INFORMATION TO USERS

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

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

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

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Detecting and Extracting Complex Patterns from Images and Realizations of Spatial Point Processes

Daniel Charles Islip Walsh

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

Doctor of Philosophy

University of Washington

2000

Program Authorized to Offer Degree: Statistics

In presenting this dissertation in partial fulfillment of the requirements for the Doctorial degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, or to the author.

Signature 

Date 8/30/00

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Daniel Charles Islip Walsh

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Adrian E. Raftery

Reading Committee:

Adrian E. Raftery

Tilmann Gneiting

Thomas S. Richardson

Date: 8/30/00

University of Washington

Abstract

Detecting and Extracting Complex Patterns from Images and Realizations of Spatial Point Processes

by Daniel Charles Islip Walsh

Chair of Supervisory Committee

Professor Adrian E. Raftery
Statistics

A common goal in the field of Computer Vision is the detection and extraction of patterns (e.g. lines, object boundaries) from *binary image data.* These images routinely occur as the product of edge detection algorithms, mathematical morphological operations, or simple thresholds, applied to a gray-scale or color images. In certain cases, binary image data can resemble *spatial point patterns.* In the statistics literature the usual goal of analyzing a point pattern is to make inference about the underlying process that generated the data. Both of these problems could be broadly classified as *pattern recognition* tasks.

In this dissertation, we investigate the following pattern recognition problems:

1. How to distinguish between mixtures of spatial point processes.

2. How to identify a subset of points in a spatial point pattern that displays a particular stochastic pattern.

3. How to efficiently identify parametric curves in binary edge images.

Problems 1 and 2 are motivated by the task of detecting minefields from aerial images. We develop partial Bayes factors to approach problem 1 and investigate their effectiveness via a simulation study. The second problem is tackled by developing a stochastic model to describe the particular stochastic pattern we are seeking (in this case the location of the mines in a minefield). A Markov chain Monte Carlo algorithm is outlined to fit the model. Results from applying the model to several datasets are presented.

In the third problem, we introduce the Hough transform as a method for detection of parametric curves in binary images. We re-examine this transform through the framework of importance sampling, which leads to a better understanding of its properties. An improved algorithm for parametric curve detection is given.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Adrian Raftery, for his advice and guidance, during the past three years. I would also like to thank the other members of my committee, Tilmann Gneiting and Thomas Richardson, particularly for their insightful comments and diligent reading of the draft of my dissertation. Special thanks goes to Marsha Landolt, the Dean of the Graduate School, who stood in at the last minute at my dissertation defense.

I would like to acknowledge, Christian Posse for helpful comments on Chapter 3, and Julian Besag, for suggesting that the Hough transform would be an interesting topic to investigate.

I am grateful to the Office of Naval Research, in particular to grants N00014-97-1-0736 and N00014-96-1-0192, for the majority of my financial support during my research.

I would like to thank my fellow graduate students, past and present, for their friendship and camaraderie over the last five years. In particular I would like to thank everyone who participated on the Department's softball team and made it such a source of enjoyment for me.

I would especially like to thank my parents, Dennis and Janet Walsh, for their unfailing support and love throughout my life, particularly for my decision to pursue my graduate studies overseas. Finally, I would like to thank my wife, and best friend, Beatrix Jones, whose enthusiasm, encouragement, and support was unwavering.

DEDICATION

This dissertation is dedicated to my parents, Dennis and Janet Walsh, and to my wife, Beatrix Jones.

Chapter 1

INTRODUCTION

1.1 Motivation

The extraction of complex patterns from an image, in the interest of obtaining a more complete understanding of the image's composition, is an issue of fundamental importance in the field of Computer Vision. Whether this pattern recognition task is as "simple" as identifying all lines in an image, or as complex as recognizing a human face, the gold standard that one usually attempts to achieve is the human eye (or more correctly the human mind). The human eye has the marvelous ability to be presented with a new image, and almost instantaneously be able to, with a high degree of accuracy, determine the content of the image: the composition of the background, the number of foreground components, differences in textures and brightness, etc. The goal for the computer vision engineer when attempting to mimic the human eye, is to produce reliable image processing algorithms that perform well in a wide variety of situations, and which are computationally efficient.

However there are occasions when the human eye is baffled, where an underlying complex pattern in the image goes unnoticed. It may be the case however, that a image processing algorithm, engineering transform, or statistical technique exists that may be able to tease out the latent signal.

Often it is beneficial to pre-process the original image data into a binary format, e.g. via an edge detector, before attempting to identify patterns. Each pixel in the image is assigned to be in the foreground (pixel = 1) or background (pixel = 0).

These binary data can be thought of as a realization of a spatial point process on a discrete grid. Typically in the point process literature, the goal is to compare, via hypothesis tests, various competing models for the data. This can be thought of as another form of pattern recognition. In this dissertation, we consider the following pattern recognition problems:

- How to distinguish between mixtures of spatial point processes.
- How to identify a subset of points in a spatial point pattern that displays a particular stochastic pattern.
- How to efficiently extract parametric curves from binary edge images.

The first two situations are difficult for the human eye. The main objective here is to successfully detect/extract a complex pattern from the data. Computational issues are not our foremost concern. The last situation is one in which the patterns are easily identifiable by eye. The goal is to develop a robust, automatic method that performs well but is also computationally efficient. This research evolved from a problem posed by the Office of Naval Research, namely the minefield problem (see Section 1.2.1).

1.2 Background

In this section we briefly describe the problem that motivated our research. We also review some of the literature underpinning it.

1.2.1 The Minefield Problem

In order to detect minefields in coastal areas before troop deployment, the U.S. Marine Corps developed the Coastal Battlefield Reconnaissance and Analysis (COBRA) program. The program consists of flying an unmanned aerial vehicle, fitted with

multi-spectral video cameras over the area of interest. The area is imaged and the images are either stored, or transmitted directly to a processing station.

The first stage of image processing identifies potential mines in the images. Unfortunately it also identifies many objects that are not mines. Thus we can consider the data to be a spatial point pattern of possible mine locations. The task at hand is therefore to decide whether or not a minefield is present in the data, and if so decide which objects are mines.

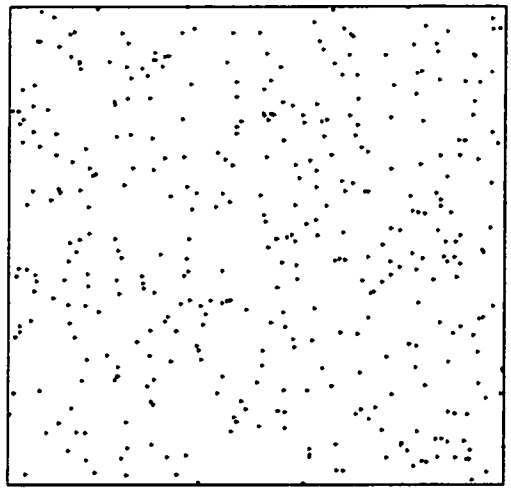
1.2.2 Spatial Point Processes

Consider a dataset that consists of the locations of a set of "small" objects in a bounded region of space. Such a dataset is known as a *spatial point pattern*. If the locations were all generated by a common stochastic process, then we could call these data a realization of a *spatial point process*. In the literature, an object in a spatial point pattern is referred to as an *event*. The term *point* refers to any arbitrary location in the sample space. The simplest spatial point process is the Poisson process, which has the following properties:

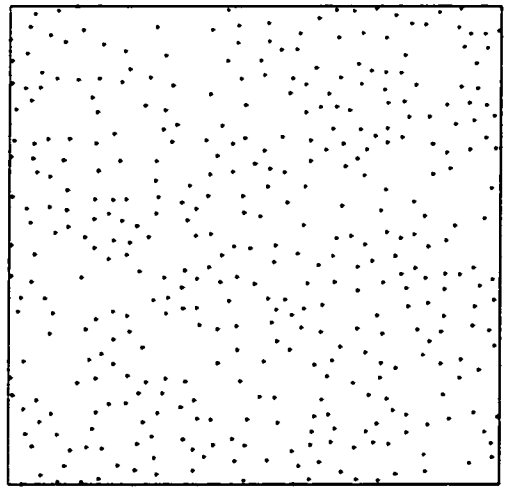
1. The number of events in any bounded region, A, is Poisson($\lambda|A|$), where λ is rate of the process and $|A|$ is the area of the region.

