1}^{N} \sum_{c=1}^{C} (p(t_{n,(l_n+s_n)})|l_c, S_c)^{\delta_c} \right] \prod_{d=1}^{D} [p(t_d|\gamma)]^{1-l_d} \tag{9.3}
\]

where, \( \delta_c \) is equal to one if the subsequence belongs to the class \( c \) and zero otherwise.

### 9.4 Experiments

Several time-series datasets are made publicly available at [152, 83]. We use the sub-sequences in the dataset ‘CBS’ which consists of 3 classes and 900 test samples. However, these are processed datasets that contain only one subsequence in each test and train sample provided. To test the accuracy of the developed model, these test sequences are concatenated at random points in background noise resulting in a single time sequence of length 20000. Figure 9.2 shows a sample from the test sequence and Figure 9.3 shows the sub-sequences of interest in the data.
We also perform experiments using $1 - NN$ method. This method classifies a given sequence to its corresponding class based on euclidean distance between it and all the training samples. It does not need specification of parameters and is popularly known as the nearest neighbor algorithm. It is a simple, yet powerful method to classify sub-sequences and is used as a baseline for comparison. Results from $1 - NN$ with dynamic time warping (DTW) are also reported. This technique maps a given signal to the test signal to calculate the nearest distance. This is a popular technique known for its capability to perform non-linear mapping.

Precision and Recall values are reported in Table 9.1. A sub-sequence is considered detected, if the algorithm has at least 50% overlap with it and classifies it to the correct class. A sliding window is used for $1 - NN$ and $1 - NN$ with DWT approaches. Note that both the methods have a poor overall performance compared to our method.

Table 9.1: Overall Precision Recall values for sub-sequence matching.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$-NN (Euclidean)</td>
<td>0.74</td>
<td>0.83</td>
</tr>
<tr>
<td>$1$-NN with DTW</td>
<td>0.45</td>
<td>0.69</td>
</tr>
<tr>
<td>Ours</td>
<td>0.92</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table 9.2: Precision Recall values for sub-sequence matching of each class.

<table>
<thead>
<tr>
<th>Method</th>
<th>Class-1 Precision</th>
<th>Class-1 Recall</th>
<th>Class-2 Precision</th>
<th>Class-2 Recall</th>
<th>Class-2 Precision</th>
<th>Class-2 Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$-NN (Euclidean)</td>
<td>0.58</td>
<td>0.84</td>
<td>0.76</td>
<td>0.97</td>
<td>0.9</td>
<td>0.68</td>
</tr>
<tr>
<td>$1$-NN with DTW</td>
<td>0.28</td>
<td>0.71</td>
<td>0.70</td>
<td>0.97</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.93</strong></td>
<td>0.42</td>
<td><strong>0.86</strong></td>
<td><strong>0.97</strong></td>
<td><strong>0.97</strong></td>
<td><strong>0.84</strong></td>
</tr>
</tbody>
</table>

The strength of the proposed method lies in its capability to detect multiple sub-sequences at various scales and sizes in a given time sequence. However, it cannot incorporate the deformation
Figure 9.2: A part of the test sequence is plotted. Red indicates the sub-sequence of interest and blue indicates background.

Figure 9.3: Plots showing 3 classes of dataset ‘CBF’.
of shape such that a part of it is stretched while the rest of it remains the same. In the future, this non-linear variation can be incorporated into the shape model as an additional parameter.
Chapter 10

VOTERS Project

Statistics suggest Federal Government spends about 50 billion dollars each year on roadways, a considerable portion of which goes to road rehabilitation and maintenance [5]. Additionally, approximately one third of all U.S. traffic fatalities are directly associated with poor pavement conditions [6]. Frequent road maintenance and preservation operations are required to ensure the fulfillment of roads socioeconomic role. If road defects are inspected at an early stage, it results in huge savings (of at least 5 times less money) [1]. Current pavement evaluation methodologies, however, are usually not capable of seizing this opportunity. Consequently, these methods often face challenges such as intrusive data gathering (e.g. stopping traffic), manual effort and subsequently infrequent data collection and limited coverage. As a result, these inspections are performed years apart which precludes timely repairs and rehabilitation of the pavements resulting in extravagant amounts of wasted time and money. Furthermore, obtaining the life cycle of the time-varying behavior of the civil infrastructure is out of reach for most agencies due to the difficulties in monitoring the pavement status. Proper assessment of maintenance operations and priorities need to be armed with an efficient road inspection method that is non-intrusive (e.g. does...
not require traffic blockage), fast, and requires minimum manual effort. This also enables regular monitoring. Such an inspection method has been proposed by the VOTERS project.

The Versatile Onboard Traffic Embedded Roaming Sensors (VOTERS) project developed a multi-modal mobile sensor system capable of collecting pavement related information at traffic speed. This system consists of acoustic, optical, electromagnetic, and GPS sensors mounted on a vehicle [149, 130]. To differentiate crack types and their severity, a prominent road distress contributing to pavement deterioration [131, 156], a camera has been mounted on the rear of the VOTERS prototype. An automated algorithm capable of carrying out such analysis is presented.

This section focuses on surface crack detection using information from the video camera mounted on the rear of the vehicle. The detection of surface defects from the video frames and their subsequent classification are the two main purposes of this sensor. However, the average amount of data collected by the camera is 7GB/hour. Clearly, it is a huge number and manual inspection of individual images is time consuming and expensive. Automation of crack detection from images collected and subsequent classification would result in huge savings of time and money.

### 10.1 Camera System (SVAD)

The video collection system consists of a Prosilica GT1920C color camera with a resolution of $1936 \times 1456$ and maximum frame rate of 40 fps. An 8mm lens is used with the 2/3 inch sensor. The camera can be triggered to take images in several modes. This is an essential feature to capacitate an adaptive control on the rate at which images are taken based on the speed of the van. Such control facilitates recording images with constant overlap between them and efficient use of disk space.
Table 10.1: Specification for acquisition software of Camera

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
<td>Auto</td>
</tr>
<tr>
<td>Exposure</td>
<td>300</td>
</tr>
<tr>
<td>Acquisition Mode</td>
<td>Freerun, Continuous</td>
</tr>
<tr>
<td>JPEG Compression</td>
<td>95 %</td>
</tr>
</tbody>
</table>

The PvAPI Software Development Kit (SDK) for Linux version 1.26 on CentOS operating system provided by Allied Vision Technology (AVT) is modified and integrated into the VOTERS central system. The software can be found at http://www.alliedvisiontec.com/us/products/legacy.html. This choice of system enables integration into a centralized framework with other sensors in the van. Details of the system architecture can be found in [162].