2. Given that there are N events in the region, A, the locations of all N events are independent, and uniformly distributed throughout A.

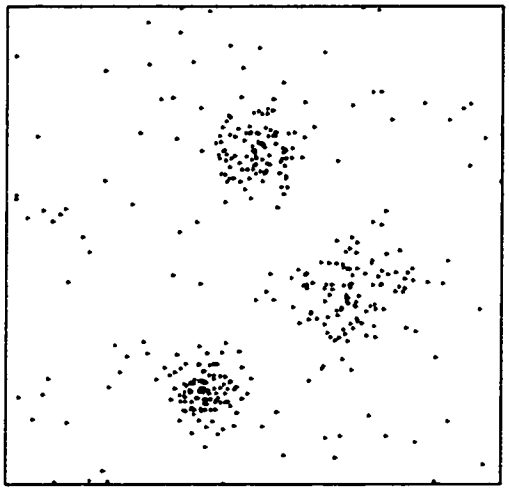
The Poisson process is sometimes referred to as complete spatial randomness (CSR). Point processes that depart from the Poisson distribution tend either to exhibit *inhibition* or *clustering* (see Figure 1.1). Until recently the point process literature confined itself to the problem of comparing simple models for inhibition or clustering versus CSR. Reviews of the spatial point process literature can be found in Diggle (1983), Cressie (1993), and Stoyan and Kendall (1995). The problem of classifying



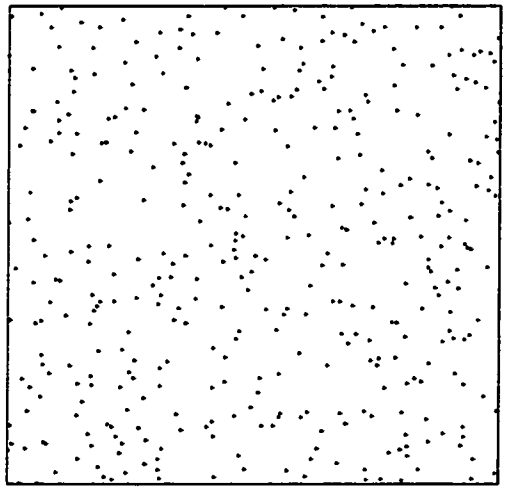
(a) Poisson spatial point pattern.



(b) Inhibited spatial point pattern.



(c) Clustered spatial point pattern.



(d) Mixture of inhibited and Poisson spatial point pattern.

Figure 1.1: Realizations of different classes of spatial point processes.

mixtures of spatial point processes has received far less attention. It was tackled by Raghavan, Goel, and Ghosh (1998). Their approach was to develop a supervised pattern recognition scheme using various functionals of the data.

1.2.3 Markov Chain Monte Carlo

Consider a random vector X with distribution $\pi(\cdot)$, from which one wishes to draw a sample, $x = \{x_1, \ldots, x_n\}$. Markov chain Monte Carlo (MCMC) is a technique that allows one to draw a *dependent* sample from $\pi(\cdot)$. It achieves this by constructing a Markov chain with transition kernel $P(x_{t+1}|x_t)$ whose limiting distribution is $\pi(\cdot)$. This is achieved in the following way:

1. Define a *proposal* distribution $q(\cdot|x_t)$.
2. Set X to some initial value x_0. Let $t = 0$.
3. Sample a value, x^*, from $q(\cdot|x_t)$.
4. Set $x_{t+1} = \begin{cases} x^*, & \text{with probability } \alpha(x^*|x_t) = \min\left(1, \frac{\pi(x^*)q(x_t|x^*)}{\pi(x_t)q(x^*|x_t)}\right) \\ x_t, & \text{else} \end{cases}$
5. Increment t, and return to step 3, until $t = n$.

The sample size, n, needs to be large enough to ensure the Markov chain has converged to its limiting distribution. However, knowing when the Markov chain has converged can be difficult. The above algorithm is known as the Hastings algorithm (Hastings 1970), and is a generalization of the Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953), where $q(x^*|x_t) = q(x_t|x^*)$. Although MCMC has been around in the field of statistical physics for over forty years, it was only in the last fifteen years or so that it has become widely used in statistics, particularly in the area of Bayesian inference. Reviews of MCMC methods can be found in Gilks, Richardson, and Spiegelhalter (1996), and Smith and Roberts (1993).

1.2.4 Hough transform

The Hough transform is a technique for detecting parametric curves (e.g. lines, circles, ellipses) in edge detected (binary) images. The Hough transform was originally proposed by Hough (1962) as a method for automatically tracking particle tracks in a bubble chamber. It can be thought of as a vote-counting procedure. Each point (foreground pixel) in the image votes for the parameters of any curve that passes through it. Thus, the difficult global optimization problem of detecting instances of a particular type of curve in an image space, is reduced to the problem of finding local peaks in the parameter space of the curve. Unfortunately for curves more complex than lines, the performance of the Hough transform suffers from large storage requirements and slow execution times.

A new class of Hough transforms, called probabilistic Hough transforms, attempted to address these limitations. These new Hough transforms greatly reduced the storage requirements of the standard Hough transform and decreased its execution time as well. Unfortunately their performance suffered when the image was complex in nature, i.e. noisy, containing multiple objects. A detailed introduction to the Hough transform can be found in Section 4.1.

1.3 Outline of Dissertation

The problem of classifying mixtures of spatial point processes is introduced in Chapter 2. A previous approach to this problem by Raghavan, Goel, and Ghosh (1998), is discussed, as is earlier literature concerned with the simpler problem of testing pure (i.e. non-mixture) point process models. Bayes factors are briefly reviewed, before partial Bayes factors are proposed as a tool for comparing evidence for competing point process models. Results of a simulation study are shown.

The COBRA problem, the problem of detecting individual mines from a processed aerial image, is discussed in Chapter 3. A stochastic model for the location of mines

in a minefield is presented and a Markov chain Monte Carlo algorithm to fit the model is outlined. Results from fitting this model to several datasets are shown.

In Chapter 4 the relationship between importance sampling and the Hough transform is explored. It is shown that the use of a target distribution to assess the quality of sampled curves can lead to improvements in the efficiency and accuracy of the Hough transform. Probabilistic stopping rules to determine the required number of curve parameters to sample are developed. A simple clustering technique is also given to allow the detection of multiple curves from a sample of curve parameters.

Chapter 5 summarizes the major results of this dissertation and outlines possible avenues for future research.

1.4 Contributions

The original contributions of this dissertation are:

- Investigation of the applicability of partial Bayes factors to classify spatial point processes, in particular, mixtures of spatial point processes.

- Development of a stochastic model for the placement of mines in a minefield. Implementation of a Markov chain Monte Carlo algorithm to fit this model.

- Investigation of the relationship between importance sampling and the Hough transform. Demonstration of the importance of defining a target distribution, to measure the quality of each sampled curve.

- Development of probabilistic stopping rules to determine if a sufficient number of curve parameters have been sampled when performing the Hough transform.

- Development of an efficient clustering algorithm that allows the Hough transform to detect multiple curves from one set of sampled parameters.

Chapter 2

CLASSIFICATION OF MIXTURES OF SPATIAL POINT PROCESSES VIA PARTIAL BAYES FACTORS

2.1 Introduction

We investigate the problem of comparing competing models for spatial point process data. In particular we are interested in testing the hypothesis that the data were generated by a Poisson process (i.e complete spatial randomness) versus a mixture of a Poisson process and an inhibited process. The motivation behind this methodology is the problem of minefield detection. An aerial view of a possible minefield has been imaged. This image is processed into a set of object locations. Each object is either a mine or can be considered to be clutter or noise. The mines are assumed to be laid out in such a way that two mines are unlikely to be close to together.A hard-core Strauss process is one way to model this inhibition. The noise points are assumed to be located randomly throughout the study region. The inherent difficulty of this problem can be seen in Figures 2.1(a), 2.1(b), and 2.1(c). This is a problem where the human eye offers few visual cues, yet statistical techniques can produce surprisingly good results.

Until recently the point process literature confined itself to the problem of comparing a simple model for inhibition or clustering versus complete spatial randomness (see Diggle 1983, and Cressie 1993). The problem of classifying mixtures of spatial point processes was tackled by Raghavan, Goel, and Ghosh (1998). Their approach was to develop a supervised pattern recognition scheme using functionals based on nearest neighbor distances, second order statistics and spatial tessellations. Our ap-

proach to this problem is similar in nature. We propose comparing the evidence for each model by using *partial Bayes factors*. We use the term partial Bayes factor to describe a Bayes factor, a ratio of integrated likelihoods, based on only part of the available information, namely that information contained in a small number of functionals of the data.

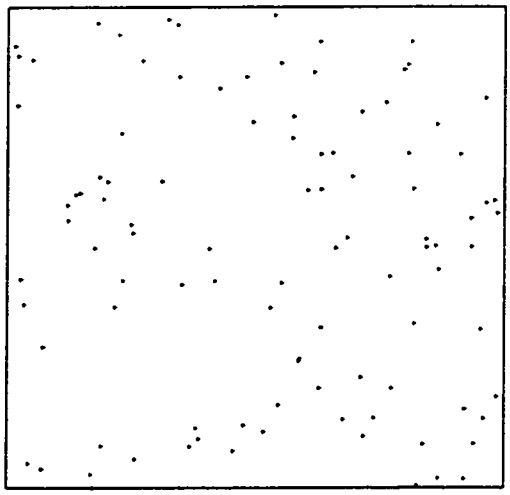
In the following sections we describe the spatial point process models we use and formulate the minefield problem as a hypothesis testing problem. We briefly review Bayes factors and define partial Bayes factors in Section 2.4. In Section 2.5 we discuss possible summary statistics one could use. Results of a simulation study are presented in Section 2.6 and are discussed in Section 2.8.

2.2 Point Process Models

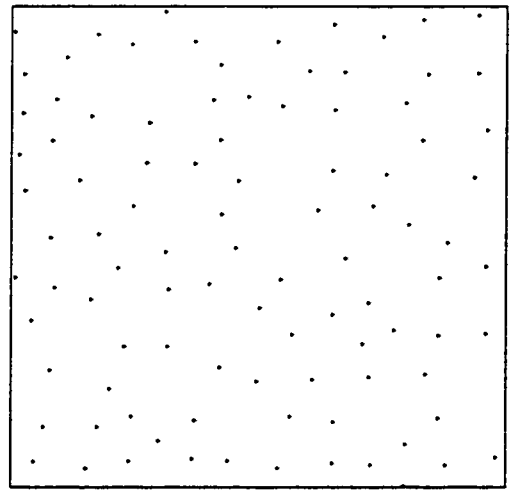
First we will describe some basic notation that we shall use throughout this chapter. To avoid possible ambiguities associated with the word "points", we shall refer to locations of objects as *events*, and let the word *point* refer to any point in the sample space. The sample space or study region will be denoted by A, and $|A|$ will denote the area of this region. Let N be the total number of events, n_0 be the number of noise events, and m be the number of mines. Let d_{ij} be the distance between the i^{th} and j^{th} events, and let $d_i = \min_j d_{ij}$. We shall condition on the number of events, N, and the study region, A, throughout. Let $Y = (y_1, \ldots, y_N)$ be a random vector taking values in A^N, that represents the locations of all events in A.

2.2.1 Noise process

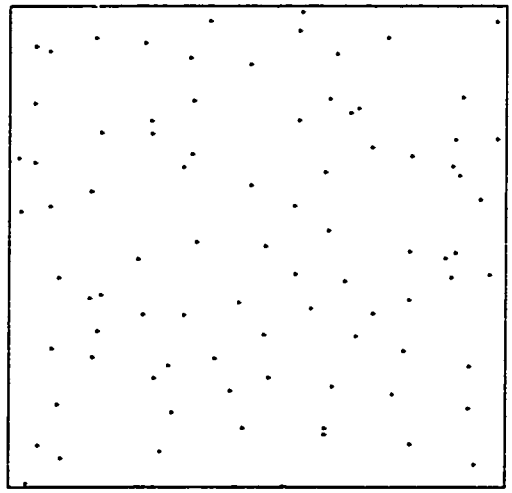
As was mentioned in the introduction, the noise events are considered to be scattered randomly throughout A. Under the hypothesis that no minefield is present, and given that we are conditioning on the number of events in A, the likelihood of Y is uniform over A^N, i.e.:



(a) Uniform process.



(b) Strauss process.



(c) Mixture process.

Figure 2.1: Examples of spatial point patterns.

$$P_u(Y) = \frac{1}{|A|^N}.$$

2.2.2 *Minefield process*

The mines are assumed to be spread evenly over A. This implies that the minefield process displays *inhibition*. A simple model for an inhibited process is the *Strauss* process (Strauss 1975; Kelly and Ripley 1976). The likelihood for the Strauss process is:

$$P_s(Y \mid \theta) = C \prod_{i<j} g_\theta(d_{ij}),$$

where $g(\cdot)$ is the *interaction* function, given by:

$$g_\theta(d) = \begin{cases} \gamma, & 0 \leq d < \rho, \text{ where } \gamma \in [0,1] \\ 1, & d \geq \rho, \end{cases}$$

and where the parameters of the Strauss process are denoted by $\theta = \{\rho, \gamma\}$. The extent of the interactions between two events is controlled by ρ, while the nature of these interactions is determined by γ. If $\gamma = 0$ the process is known as a *hard-core* process. In this process, two events are forbidden to be within distance ρ of each other. Alternatively, if $\gamma = 1$, the process is simply a uniform process on A. Values of γ between 0 and 1 discourage but do not forbid events to be within distance ρ of each other. Note that the normalizing constant, C, of the Strauss process can be difficult to calculate, especially for processes demonstrating strong inhibition (see Diggle, Fiksel, Grabarnik, Ogata, Stoyan, and Tanemura 1994).

2.2.3 *Mixture process*

Consider a superposition of a Strauss process with a uniform process. Let $Y = Y_u \cup Y_s$, where Y_u are the events generated by the uniform process and Y_s are the events

generated by the Strauss process. Let Z be a variable indicating to which group each observation belongs, i.e.

$$Z_i = \begin{cases} 0, & \text{if } y_i \in Y_u \\ 1, & \text{if } y_i \in Y_s, \end{cases}$$

for $i = 1, \ldots, N$. Note that $\sum_1^N Z_i = m$, the number of Strauss events (mines). If Z is known, then the likelihood for the mixture process can be written as:

$$P_m(Y \mid Z, \theta) = P_u(Y_u \mid Z, \theta) \times P_s(Y_s \mid Z, \theta).$$

If Z is unknown (as would be the case in practice) then we must sum over all the values of Z, multiplied by their respective probabilities, i.e.

$$\begin{aligned} P_m(Y \mid \theta) &= \sum_{z \in Z} P_m(Y \mid Z, \theta)\pi(Z \mid \theta) \\ &= \sum_{m=0}^{N} \sum_{z \in Z \mid \sum z = m} P_m(Y \mid Z, \theta)\pi(Z \mid m, \theta)\pi(m \mid \theta). \end{aligned}$$

Given the problem of obtaining the normalizing constant for a Strauss process, this sum is extremely difficult to compute.

2.3 *Formulation of the Minefield Problem as a Hypothesis Testing Problem*

We cast the minefield problem in terms of two competing hypotheses. Here we model the minefield process as a hard-core Strauss model. Thus, for a given point pattern Y, the competing hypotheses of interest are:

$$\begin{aligned} H_0 \ : \ & \text{No minefield present} \\ & Y \sim \text{Uniform}(N, A); \end{aligned}$$

H_1 : Minefield present

$Y = Y_u \cup Y_s$ where

$Y_u \sim \text{Uniform}(n_0, A)$,

$Y_s \sim \text{Strauss}(m, A, \rho, \gamma = 0)$,

and $m + n_0 = N$.

2.3.1 Prior decomposition

Under H_1, there are two unknown model parameters, ρ and m. In a Bayesian framework, one must specify a prior distribution $\pi(\rho, m)$ on ρ and m. The prior could be decomposed in the following ways:

1. $\pi(\rho, m) = \pi(\rho) \times \pi(m)$ (Assuming ρ and m are independent.)

2. $\pi(\rho, m) = \pi(\rho \mid m) \times \pi(m)$

3. $\pi(\rho, m) = \pi(m \mid \rho) \times \pi(\rho)$.