Table 10.1 shows the specifications used for acquisition of images by the camera. The camera adjusts the gain value automatically depending on the amount of light available. This enables proper registration of images in different lighting conditions without user intervention.

A trigger wheel that is mounted on the van reports distance traveled by the van every 10 milliseconds, to a central system on the van. Camera sensor retrieves this information and calculates the intervals at which camera should register the image to hard-disk. The camera is set to 'free run' mode, in which the camera triggers images at highest rate possible for the given exposure and saves them in a memory buffer. The buffer also contains information of the time stamp associated with each image. Based on the time stamp and trigger wheel information, distance traveled by the van between any two images is calculated. Based on the minimum distance required for overlap of consecutive images, the frame/image is either dropped or saved to the hard disk. The target minimum distance between any two frames recorded in this case is 900mm. This implies that the camera saves at-least one image for every 900mm traveled by the van. This software is
implemented in C++ on a Linux platform. A JPEG compression with 95% quality is performed before saving a frame to the disk. This results in a significant saving of memory and has proved to have no impact on the processing quality.

Images are collected on a hard-drive and processed after the van finishes survey. Approximately 55,000 images are collected per hour while the van drives with an average speed of 50 km/h.

10.2 Literature Review

Many attempts were made in the past by researchers to come up with an automated system for detection of cracks. The following paragraphs present a brief review of related literature. Wang et al. proposed stereo-vision to establish a 3D surface model using two cameras [148, 147]. Cracks are detected from the geometric modeling, based on vertical variations on the pavement surface. While this method provides a decent detection capability, it requires a lot of computation power and storage for the huge amounts of data and overlap required. VOTERS mode of acquisition on the other hand is very simple, requires a single camera and much less processing power.

Crack detection primarily based on edge detection techniques was attempted in [101]. A comparison and effectiveness of four crack detection techniques: Fast Haar Transform, Fast Fourier Transform, Sobel and Canny was discussed in [7]. Many techniques [95, 140] focus on edge detection and thresholding for background separation using methods like wavelet transform followed by post-processing steps that use neural networks, mutual information, or erosion operator. While these methods are usually successful in discovering the cracks, their detection typically contains considerable number of false positives. This is mainly because of their focus on edges, which is not the only characteristic defining cracks.
Numerous studies [111, 78, 122, 138, 154, 159] have suggested classifying a block of pixels rather than each pixel to the crack/no-crack category. A typical drawback of these methods lies within the decreasing resolution. Cracks are identified in the image as blocks rather than pixels. Hough transform was used to establish connectivity in [29, 55, 121]. Some researchers suggest that the Hough transform would aid eliminating noisy detections by discarding the cracks that do not lie on a straight line. This has the drawback of eliminating true positives as all cracks might not lie on a straight line.

Circularly symmetric binary pattern to determine if the center pixel belongs to a crack was proposed in [74]. In this method, a pattern number is given to each pixel based on the classification. Pixels that exhibit huge transition from their neighbors are discarded. Blob-like pixels are then discarded in post-processing and orientation, length, width and crack count are calculated. Percolation processing to determine the cracks on concrete surfaces was studied in [155]. In this study, the number of starting points for percolation is determined using edge detection. Most work in the literature therefore uses fixed scale or discrete number of orientations which is less than ideal.

A segment extending technique was proposed in [98] to effectively eliminate noise and establish connectivity. However, it does not classify the detected crack into any type/class and results are shown on two example images. Building a crack probability map using tensor voting, which enhances the connection of the crack fragments with good proximity and curve continuity was proposed in [165]. In this study, a set of crack seeds are sampled from the crack probability map. They are then represented by a graph model from which the minimum spanning trees are derived. Recursive tree-edge pruning is then conducted to identify desirable cracks. The results of this algorithm look promising, even though the types of cracks are not identified.

Use of Markov random fields, in which a lattice composed of straight line segments is projected
on the image was suggested in [40] and the results were reported on one image. 2D matched filter to adapt the mother wavelet (that results in a multi-scale detection) followed by Markov random fields to segment the structures in the image was used in [25]. Results were reported on multiple images and appear noisy. Crack detection using wavelet decomposition and Markov random fields was proposed in [24] which provided a ground truth data of 32 images (including synthetic and real) for reporting quantified results. Evaluation of different techniques for crack detection is challenging due to the absence of a standard dataset with ground-truth annotations. The annotations should also contain details of types of cracks (longitudinal, transversal and alligator) as these features are a key component in the evaluation of pavement condition and maintenance strategies by civil engineers.

Several contributions are made here. First, two techniques are proposed for detection of cracks along with their types. First method is supervised while the second is unsupervised and requires no training, has very intuitive and limited number of model parameters. Second, a dataset of 150 images with ground-truth annotations containing details of the types of cracks is provided. This dataset is made publicly available at (http://www1.coe.neu.edu/~sghanta/dataset/). Third, a quantitative comparison of the results obtained by the proposed methods is presented.

10.3 Algorithms

The goal is to develop an automated algorithm that detects cracks along with their types and can take into account the connectivity in cracks irrespective of the direction. Two strategies to do this are presented in the following sections. The first strategy uses a multi-class logistic classifier to identify the cracks in four directions at a constant scale. The second strategy utilized a multi-scale Hessian based classifier that can identify cracks in any direction at different scales. The results of
the algorithm are compared with ground-truth annotations to quantify the accuracy of the proposed algorithm. A public dataset containing the raw images along with their ground-truth annotations is provided. Making available such a dataset will facilitate quantitative comparison with other algorithms in the future.

All the images are pre-processed to ensure consistency with respect to area covered by each pixel and reduce the number of false detections. A series of pre-processing steps are performed on the acquired images before detecting cracks. The images are acquired by a camera projected from the rear of the vehicle which results in angular distortion. Processing is required to eliminate camera angle distortion and get a birds eye view of the pavement. This ensures consistency across the image in terms of pixel/m² making placement on geo-referenced maps more accurate. Therefore, homographic translation has been introduced as a pre-processing step. Parts of the image that contain lane and pedestrian markings on the road are detected and removed so that they don’t interfere with the crack detection process. This section describes the details of all pre-processing steps implemented.