The choice as to which prior is more appropriate is debatable. If prior information about the number of mines and inhibition distance was good, then the independence assumption of the first prior may be reasonable. However, given that we are conditioning on the study region, A, and the total number of events, there exists a constraint on the maximum separation between two events and the total number of mines in A. Diggle (1983) noted that the maximum proportion of a finite region, A, that can be covered by non-overlapping discs, of radius ρ, is achieved when the discs are packed in a equilateral triangular lattice. This suggests the maximum value of ρ, given that there are m points in A, (ignoring edge effects) is:

$$\rho_{max} = \sqrt{\frac{2}{\sqrt{3}} \frac{|A|}{m}}$$

This bound will be useful in setting a prior for ρ conditional on m. For instance if one has only a vague idea on the number of mines, but knows that they are closely packed together, then one could use priors of the following form:

$$\begin{aligned} \pi(m) &= \text{Discrete Uniform}(m_1, m_2) \\ \pi(\rho \mid m) &= \text{Uniform}(\alpha_1 \rho_{max}, \alpha_2 \rho_{max}), \text{ where } 0 \leq \alpha_1 < \alpha_2 \leq 1 \end{aligned}$$

These are the prior distributions we used in our simulation study in Section 2.6.

2.4 Partial Bayes Factors

In this section we briefly introduce Bayes factors and define what we mean by partial Bayes factors. Consider data Y which is assumed to have arisen under one of the two competing hypotheses, H_0 or H_1. Let θ_i be a d_i-dimensional vector of parameters associated with hypothesis H_i $(i = 1, 2)$, and let $\pi_i(\theta_i \mid H_i)$ denote its prior distribution. Let the the probability density of Y given the value of θ_i, i.e. the likelihood function, be denoted by $P(Y \mid \theta_i, H_i)$. The Bayes factor for choosing H_1 over H_0, is given by:

$$\begin{aligned} BF_{10} &= \frac{P(H_1 \mid Y)}{P(H_0 \mid Y)} \Big/ \frac{P(H_1)}{P(H_0)} \\ &= \frac{P(Y \mid H_1)}{P(Y \mid H_0)} \\ &= \frac{\int P(Y \mid \theta_1, H_1) \pi_1(\theta_1 \mid H_1)\, d\theta_1}{\int P(Y \mid \theta_0, H_0) \pi_2(\theta_0 \mid H_0)\, d\theta_0} \end{aligned}$$

In other words the Bayes factor is the ratio of integrated likelihoods. The Bayes factor provides evidence for one hypothesis over another. Kass and Raftery (1995) review the history, development, and use of Bayes factors. A guide for interpreting Bayes factors, proposed by Kass and Raftery (based on Jeffreys 1961), is given in Table 2.1.

Table 2.1: Guide for interpreting Bayes factors.

$2\log_e(B_{10})$	B_{10}	Evidence for H_1
0 to 2	1 to 3	Weak
2 to 5	3 to 12	Positive
5 to 10	12 to 50	Strong
> 10	> 150	Decisive

In the mixture models we consider, while it is possible to simulate realizations from each model, it is impossible to write the likelihood down explicitly since the normalizing constant and the group memberships, Z, are unknown. Thus we resort to the following approximation based on a summary statistic, X (or a vector of several summary statistics, X):

$$\frac{P(Y \mid H_1)}{P(Y \mid H_0)} \approx \frac{P(X \mid H_1)}{P(X \mid H_0)}$$

This approximation will be good if X distinguishes well between the different point processes. Given this statistic X, the partial Bayes factor for H_1 versus H_0 and a given realization x of X, is:

$$\begin{aligned} PBF_{10} &= \frac{\int P(x \mid \theta_1, H_1)\pi_1(\theta_1 \mid H_1)\, d\theta_1}{\int P(x \mid \theta_0, H_0)\pi_2(\theta_0 \mid H_0)\, d\theta_0} \\ &= \frac{I_1(x)}{I_0(x)} \end{aligned}$$

We can calculate these integrated likelihoods by quadrature methods or by Monte Carlo integration. If $\theta_i = \{\theta_i^{(1)}, \ldots, \theta_i^{(K)}\}$ is a random sample of size K from the prior under hypothesis i, and $\widehat{P}(X \mid \theta_i^{(j)}, H_i)$ is an estimate of $P(X \mid \theta_i^{(j)}, H_i)$, then the Monte Carlo estimate of I_i is:

$$\widehat{I}_i(x) = \frac{1}{K}\sum_{j=1}^{K} \widehat{P}(x \mid \theta_i^{(j)}, H_i)$$

To obtain the estimated density function $\widehat{P}(X \mid \theta_i^{(j)}, H_i)$, we simulate 100 point patterns from H_i with parameters $\theta_i^{(j)}$, and calculate their summary statistics. Let these 100 summary statistics be denoted by $X_i^{(j)}$. A standard density estimation procedure, e.g. kernel density estimation (Silverman 1986), is then applied to X_i to obtain $\widehat{P}(X \mid \theta_i^{(j)}, H_i)$.

Obviously the selection of X is important. We discuss choices of X below. Note that nowhere do we assume that X is univariate. A bivariate or higher dimensional statistic may give better discrimination between the hypotheses. However, this may lead to excessive computation as well. Also, density estimation in more than one dimension can be difficult.

2.5 Summary Statistics

The types of summary statistics considered by Raghavan, Goel and Ghosh (hereafter RGG) for their supervised pattern recognition scheme fell into three main categories: nearest neighbor distances, second order statistics, and spatial tessellations. We describe each category below.

2.5.1 Nearest Neighbor Distances

The empirical cumulative distribution function (CDF) of the nearest neighbor distances between all events is given by:

$$\mathbb{G}_N(d) = \frac{1}{N}\sum_{i=1}^{N} 1_{\{d_i<d\}}, \quad d > 0.$$

This function can highlight differences in small-scale interactions between different point process models. A similar function to $G(d)$ is $F(d)$, the empty space function.

This is the CDF of the distance of an arbitrary fixed point in A to the nearest point of the spatial point pattern. RGG recorded the minimum, the mean, the coefficient of variation, skewness and kurtosis of $\mathbb{G}(\cdot)_N$, and $\mathbb{F}(\cdot)_N$, as well as the ratio $\mathbb{G}(\cdot)_N/\mathbb{F}(\cdot)_N$.

2.5.2 Second Order Statistics

The K-function, (Bartlett 1964; Ripley 1976, 1977; Cressie 1993, Ch. 8) has been used extensively as an exploratory tool for analysis of point patterns, in particular their second order statistics. For a spatial point process of intensity λ, it is defined as:

$$K(d) = \lambda^{-1} \text{ E (\# of events within distance } d \text{ of an arbitrary event)}$$

An estimator that corrects for edge effects was given by Ripley (1976):

$$\widehat{K}(d) = \frac{|A|}{N^2} \sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} w_{ij} 1_{\{d_{ij} < d\}}, \quad d > 0.$$

where w_{ij} is the proportion of the circumference of a circle centered at event i that passes through event j, that is inside the study region A. The intensity of the spatial point process, λ, is estimated by $\widehat{\lambda} = \frac{N}{|A|}$.

If the underlying process over a region with area $|A|$ is uniform, then the distribution of events within a ball of radius d around a given event, assuming the ball is contained in A, is binomial with mean $np = \frac{N}{|A|}\pi d^2$. Thus, $K(h) = \pi h^2$ and $\sqrt{K(d)/\pi}$ versus d is a line of slope one through the origin. RGG proposed the following two statistics based on the K-function:

1. The difference between the area under $\sqrt{\widehat{K}(h)/\pi}$ and the 45^0 line over the initial part (from $\min_i(d_i)$ to $\max_i(d_i)$) of the curve: i.e.

$$\int_{\min_i(d_i)}^{\max_i(d_i)} \left(\sqrt{\widehat{K}(u)/\pi} - u \right) du$$

2. The slope of $\sqrt{\widehat{K}(h)/\pi}$ from $\min_i(d_i)$ to $\max_i(d_i)$.

We propose a similar statistic based on the K-function. Under strict inhibition, Isham (1984) showed that in the plane the K-function for the Strauss process with $\gamma = 0$ is approximately:

$$K(d) = \begin{cases} 0, & 0 \leq d \leq \rho \\ \pi d^2 - \pi \rho^2, & d > \rho \end{cases}$$

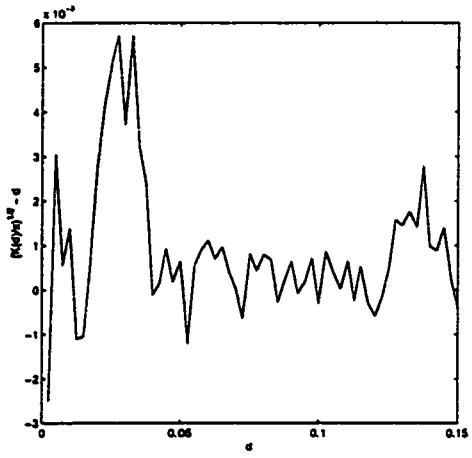
For this process, clearly there is a change in the K-function at the point ρ which defines the inhibition process. Even for the K-function of a mixture process, we expect a change in the behavior of the estimated K-function, since it is a mixture of the inhibited K-function and the uniform K-function. We can estimate ρ by:

$$\arg\min_d \sqrt{\widehat{K(d)}/\pi} - d$$

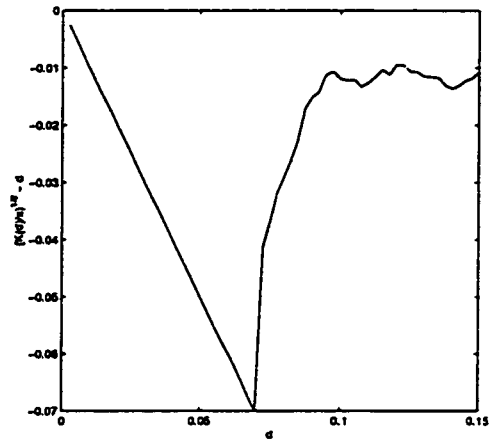
Figure 2.2 shows the K-functions for the spatial point patterns of Figure 2.1. The differences between the plots are apparent: the K-functions of the Strauss process and the mixture process both have sharp drops at $h = \rho$, while the K-function of the Poisson process is stationary with mean zero.

2.5.3 Spatial Tessellations

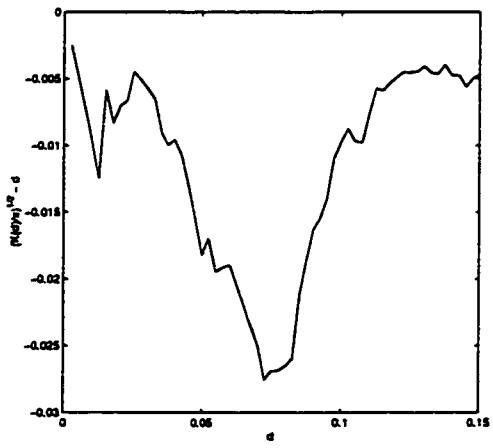
RGG also investigated using spatial tessellations (e.g. see Okabe, Boots, and Sugihara 1992) to distinguish between point process models. The simplest spatial tessellation is the Voronoï tessellation. Here every point in A is associated with the nearest event in A. This results in the study region, A, being partitioned into polygonal tiles (or Voronoï cells) (see Figure 2.3). RGG found the second central moment of the areas of the Voronoï cells to be a good summary statistic.



(a) Uniform process.

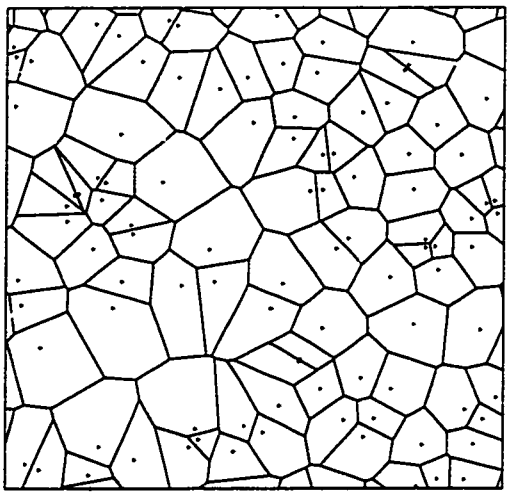


(b) Strauss process.

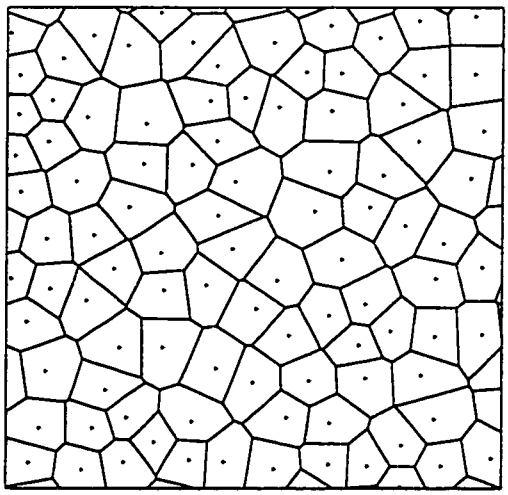


(c) Mixture process.

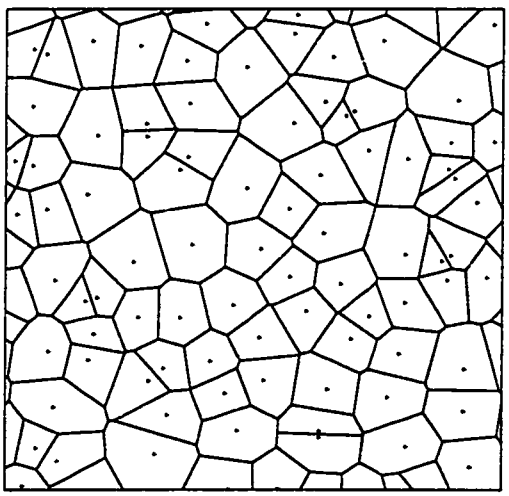
Figure 2.2: K-function plots of each spatial point pattern of Figure 2.1.



(a) Uniform process.



(b) Strauss process.



(c) Mixture process.

Figure 2.3: Voronoï tessellations of each spatial point pattern of Figure 2.1.

2.6 *Simulation Study*

We performed a simulation study to assess the performance of partial Bayes factors in the minefield problem. The simulation study is a simple 2^2 factorial design. The two factors we considered were: the number of noise events, n_0, and the amount of prior information. Various other factors could have been considered, including the number of mines, and the inhibition distance. The parameters used in the simulation study are are given below in Table 2.2.

Table 2.2: Parameters of simulation study.

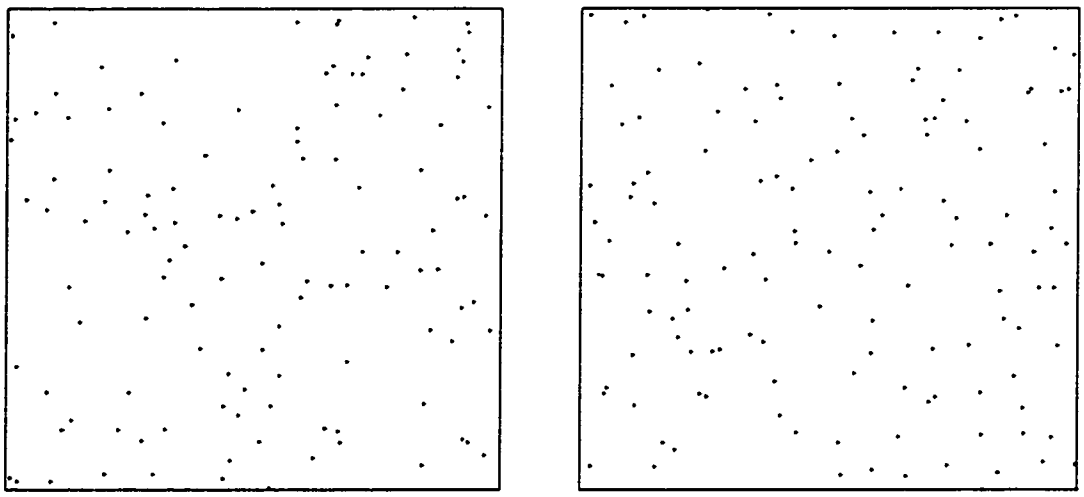
Variable	Value
A	$(0,1)^2$
m	50
n_0	30, 50
ρ	$\frac{1}{2}\rho_{max}$ (0.076)

We shall refer to the different noise levels as being high ($n_0 = 50$) or low ($n_0 = 30$). Typical realizations of each of these spatial point processes are shown in Figure 2.4. As one can see, both of these point patterns are not easily distinguished by eye from a realization of a uniform process.