**Calibration** Due to the angle at which the camera is projected on the rear end of the vehicle, there will be angular distortion in the image. This has the effect of showing lines that are originally parallel as intersecting lines. This distortion can be easily corrected as the transformation required is linear. The parameters of the homography matrix required for correction are calculated by taking four points on the calibration image as reference. The calibration and correction process described in [59] is used here. An example images along with the corrected image is shown in Figure 10.1. Note that all the images and ground-truth annotations that will be made publicly available have already been corrected for this distortion. Therefore, all images have a birds eye view and are
consistent in terms of area covered by each pixel in the image.

![Calibration image](image1.png) ![Image after angular distortion correction](image2.png)

Figure 10.1: Distortion correction from calibration image

**Lane Markings**  Lane markings interfere with the crack processing algorithm as they exhibit a sharp gradient at the edges. Masking out the parts of the image containing the lane marking does not solve the problem as the neighboring pixels will still be detected as cracks due to the gradient they exhibit with pixels that have been masked. To solve this problem, pixels in the image that have a high intensity value (> 200, which means the pixels are almost white in color) with a minimum of 1000 connected components and a 8 neighborhood rule are detected as lane markings (the intensity range of an image is [0, 255]). Then, intensity value of these pixels is modified to the average value of a mask of $11 \times 11$ around it excluding the pixels belonging to the lane markings within the mask.

### 10.3.1 Multi-Class Logistic Classifier

Using a probabilistic model for identification and classification of cracks is intuitive and effective.

Five classes are used in the algorithm, longitudinal crack, transverse crack, diagonal crack type 1, diagonal crack type 2, and not a crack. The algorithm itself learns the templates to be used for each class as opposed to the user giving a pre-defined template to the classifier. The proposed method
proceeds in steps and gives a probability value of each pixel to belong to a class rather than having a hard constraint that a pixel can only belong or not belong to a particular class. For example, the probability of a pixel can be 0.8, 0.1, 0.04, 0.05 and 0.01 to belong to a longitudinal, diagonal 1, diagonal 2, transverse and no-crack class respectively. This gives us freedom to work on the initial belief and adjust and improve it in further processing. The only constraint imposed is that the sum of the probability of each pixel belonging to different classes should be 1.

A Markov random field defined over a set of variables is an undirected graphical model that follows Markov property [57]. In case of images, mathematically it can be represented as

$$P(X(s) = v^{(s)}|X(S_{-s})) = P(X(s) = v^{(s)}|X(N(s)) = V^{N(s)})$$  \hspace{1cm} (10.1)$$

where, $s$ refers to a site in the image. $S$ refers to all the sites in the image. $v(s)$ is the intensity value at the site $s$. $S_{-s}$ refers to all the sites in the image except the site $s$. $N(s)$ refers to the Markov blanket which is also called the neighborhood of the site $s$ [86]. This implies that the probability of a particular intensity value at a given site depends only on its neighbors $N(s)$. Additional information about intensity values at other sites in the image does not alter its probability value. To define a distribution over the variables in a Markov random field, the following definitions are essential. A clique $c$ in a graphical model is defined as a set of sites such that if $s_i, s_j \in c$, then they both belong to a neighborhood $N(i)$ and $N(j)$. If $C$ is a set of cliques, a Gibbs distribution is defined as

$$P(x) = \frac{1}{Z} exp^{U(x)/T}$$  \hspace{1cm} (10.2)$$

where, $Z$ is the partition function, $T$ is called temperature, $U(x) = \sum_{c \in C} V_c(x)$, where $V_c$ are
sum of clique potentials over all $c \in C$. This can also be written as

$$P(x) = \frac{1}{z} exp^{\sum_k \lambda_k f_k(x)}$$  \hspace{1cm} (10.3)$$

with $\lambda_k$ being a model parameter and $f_k(x)$ the feature extraction function. The partition function is given by

$$Z = \sum_{x'} exp^{\sum_k \lambda_k f_k(x)}$$  \hspace{1cm} (10.4)$$

A partition function ensures that the distribution sums to 1. Calculation of the normalization constant can become intractable especially in case of images where the dimensionality of the data is high and $x$ can take values from 1 to 255. Let there be an additional variable $y$ that is of dimension $k$ and takes only binary values. $y = [y_1, y_2, \ldots, y_k]$. Here $k$ is the number of classes. The goal is to classify each pixel in the image into one of $k$ classes. For every pixel, $y_j = 1$ only in case that pixel belongs to the class $j$. The Markov random field equation in this case is given by

$$P(x, y) = \frac{1}{Z} exp^{\sum_k \lambda_k f_k(x, y)}$$  \hspace{1cm} (10.5)$$

and $Z$ is given by

$$Z = \sum_{x'} \sum_{y'} exp^{\sum_k \lambda_k f_k(x, y)}$$  \hspace{1cm} (10.6)$$

Clearly, the complexity of calculating the partition function has increased. A conditional random field is a nice trick applied to this generative model, which eliminates the calculation of the otherwise intractable partition function. A conditional random field is a Markov random eld [118], globally conditioned on $X$, the random variable representing observations which corresponds to
CHAPTER 10. VOTERS PROJECT

intensity values in the case of an image. Mathematically, a conditional random field is derived as

\[
P(y|x) = \frac{P(x, y)}{P(x)} = \frac{P(x, y)}{\sum_{y'} P(x, y)} = \frac{1}{Z} \exp \sum_k f_k(x, y) = \frac{\exp \sum_k f_k(x, y)}{\sum_{y'} \exp \sum_k f_k(x, y)}
\] (10.7)

Clearly, the partition function has cancelled out and one can represent \(\sum_{y'} \exp \sum_k f_k(x, y)\) by \(Z(x)\). \(Z(x)\) is not a constant and has to be calculated for every set of \(x\) values, it is called the normalization factor.

**Multiclass Logistic Classifier** The simplest model of conditional probability is given by multiclass logistic regression 11. The mathematical formulation of the model is given by

\[
P(y_k = 1|x, \theta) = \frac{\exp(\theta_k^T x)}{\sum_k \exp(\theta_k^T x)}
\] (10.8)

with \(\theta = [\theta_1, \ldots, \theta_k]^T, \theta_k = [\theta_{k1}, \ldots, \theta_{kd}]^T, x = [x_1, \ldots, x_d]^T\). The goal is to estimate the model parameters from the training data given by \(D = [(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)]\). This comes down to finding the maximum of the likelihood function given by

\[
l(\theta|D) = -\sum_i \left[ \sum_k y_{ik} \theta_k^T x_i - \ln \sum_k \exp(\theta_k^T x_i) \right]
\] (10.9)

The parameter values obtained using the ML estimate might cause over-fitting. To avoid over-fitting the data, we can have a Gaussian prior on the values of \(\theta\), with a zero mean and a variance \(\sigma^2\), i.e. the distribution of the parameters is given by

\[
P(\theta_{kj}|\sigma_{kj}) = N(0, \sigma_{kj})
\] (10.10)
The hyper-parameter in this case would be $\sigma_{kj}$. This takes care of most values being close to zero and avoids over-fitting, but does not make them exactly equal zero. This is called the MAP formulation as it incorporates the prior distribution of parameters into the likelihood function. The modified model is given by

$$l_{MAP}(\theta|D) = l(\theta|D) + \ln(P(\theta_{kj}))$$  \hspace{1cm} (10.11)

To induce sparsity in the parameter values, a double exponential (Laplace) prior distribution on the is used, given by

$$P(\theta_{kj}|\lambda_{kj}) = \frac{\lambda_{kj}}{2} \exp\left(-\lambda_{kj}|\theta_{kj}|\right)$$  \hspace{1cm} (10.12)

This becomes an optimization problem that can be solved using the coordinate decent algorithm [66]. The mathematical steps involved are clearly described in the paper [100]. This has the effect of automatically determining the templates to be used for classification. The sparsity constraint takes care of over-fitting and most of the less significant pixels around the central pixel are given zero weight.

The calibration image along with the images from several field tests are collected. All images are corrected for angular and radial distortion. The corrected images are divided into three disjoint subsets. Tuning of the hyper-parameters is necessary for good results. This is achieved by measuring the performance of the algorithm on the validation dataset and tuning the parameters to have the best results. Once the hyper-parameters have been calculated, they are stored in memory and the training step calculates the parameters of the model for classification. The output of the training step is a set of templates with different weights to the pixels around the central pixel. The test
images are processed using these parameters. The resulting images are post-processed to display the final result of the algorithm.

Goal of the crack detection algorithm is to detect all the cracks in a given image and calculate the percentage of cracked pixels in it. To achieve this, a multiclass logistic regression is used with Laplace priors on the parameters to make them sparse as described previously. The observations in this case are the pixel intensity values and the unknown variables $y$ represent the class each pixel belongs to. Each pixel label is calculated independent of the other pixels. To label each pixel in the image, a patch of $[nxn]$ pixels around it is chosen and given as input to make a decision on the probability of it belonging to a particular class. This is done for each and every pixel in the image.

**Training:** The goal of training algorithm is to estimate the parameters of Equation 10.8. Laplace priors on parameters guarantees that there is no over-fitting and the values are sparse. The training observations consist of image patches of size $n \times n$ whose label $y$ is already known. These are obtained by manual selection by a user from the images in the training dataset. Each patch is made into a $[n^2 \times 1]$ size vector and used as input directly. $y$ is formulated using a 1 of $K$ coding scheme. The four classes that the algorithm aims to classify all the pixels into are, longitudinal, transverse, diagonal and no-crack. Once the training data from all the classes is available, model parameters are determined from the optimization of Equation 10.12.

**Testing:** Goal of this stage of the algorithm is to detect all the potential crack pixels along with their class in the image. This algorithm is given the values of parameters determined from training step $\theta$ and images which form the observation $x$ as input. When an image is given to the algorithm to determine the class of all the pixels, it is done independently for each pixel by considering
Figure 10.2: Figure 4: (a) Longitudinal crack template (b) Diagonal crack template of type 1 (c) Transverse crack template (d) Diagonal crack template of type 2
a patch of size \([nxn]\) around it. The probability of each pixel belonging to a class is calculated using Equation 10.8. Therefore, for any given pixel, the output variable \(y\) is vector that consists of probabilities of that pixel belonging to each class. For example, the output variable \(y\) could be \([0.01, 0.8, 0.1, 0.04, 0.05]\) which means that the pixel belongs to transverse crack with a probability of 0.01, a longitudinal crack with a probability 0.8, a diagonal crack of different orientations (type 1 or 2) with a probability 0.1 and 0.05 respectively and not a crack with probability 0.05. Since the probability of the pixel belonging to longitudinal class is more than 0.5, it can be classified as longitudinal. Note that two, instead of one class is assigned to diagonal cracks. This modification is required to take into account the different orientations that belong to a diagonal crack.

### 10.3.2 Multi-scale Hessian Based Detection

Pixels belonging to cracks in the image are almost always darker in intensity value and form a ridge pattern. Frangi [53] proposed use of multi-scale second order local structure (Hessian) for detecting and enhancing vessel structures in medical images. This method detects tubular/ridge structures in the image in all the directions. To make the approach multi-scale, a measurement scale parameter is introduced that can be varied. This parameter can be determined by the user based on area covered by each pixel in the image and the maximum/minimum width of the ridges to be detected.