2.6.1 *Priors*

We decomposed the prior distribution on ρ and m in the following way.

$$\begin{aligned} \pi(\rho, m) &= \pi(\rho \mid m) \times \pi(m) \\ \pi(\rho \mid m) &= \text{Uniform}(\alpha_1 \rho_{max}, \alpha_2 \rho_{max}) \\ \pi(m) &= \text{Discrete Uniform}(\lfloor \beta_1 N, \beta_2 N \rfloor) \end{aligned}$$



(a) Low noise. (b) High noise.

Figure 2.4: Typical minefields of simulation study.

We selected three sets of values for α_1, α_2, β_1, and β_2, that would correspond to "diffuse", "compact", and "tight" priors. These are shown in Table 2.3. We also created a prior corresponding to "perfect" prior information (i.e. a prior with point mass on $m = 50, \rho = \frac{1}{2}\rho_{max}$.

Table 2.3: Parameters of prior distributions.

Prior	α_1	α_2	β_1	β_2
Diffuse	0.3	0.7	0.10	0.90
Compact	0.3	0.7	0.30	0.70
Tight	0.4	0.6	0.40	0.60

2.6.2 *Summary Statistics*

Two different summary statistics were considered to calculate the partial Bayes factors. The first summary statistic was based on the K-function, and the second (due to RGG) on the Voronoï tessellation. We shall refer to them as X_K and X_V, respectively.

$$\begin{aligned} X_K &= \arg\min_d \sqrt{\widehat{K(d)}/\pi} - d \\ X_V &= \text{Second central moment of the areas of the Voronoï cells} \end{aligned}$$

2.6.3 *Edge Effects*

Both of the above statistics can suffer from edge effects. Edge effects occur because events near the boundary have fewer neighbors than they should. We accounted for edge effects by generating all point patterns on a region with a 20% border. Thus, instead of generating a point pattern with N events on the unit square, we generated $\lfloor 1.96 \times N \rfloor$ events on $(-0.2, 1.2)^2$. The factor 1.96 is the ratio of the areas of the two regions.

2.6.4 *Results*

We simulated 100 point patterns on the unit square (accounting for edge effects as described above) under each hypothesis, and for each value of n_0 (i.e. a total of 400 datasets). We calculated the partial Bayes factors for each dataset using both summary statistics, X_K and X_V, and the prior distributions given in Table 2.3. The partial Bayes factors are calculated in terms of evidence for H_1 over H_0. The integration was performed using simple numerical quadrature. The misclassification rates are given in Table 2.4. From this table we can see that the total misclassification rate (assuming that each hypothesis is equally likely a priori) was 22% for the partial Bayes factor based on the K-function, and 25% for the partial Bayes factor based

Table 2.4: Simulation study: Percentage of misclassifications.

Prior	Noise Level	K-function		Voronoï	
		H_1	H_0	H_1	H_0
Diffuse	Low	4	40	40	10
Compact	Low	6	32	33	13
Tight	Low	10	31	33	13
Perfect	Low	10	22	34	12
Diffuse	High	13	40	47	14
Compact	High	21	29	35	16
Tight	High	22	29	33	23
Perfect	High	21	27	31	25
Average Error Rate		13	31	35	15
Total Error Rate		22		25	

on the Voronoï tiling. While these total error rates are similar, the partial Bayes factor based on X_K was more successful at correctly classifying minefields than noise processes. The opposite was true of the partial Bayes factor based on X_V. As one would expect, an increase in the amount of (correct) information contained in priors improves the discrimination in both cases.

The histograms of these partial Bayes factors are shown in Figures 2.5, 2.6, 2.7, and 2.8. These plots are summarized in Tables 2.5, 2.6, 2.7, and 2.8. The results for each statistic are summarized below:

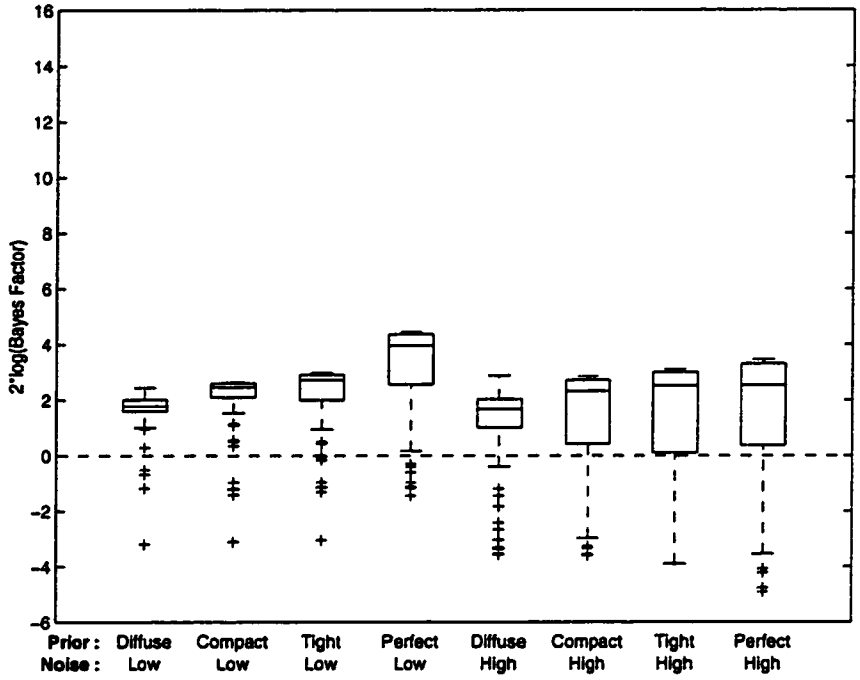


Figure 2.5: Partial Bayes factors for H_1, based on X_K, under H_1.

Table 2.5: Breakdown of partial Bayes factors for H_1, based on X_K, under H_1.

Prior	Noise Level	% Evidence For H_0				% Evidence For H_1			
		Decisive	Strong	Positive	Weak	Weak	Positive	Strong	Decisive
		$(-\infty,-10]$	(-10,-5]	(-5,-2]	(-2,0]	(0,2]	(2,5]	(5,10]	$(10,\infty)$
Diffuse	Low	0	0	1	3	64	32	0	0
Compact	Low	0	0	1	5	15	79	0	0
Tight	Low	0	0	1	9	14	76	0	0
Perfect	Low	0	1	0	9	10	80	0	0
Diffuse	High	0	0	8	5	61	26	0	0
Compact	High	0	0	8	13	21	58	0	0
Tight	High	0	0	9	13	18	60	0	0
Perfect	High	0	0	10	11	21	58	0	0

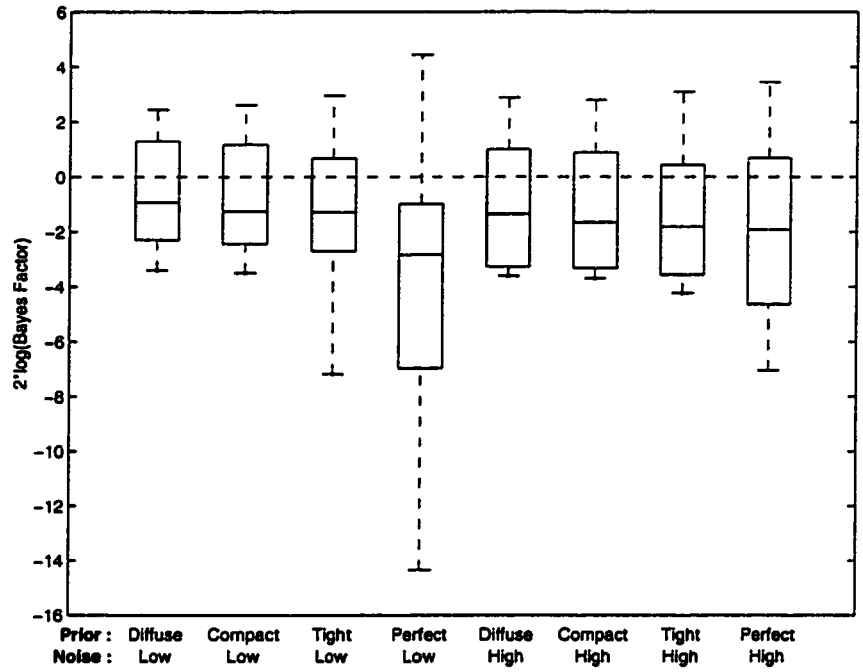


Figure 2.6: Partial Bayes factors for H_1, based on X_K, under H_0.

Table 2.6: Breakdown of partial Bayes factors for H_1, based on X_K, under H_0.