Let the image be denoted by \(I\) which contains \(N\) pixels and is given by \(I = [a_1, \ldots, a_n, \ldots, a_N]\). Hessian matrix of the image is computed at a location \(n\) and scale \(s\) given by Equation 10.13.

\[
H_{a_n, s} = \begin{bmatrix} I_{xx}(a_n; s) & I_{xy}(a_n; s) \\ I_{yx}(a_n; s) & I_{yy}(a_n; s) \end{bmatrix} \tag{10.13}
\]
where, $I_{xx}(n, s)$, $I_{yy}(n, s)$, $I_{xy}(n, s)$ are the second order derivatives, where $x$ and $y$ represent the direction of the gradient along the $x$ axis and $y$ axis respectively. The equation for differentiation is defined in Equation 10.14.

$$I_x(a_n; s) = \frac{\partial I(a_n, s)}{\partial x} = s^y I(a_n) * \frac{\partial G(a_n, s)}{\partial x}$$  \hspace{1cm} (10.14)

where, $\gamma$ is introduced to normalize the response of differential operators at different scales and is set to unity when no scale is preferred (which is the case here). The Gaussian kernel of standard deviation $s$ is given by Equation 10.15.

$$G(a_n, s) = \frac{1}{2\pi s^2} e^{-\frac{|a_n|^2}{2s^2}}$$  \hspace{1cm} (10.15)

The eigenvalues and eigenvectors of the Hessian calculated using Equation 10.13 can be analyzed to detect the geometric structures in the image. The eigenvector corresponding to the largest absolute eigenvalue ($\lambda_1$) corresponds to the direction of greatest curvature at the given pixel for a scale $s$. The other eigenvector (with eigenvalue $\lambda_2$) represents the direction of least curvature. The absolute eigenvalues represents the sharpness of the curvatures in the respective directions. Cracks in the image form ridge like structures where pixels belonging to crack show lower intensity and high curvature.

The eigenvalue and eigenvector decomposition of the Hessian matrix for a image hence consists of three output matrices: (i) The highest eigenvalue for each pixel across scales, (ii) The direction of curvature corresponding to highest eigen value at each pixel and (iii) Matrix containing the scale $s$ at which highest eigenvalue was detected. An example is shown in Figure 10.3. Detecting cracks
CHAPTER 10. VOTERS PROJECT

(a) Original image.  
(b) Eigen values corresponding to scale with highest curvature.

(c) Scale at which highest curvature was detected.  
(d) Direction of highest curvature.

Figure 10.3: Results from Hessian decomposition per pixel on example image.

(a) Threshold = 0.4  
(b) Threshold = 0.2

Figure 10.4: Results from different thresholds on Eigenvalues at different scales.
from these outputs is achieved by imposing constraints on the matrices.

To detect cracks in any direction in the image, the eigenvalue matrix is thresholded. The eigenvalue matrix at each scale is in the range \([0, 1]\). 5 different scales are used for detection of cracks where each scale corresponds to a different thickness. The number of scales to be used can be chosen from domain knowledge of image resolution and the thickness of cracks that need to be detected. In a case where the image is very noisy, a constraint on minimum number of connected pixels required for it to qualify as a crack can be used as a post-processing step. Domain knowledge about the contrast exhibited by cracks at different thickness can be used to set the threshold value.

The algorithm does not require training. Results with different parameter values on a validation set are presented. This is useful to get a quantitative understanding of how results vary with increase and decrease in the threshold value and will help future researchers to chose an appropriate value. Details of this analysis is presented in the experiments section.

### 10.3.3 Alligator Area Detection

Alligator cracking, also known as fatigue/crocodile cracking is an asphalt pavement distress characterized by huge number of interconnected cracks. An example image is shown in Figure 10.7(e). It is important to detect and correct alligator cracking because advanced alligator cracking can be very expensive to fix and often leads to formation of potholes. Measuring the area of the pavement covered by alligator cracking is critical for determining the overall condition of the road.

The following steps are implemented to detect the area covered by alligator cracks. Each image is divided into sub-parts and the number of pixels that belong to crack are calculated. One strategy would be to classify the sub-parts containing more crack pixels than a threshold as areas
containing alligator cracks. However, this feature alone is not discriminative enough as there can be subparts with thick longitudinal or transverse cracks (cracks that are parallel and perpendicular to the pavement’s centerline/laydown respectively) alone that confuse the classifier. To overcome this, the number of pixels that are in longitudinal and transverse direction are calculated and subtracted from the total number of cracks in the subpart. This results in two features from each subpart: the total number of pixels belonging to cracks and the pixels belonging to cracks that are neither transverse or longitudinal in direction. Note that there are a total of three parameters that can vary in this case: the number of subparts that the image should be divided into and the threshold on the two features extracted. Based on how fine grained the alligator crack detection should be, one can decide the number of sub-parts the image should be divided into. The results of the algorithm on a validation set are presented. This helps understand the response/results of algorithm for increase and decrease in each parameter. Details of this analysis are given in the experiments section.

10.3.4 Longitudinal and Transverse Crack Detection

As the name suggests, longitudinal and transverse cracks are parallel and perpendicular to the pavement’s centerline/laydown direction respectively. Severity of cracking is determined from the length and width of the cracks. This value contributes in the calculation of the overall condition of pavement.

Longitudinal cracks in the image are determined by retaining pixels from the cracks that have a direction within a tolerance $\epsilon = 30^\circ$ in either directions from $90^\circ$ angle. However, the output from this is very noisy as sub-parts of non-longitudinal cracks are also be detected. One strategy would be to remove the cracks with a constraint on minimum number of pixels required to qualify as a
longitudinal crack. However, very often, a continuous crack appears with very small discontinuities and this information is lost if parts of the crack are deleted using the above constraint. An example is shown in Figure 10.7(a). A strategy to get rid of noise while retaining the true longitudinal cracks even in the presence of discontinuities is presented in this section. Let there be a total of $M$ independent longitudinal cracks detected in the image. Let their locations be saved in the variable $L = [l_1, \ldots, l_M]$, where $l_m = [l_{s,m}, l_{e,m}]$ corresponding to the start and end point of the $m^{th}$ longitudinal crack. Let $p = [p_1, \ldots, p_M]$ be the length of each independent crack. The procedure to merge cracks is shown in Algorithm 3.

**Algorithm 3** Merge discontinuous longitudinal cracks

**Initial:** All longitudinal cracks in the image are detected and we have $M$ independent cracks; $L = [l_1, \ldots, l_M]$, $p = [p_1, \ldots, p_M]$.

**Step 1:** Update $\text{merge} = 0$.

For each possible pair $(i, j)$, where $i \neq j$,

if (dist($l_{e,i}, l_{s,j}$) < min($p_i, p_j$))

{ 
merge =1
M=M-1
update $L$ and $p$
}

**Step 2:** if $\text{merge} =1$, goto Step 1, else quit.

The above algorithm ensures that continuous longitudinal cracks with small discontinuities are detected as a single crack. After this, independent cracks with less than a minimum number of connected pixels are discarded as noise. The same process is followed for detecting the transverse cracks in its respective direction. Width and length of the cracks are determined once they have been identified.
10.4 Experiments

In this section, analysis and performance of the described algorithm is presented. Segmentation accuracy along with average segmentation accuracy is used as the measure of performance. Segmentation accuracy is defined as the ratio of sum of true positives (TP) and true negatives (TN) with the sum of true positives, false positives (FP), false negatives (FN) and true negatives (TN) \( \frac{TP+TN}{TP+TN+FP+FN} \). Average segmentation accuracy is defined as the ratio of true positives with the sum of true positives, false positives and false negatives \( \frac{TP}{TP+FP+FN} \) [49]. Analysis of the results on a small validation set of 50 images for different parameter values is presented. Then, results of the algorithm on another set of 150 images is presented. Ground truth data is available for all the images on which results are reported.

10.4.1 Determination of Optimal Parameters

**Crack detection threshold:** The output of the algorithm is a binary image with information about the direction of crack at each location. This binary image is generated by thresholding the eigenvalue matrix at different scales. In this case, 5 scales are used. A higher scale corresponds to a larger thickness of the crack. One can choose different number of scales based on the application and image resolution. Based on this dataset, 5 scales could capture all the cracks in the images. The threshold value range is [0, 1], where 0 indicates that there is no ridge structure at the given location and 1 indicates a very sharp ridge. If the threshold value is chosen to be too low (e.g. 0.2), a lot of noisy pixels are detected as cracks. On the other hand, a very high threshold value (e.g. 0.9) misses some of the crack pixels in the image. Using a value of 0.4 for all scales performs well if the goal is to capture all the cracks along with some associated noise which can be eliminated in
post-processing. If the goal is to minimize the number of false detections at the cost of leaving out a few true crack pixels, a higher threshold value can be used. One can also fine tune the parameters on a validation set to improve performance. In this section, a methodology to tune the parameters is presented.

The following steps are used to generate results for different combinations of parameters. The value of scale-1 is varied from 0.1 to 1, while keeping all the other scales at a constant value of 0.4. The value 0.4 is chosen for all other scales here as a baseline as it typically captures most of the true cracks in the image. Same procedure is followed for all the scales. The results of the algorithm for each scale are compared with the groundtruth segmentations and performance is evaluated using the overall segmentation accuracy and average segmentation accuracy metric. The results are reported in Figure 10.5. From the results, one can see that the optimal parameters for this dataset would be scale-1=0.6, scale-2=0.5, scale-3=0.5, scale-4=0.5 and scale-5=0.6. As can be seen from Figure 10.5, a small change in parameter value does not significantly effect the performance of the algorithm. In the experiments, a constant value of 0.4 is used for threshold on all scales as the goal is to capture all the cracks in the image at the cost of some additional noisy detections, which can be eliminated in post-processing.

**Alligator crack detection threshold:** After fixing the values for detection of cracks, results on the validation set for alligator area detection are reported. As described in Section 10.3.3, parameter-1 corresponds to the number of parts the image should be divided into. This value depends on the desired approximation for a given application. Typically, for pavement condition assessment [3], an approximate area covered by alligator cracking is acceptable. Parameter-1 is chosen to be 16 in this case as each image covers an area of $2.3348 m^2$ and one does not need a more
(a) scale-1 is varied.  
(b) scale-2 is varied.  
(c) scale-3 is varied.  
(d) scale-4 is varied.  
(e) scale-5 is varied.

Figure 10.5: Analysis of scale parameter variation for crack detection.
fine-grained result for pavement condition assessment. The results of the algorithm for different combinations of parameter-2 and parameter-3 is presented in Figure 10.6. The optimal parameters for this dataset are parameter-2=0.08 and parameter-3=0.048, which is a trade-off between the two measures of accuracy. Using these optimal parameters, following experiments are performed on a test dataset.

![Graph showing segmentation accuracy](image)

(a) Segmentation accuracy

![Graph showing average segmentation accuracy](image)

(b) Average segmentation accuracy

Figure 10.6: Analysis of parameter variation for alligator area detection.

### 10.4.2 Results

The dataset used here includes three sets, each containing 50 images, from three road condition categories of ‘Good’, ‘Fair’ and ‘Poor’ in the City of Brockton, MA. This classification has been made from an ASTM manual inspection [4] performed to assess the pavement condition of Brockton roadways [130]. Using a dataset that contains all types of cracks at varying severity levels ensures a fair comparison of the algorithm. Additionally, all the images are processed using another crack detection algorithm proposed in [59] as well. This algorithm is referred to as ‘MLR’
Figure 10.7: Example results of the algorithm. Green indicates alligator cracking, blue indicates longitudinal cracking, violet indicates transversal cracking and red indicates other random cracking. (a) and (e) are the original images, (b) and (f) are groundtruth labels, (c) and (g) are results of the ‘HMF’ algorithm, (d) and (h) show results of ‘MLR’.
(Multi-class Logistic Regression). Threshold parameter on the probability value is set to 0.6 as it is the optimal value for this case. The same pre-processing steps are used to make a fair comparison between the two algorithms. ‘HMF’ (Hessian-based Multi-scale Filtering) is used in all the tables to describe the algorithm. Groundtruth is referred to as ‘GT’.

The ‘HMF’ reports the following values after processing: The number of crack pixels in the image, the area covered by alligator cracks, length and width of transversal and longitudinal cracks. As discussed before, segmentation accuracy along with average segmentation accuracy is used as the measure of performance and reported in 10.2 and 10.3. Note that the ground truth labels for cracks do not provide any information about the width of the crack as they have been labeled with a constant brush size. This makes pixel-wise comparison challenging and the following approximation is made. For every independent crack, if there is at-least 25% overlap in crack pixels, it is considered to be fully detected. This set consists of 50 images from road with a good condition (Good), average condition (Fair) and a poor condition (Poor). The three sets are chosen based on the Pavement Condition Index (PCI) value reported using the standard manual surveillance. A good correlation of the condition of roads based on PCI value with the cracks verifies that the features extracted using crack detection are of high value for pavement condition evaluation. The crack statistics for poor condition roads are worse than those for a good condition road. This can be observed from the ground-truth and algorithm results provided in Table 10.4 and 10.6.