		% Evidence For H_0				% Evidence For H_1			
Prior	Noise Level	Decisive	Strong	Positive	Weak	Weak	Positive	Strong	Decisive
		$(-\infty,-10]$	(-10,-5]	(-5,-2]	(-2,0]	(0,2]	(2,5]	(5,10]	$(10,\infty)$
Diffuse	Low	0	0	28	32	34	6	0	0
Compact	Low	0	0	34	34	16	16	0	0
Tight	Low	0	1	36	32	15	16	0	0
Perfect	Low	11	23	25	19	6	16	0	0
Diffuse	High	0	0	39	21	35	5	0	0
Compact	High	0	0	39	32	20	9	0	0
Tight	High	0	0	46	25	18	11	0	0
Perfect	High	1	19	30	23	17	10	0	0

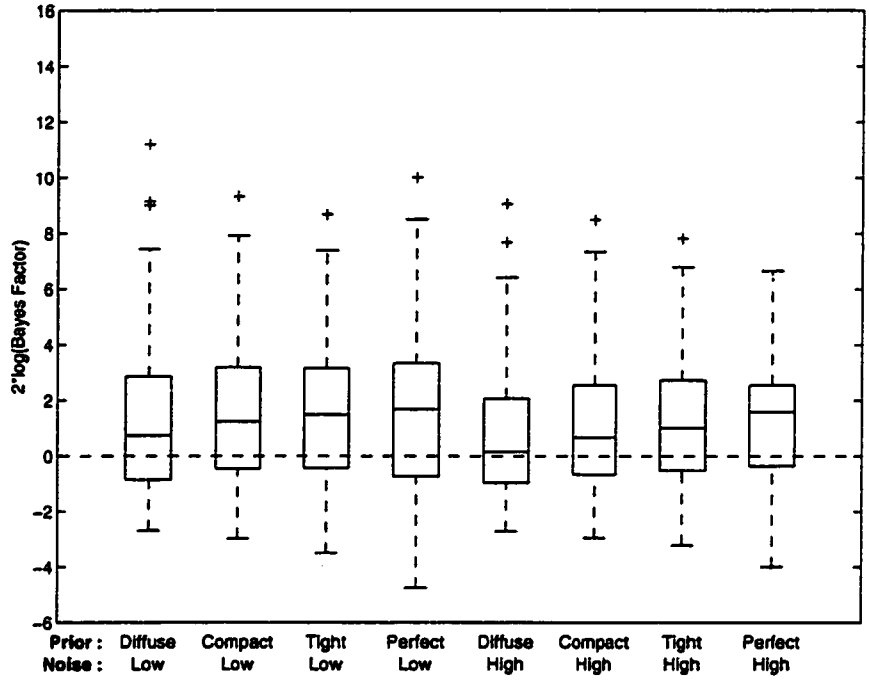


Figure 2.7: Partial Bayes factors for H_1, based on X_V, under H_1.

Table 2.7: Breakdown of partial Bayes factors for H_1, based on X_V, under H_1.

		% Evidence For H_0				% Evidence For H_1			
Prior	Noise Level	Decisive $(-\infty,-10]$	Strong (-10,-5]	Positive (-5,-2]	Weak (-2,0]	Weak (0,2]	Positive (2,5]	Strong (5,10]	Decisive $(10,\infty)$
Diffuse	Low	0	0	5	35	22	25	12	1
Compact	Low	0	0	7	26	24	30	13	0
Tight	Low	0	0	6	27	21	33	13	0
Perfect	Low	0	0	7	27	19	34	12	1
Diffuse	High	0	0	5	42	27	21	5	0
Compact	High	0	0	5	30	33	27	5	0
Tight	High	0	0	6	27	32	30	5	0
Perfect	High	0	0	6	25	29	37	3	0

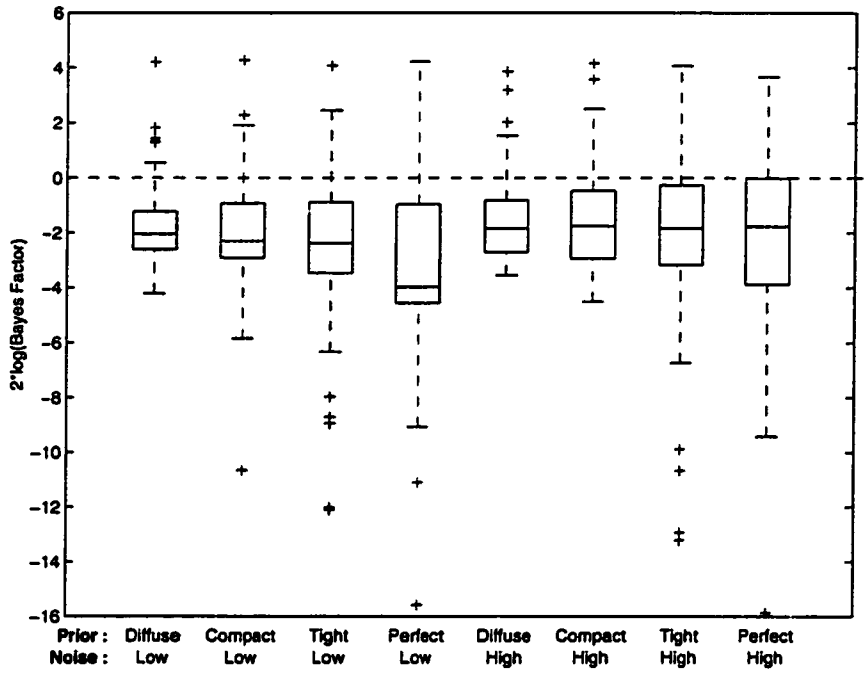


Figure 2.8: Partial Bayes factors for H_1, based on X_V, under H_0.

Table 2.8: Breakdown of partial Bayes factors for H_1, based on X_V, under H_0.

		% Evidence For H_0				% Evidence For H_1			
Prior	Noise Level	Decisive $(-\infty,-10]$	Strong (-10,-5]	Positive (-5,-2]	Weak (-2,0]	Weak (0,2]	Positive (2,5]	Strong (5,10]	Decisive $(10,\infty)$
Diffuse	Low	0	0	54	36	9	1	0	0
Compact	Low	1	4	54	28	11	2	0	0
Tight	Low	5	7	47	28	8	5	0	0
Perfect	Low	18	3	46	21	7	5	0	0
Diffuse	High	0	0	47	39	11	3	0	0
Compact	High	0	0	47	37	12	4	0	0
Tight	High	7	2	39	29	17	6	0	0
Perfect	High	4	4	40	27	18	7	0	0

X_K:

- Under H_1, the Bayes factors typically provide weak to positive evidence for the minefield hypothesis.

- Under H_0, the Bayes factors range from positive evidence for H_1 to positive for H_0 (actually decisive in the case of perfect prior knowledge). The median Bayes factor though, is approximately 1, in favor of the correct hypothesis.

- The increase in noise had a small negative effect on the performance of the Bayes factors under both hypotheses.

X_V :

- Under H_1, the Bayes factors range from positive evidence for H_1 to strong evidence for H_0 (actually decisive in the case of perfect prior knowledge). The median Bayes factor though, is again approximately 1 in favor of the correct hypothesis.

- Under H_0, the evidence for H_0 ranges from weak to positive (actually decisive in the case of perfect prior knowledge).

- There is a negligible negative effect in the performance of the Bayes factors due to the increase in noise.

2.7 COBRA Minefield Data

We applied the partial Bayes factors methodology to three minefield datasets. The datasets, referred to as Eglin, Lejeune, and Surf Zone, are part of the COBRA project which is explained in detail in chapter 3, section 3.2. The datasets can be summarized as having high noise levels and moderate inhibition distances. Also, the mines in each dataset form fairly linear rows.

Since the minefield models presented in this chapter do not model linear structures explicitly, applying the partial Bayes factor methodology directly these datasets may not be fruitful. However, the mines are more spatially inhibited than the noise points, therefore it is possible that the partial Bayes factor will detect this.

Table 2.9: Partial Bayes factors for H_1, for the COBRA datasets.