Segmentation accuracy and average segmentation accuracy values are reported for pixel-wise segmentation and alligator area. Both mean and standard deviation values are reported. Table 10.2 and 10.3 show that our algorithm has a much better performance than [59]. Figure 10.7 shows example results.

Each image is $2.3m^2$ in size. 50 images correspond to $116.74m^2$ area in total. Table 10.4 reports
Table 10.2: Pixel-wise segmentation accuracy and average segmentation accuracy values reported by algorithm with respect to ground-truth.

<table>
<thead>
<tr>
<th>Road type</th>
<th>Segmentation accuracy</th>
<th>Average segmentation accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
<td>‘MLR’</td>
</tr>
<tr>
<td>Good</td>
<td>0.98 ± 0.015</td>
<td>0.09 ± 0.18</td>
</tr>
<tr>
<td>Fair</td>
<td>0.98 ± 0.02</td>
<td>0.58 ± 0.27</td>
</tr>
<tr>
<td>Poor</td>
<td>0.93 ± 0.02</td>
<td>0.62 ± 0.1</td>
</tr>
</tbody>
</table>

the area in $m^2$ covered in alligator cracking for the total 50 images of each type. There is a strong correlation between the true condition of the road and alligator cracking as the ‘HMF’ algorithm results are very close to groundtruth. The sum of the lengths of longitudinal and transversal cracks detected in the test set of images is reported in Table 10.6. As mentioned before, a constant brush size is used and therefore the width of the cracks cannot be calculated for ground-truth data. However, results from the ‘HMF’ algorithm and [59] can be used to calculate the average width of the cracks along with their standard deviation. The results of longitudinal and transversal lengths detected per every image in each category by groundtruth, ‘HMF’ algorithm and [59] are reported in Table 10.6.

The violin plots in Figure 10.8 show a symmetric histogram along the y-axis, whose thickness represents the number of images that reported the corresponding y-axis value. Each blue dot represents one image in the dataset. Green boxes on each plot represent the median value and red colored cross symbol represents the mean. Figure 10.8(a), 10.9(a) and 10.10(a) show the alligator area, longitudinal and transversal crack length detected per image in each category (‘Good, ‘Fair’ and ‘Poor’) by groundtruth, ‘HMF’ algorithm and [59]. Note that the ‘HMF’ algorithm detections are closer to the groundtruth detections. Table 10.5, 10.7 and 10.8 reports the mean square error values with respect to alligator area, longitudinal and transversal crack length.
compared to groundtruth annotations respectively for ‘HMF’ and ‘MLR’. The same is reported in Figure 10.8(b), 10.9(b) and 10.10(b) respectively.

Table 10.3: Alligator area segmentation accuracy and average segmentation accuracy values reported by algorithm with respect to ground-truth.

<table>
<thead>
<tr>
<th>Road type</th>
<th>Segmentation Accuracy</th>
<th>Average segmentation accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
<td>‘MLR’</td>
</tr>
<tr>
<td>Good</td>
<td>0.94 ± 0.01</td>
<td>0.02 ± 0.1</td>
</tr>
<tr>
<td>Fair</td>
<td>0.86 ± 0.09</td>
<td>0.40 ± 0.32</td>
</tr>
<tr>
<td>Poor</td>
<td>0.79 ± 0.1</td>
<td>0.68 ± 0.22</td>
</tr>
</tbody>
</table>

Table 10.4: Area of alligator cracking reported by algorithm and groundtruth data in $m^2$.

<table>
<thead>
<tr>
<th>Road type</th>
<th>Alligator Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GT</td>
</tr>
<tr>
<td>Good</td>
<td>5.18</td>
</tr>
<tr>
<td>Fair</td>
<td>38.43</td>
</tr>
<tr>
<td>Poor</td>
<td>78.07</td>
</tr>
</tbody>
</table>

Table 10.5: Mean Square error in area of alligator cracking reported by ‘HMF’ and ‘MLR’ ($m^2$).

<table>
<thead>
<tr>
<th>Road type</th>
<th>Mean Square Error (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
</tr>
<tr>
<td>Good</td>
<td>0.096 ± 0.183</td>
</tr>
<tr>
<td>Fair</td>
<td>0.167 ± 0.17</td>
</tr>
<tr>
<td>Poor</td>
<td>0.333 ± 0.291</td>
</tr>
</tbody>
</table>

10.5 Conclusion

An unsupervised algorithm that can automatically detect the types of cracks from pavement surface images is presented. The algorithm reports alligator area, transversal and longitudinal crack
Table 10.6: Length(m) and width(cm) of longitudinal cracks reported by algorithm and groundtruth data.

<table>
<thead>
<tr>
<th>Road type</th>
<th>Length GT</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
<td>‘MLR’</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>8.93</td>
<td>11.3</td>
<td>2.94</td>
</tr>
<tr>
<td></td>
<td>± 0.65</td>
<td>± 1.8</td>
<td></td>
</tr>
<tr>
<td>Fair</td>
<td>19.1271</td>
<td>21.15</td>
<td>37.23</td>
</tr>
<tr>
<td></td>
<td>± 1.5</td>
<td>± 2.3</td>
<td></td>
</tr>
<tr>
<td>Poor</td>
<td>25.41</td>
<td>11.93</td>
<td>34.95</td>
</tr>
<tr>
<td></td>
<td>± 0.42</td>
<td>± 2.13</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Road type</th>
<th>Length GT</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
<td>‘MLR’</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>2.81</td>
<td>7.45</td>
<td>11.2</td>
</tr>
<tr>
<td></td>
<td>± 0.44</td>
<td>± 2.46</td>
<td></td>
</tr>
<tr>
<td>Fair</td>
<td>10.97</td>
<td>18.33</td>
<td>29.4</td>
</tr>
<tr>
<td></td>
<td>± 0.54</td>
<td>± 2.27</td>
<td></td>
</tr>
<tr>
<td>Poor</td>
<td>3.31</td>
<td>4.91</td>
<td>35.35</td>
</tr>
<tr>
<td></td>
<td>± 1.1</td>
<td>± 2.13</td>
<td></td>
</tr>
</tbody>
</table>

Table 10.7: Mean Square error in length of longitudinal cracking reported by ‘HMF’ and ‘MLR’ (m).

<table>
<thead>
<tr>
<th>Road type</th>
<th>Mean Square Error (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
</tr>
<tr>
<td>Good</td>
<td>0.13 ± 0.2</td>
</tr>
<tr>
<td>Fair</td>
<td>0.134 ± 0.15</td>
</tr>
<tr>
<td>Poor</td>
<td>0.27 ± 0.4</td>
</tr>
</tbody>
</table>

length and width, and a segmentation of the crack pixels in the image. The parameters of the algorithm that have to be tuned are limited, intuitive and robust. An analysis of the performance of the algorithm is presented using groundtruth annotations. The proposed method outperforms the results from another algorithm in literature. In the future, a number of consecutive images can be combined taking into account the overlap between them followed by an algorithm that detects crack types and calculates the statistics on a larger view.
Table 10.8: Mean Square error in length of transversal cracking reported by ‘HMF’ and ‘MLR’ (m).

<table>
<thead>
<tr>
<th>Road type</th>
<th>Mean Square Error (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘HMF’</td>
</tr>
<tr>
<td>Good</td>
<td>0.09 ± 0.33</td>
</tr>
<tr>
<td>Fair</td>
<td>0.15 ± 0.34</td>
</tr>
<tr>
<td>Poor</td>
<td>0.025 ± 0.0744</td>
</tr>
</tbody>
</table>

(a) Alligator area reported by groundtruth, ‘HMF’ and ‘MLR’.

(b) Mean square error in alligator area detection by ‘HMF’ and ‘MLR’.

Figure 10.8: Results of alligator on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset.
Figure 10.9: Results of longitudinal cracking on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset.
Figure 10.10: Results of transversal cracking on ‘Good’, ‘Fair’ and ‘Poor’ road condition dataset.
Chapter 11

Conclusion

This dissertation presents an object detection/segmentation model that can incorporate spatial context information along with shape and appearance prior. The model deploys a non-homogeneous spatial Poisson process prior to integrate contextual information to object detection within a Bayesian probabilistic setting. A Poisson process is the basis for many stochastic processes popularly used as priors in a Bayesian non-parametric setting. This dissertation introduces a nonparametric model that employs Poisson process itself as the prior. The resulting model is a nonparametric latent marked Poisson process that provides a natural framework to incorporate uncertainty in number, location, shape, and appearance of patterns in 1D, 2D and 3D data.

The model utilizes a nonparametric log-Gaussian-Cox process to infer the intensity parameter of the non-homogeneous spatial Poisson prior. Utilizing a non-homogeneous Poisson process in both 2D and 3D spatial setting is demonstrated in the experiments for supervised and unsupervised modes.

Inference on the model is challenging due to the presence of non-conjugate priors and likelihood based on partitioning of the image. This dissertation presents inference steps for learning the
model parameters using RJMCMC and a modification of the model that makes Gibbs sampling possible.

Experimental results are presented for 2D data on synthetic and two publicly available data sets which show that the model performs better than the state of the art techniques. In addition, the model also shows contour plots of stress areas where object traffic occurs using the posterior distribution of the Poisson intensity. We extend the model to 3D dataset for segmentation of dermal-epidermal junction in reflectance confocal microscopy images for dark and fair skin types. Dermal-epidermal junction of skin a feature of interest to clinicians as cancer is known to originate in its vicinity. The results of the algorithm for dermal-epidermal junction are promising with an average error of $5.41\mu m$ and $6.55\mu m$ for dark and fair skin types respectively. We also extend the model to 1D and report results on subsequence matching on a time-series dataset where the algorithm performs better than the state of the art methods.

Finally, we present the results for crack detection in surface images developed as a part of the VOTERS project. Tow automated algorithms that can detect cracks along with their types from images acquired using a camera mounted on a moving vehicle are presented. The first algorithm is based on multi-class logistic regression and requires training. The second algorithm is based on Hessian-based multi-scale filtering in the unsupervised setting. Experimental results using the two algorithms are presented, and the data along with ground-truth annotations is made publicly available.
11.1 Future Work

The presented latent shape marked Poisson process algorithm is complex and utilizes Gibbs sampling. Using a sampling framework for inference can be slow depending on the application and dimension of data. However, this can be made faster by using parallel processing. Possible future improvement is to explore parallel processing techniques to speed up the algorithm and improve its computational performance.

The detection of dermal-epidermal junction in 3D RCM stacks is accomplished as a first step towards automated detection of cancer cells in the skin. In the future, adjacent RCM slices can be mosaicked and algorithms that detect cancer cells can be developed.

In time-series data, further flexibility to the shape model can be provided by allowing freedom to undergo uneven stretching within the sub-sequence. This added flexibility can accomplish higher classification accuracy.

For the VOTERS project, cracks along with their types have been detected. A potential future direction is to mosaic adjacent images to develop algorithms that can detect defects like ‘block cracking’, which require a larger field of view. The ultimate goal of the VOTERS project is to monitor pavement quality over a period of time which will aid in the life cycle analysis and design of better repair strategies. This can be achieved by surveying the same pavement over a period of time and analyzing the crack statistics over time.

This dissertation presents a natural model for multiple object detection in images and extends them to higher (3D) and lower (1D) dimensions. Results from experiments on several datasets show the performance of the algorithm to be better than the state of the art techniques. The presented model is general, probabilistic and has scope to solve a wide range of problems as is also
demonstrated in the experiments. This flexibility to adapt to different applications will be useful to future researchers.
Bibliography


[14] Adrian Baddeley and Rolf Turner. Practical maximum pseudolikelihood for spatial point


[16] Shantanu Banik, Rangaraj M. Rangayyan, and Graham S. Boag. Landmarking and seg-


[22] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hi-
erarchical topic models and the nested chinese restaurant process. In *Advances in Neural

[23] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation.

[24] S. Chambon, J-M. Moliard C. Gourraud, and P. Nicolle. Road crack extraction with adapted
filtering and markov model-based segmentation - introduction and validation. In *Proceed-
ings of the 5th International Conference on Computer Vision Theory and Applications*, vol-
ume 2, pages 81–90. 2010.

matched filter in a markov random field for fine structure extraction: Application on road

2005.

[27] S C Chen, D M Bravata, and I Olkin. A comparison of dermatologists’ and primary
care physicians’ accuracy in diagnosing melanoma: a systematic review. *Arch Dermatol*,