Dataset	K-function			Voronoï		
	Diffuse	Compact	Tight	Diffuse	Compact	Tight
Eglin	-3.04	-3.47	-4.87	0.83	1.54	1.87
Lejeune	-1.58	-2.01	-2.34	-2.04	-2.99	-3.13
Surf Zone	-0.93	-1.65	-2.23	-0.48	-0.05	0.24

We computed the partial Bayes factors for each dataset, using both test statistics, and "diffuse", "compact", and "tight" priors. The results are shown in Table 2.9. All Bayes factors give weak evidence for the null hypothesis, except the bayes factors for the Eglin dataset, based on the Voronoï statistic. In this case the Bayes factors provide weak evidence for the minefield hypothesis.

2.8 Discussion

In this chapter we investigated the feasibility of using partial Bayes factors to classify mixtures of spatial point processes. We limited out attention to two different summary statistics with which to calculate the partial Bayes factors from. One summary statistic was based on the K-function, and the other on the Voronoï tessellation. We performed a simulation study, and found that partial Bayes factors based on both statistics provided good discrimination between the competing hypotheses we considered. When applied to the COBRA datasets, the Bayes factors, on the whole provided

weak evidence against the existence of a minefield. This wasn't unexpected, given the high noise levels in these datasets. A different approach to analyzing these data is explored in chapter 3.

An approach to this problem using a supervised pattern recognition scheme based on summary statistics of the point pattern was developed by Raghavan, Goel, and Ghosh (1998). Their approach had the advantage of being able to easily incorporate as many summary statistics as one could think of; to do that using partial Bayes factors would require high dimensional density estimation which can get notoriously difficult. Our approach did have the advantage though, of having a natural framework with which to include prior information about each competing hypothesis.

In this chapter we limited our attention to the problem of classifying spatial point processes. It is quite possible that the partial Bayes factor methodology would be applicable in other situations. This is an area of future research.

Chapter 3

DETECTING MINES IN MINEFIELDS WITH LINEAR CHARACTERISTICS

3.1 Introduction

The problem of detecting minefields, and individual mines, is of great interest to civilians and military alike. One application is the identification of mines and minefields for demining in peacetime after the end of conflicts, particularly in countries that have been extensively mined such as Egypt, Cambodia and Bosnia. Another application is the identification of mined areas for avoidance and/or possible demining in the context of military operations.

There are several ways of detecting mines, and one that shows great promise is aerial reconnaissance. An aircraft flies over an area and images it; the images are then analyzed off-line or in real time to detect minefields. This method allows one to survey large areas at relatively low cost and, in particular, at low risk to the personnel involved. One system for doing this is described in the next section.

There are technical problems with this methodology, however. A critical one is that the images produced are often noisy, and mines are hard to identify unambiguously in them. As a result, there is a first stage of image processing in which potential mines are identified in the images. While this does often succeed in identifying many of the mines, it also tends to identify many other objects as possible mines, if they "look" somewhat like mines to the initial image processing algorithm. Examples of such objects include other metal or plastic objects. The result of this image processing is a spatial point pattern of possible mine locations. The problem then arises of

identifying actual mines in a spatial point pattern containing both mines and clutter. This is the problem that we address in this chapter.

It is important to distinguish between two kinds of problem of this type. The first is the "field of regard" problem, in which a large area is surveyed aerially, large enough that one can assume that any minefield in the area is wholly or partly contained within the image (Muise and Smith 1992). Then the analyst's goal is to divide the image into two parts: one, with a low intensity spatial point pattern in which only clutter is present, and the other, with a higher intensity point process, in which both mines and clutter are present. This problem can be solved by assuming that the data are generated by a mixture of two spatial Poisson processes with different intensities and performing statistical inference for this model. Solutions to this problem under different assumptions have been proposed by Allard and Fraley (1997), Byers and Raftery (1997, 1998), Stanford and Raftery (1997), Dasgupta and Raftery (1998), and Fraley and Raftery (1998).

The methods proposed worked well, with high detection rates and low false positive rates. Allard and Fraley (1997) showed that the performance of the methods is quite robust to departures from the Poisson assumption.

Here we address the other kind of problem, the "field of view" problem, in which the area surveyed is much smaller and is likely to be completely or largely contained within the minefield, if a minefield is present. The goal here is to ascertain whether or not a minefield is present, and if so, identify which points are mines. This is a much harder problem, because we cannot use a contrast between intensity levels to solve it. Also, unlike many image analysis problems, it baffles the human eye, which typically cannot pick out mines in such an image. Figure 3.1 shows one point pattern that we analyzed. The noise level is considerable and the location of the mines is not apparent to the eye.

The minefields that we are concerned with are characterized by nearly parallel, almost linear rows of roughly evenly spaced mines. There are many reasons to expect

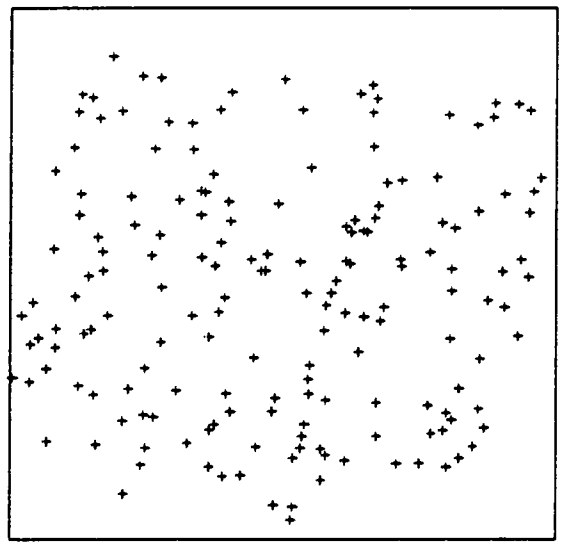
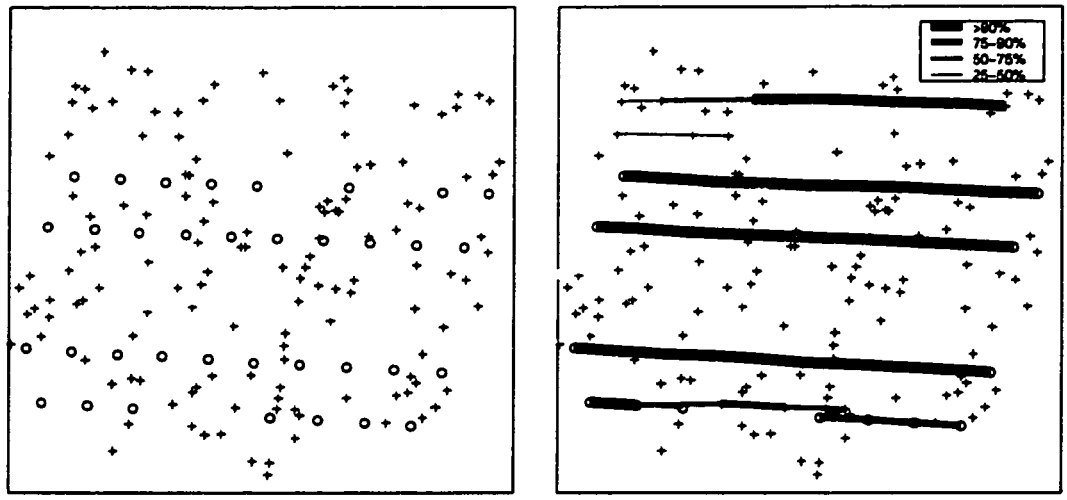


Figure 3.1: Eglin Air Force base data - Mines hidden.

this type of minefield configuration. If the mine-laying force wishes to be able to remove the mines later, either to facilitate its own operations or to avoid harming civilians, they will be able to locate the mines they laid more easily if they have been laid in a regular pattern. A second reason is efficient use of assets: the same number of mines can cover a greater area if laid in a regular pattern than in a random or clustered pattern. A third reason is that mines are often laid by machines, which will tend to operate in a regular way. Lake and Keenan (1995) summarized nicely why such mine laying procedures are plausible:

> From the point of view of military doctrine (e.g. place mines in nn rows xx meters apart), tactical efficiency (e.g. economic use of resources to ensure coverage against enemy movement), and from inherent limitations in the mine laying process (e.g. safety, inability of humans to emulate a truly random process), it is certainly plausible to hypothesize minefield models which exhibit collinearity, equal spacing, and/or other forms of regularity.



(a) Eglin Air Force base data - Mines shown (o = mines, + = noise).

(b) Edge Classification - Eglin Air Force base. The width of the edges are proportional to their posterior probabilities.

Figure 3.2: MCMC Results - Eglin Air Force base.

Furthermore Muise and Smith (1995) developed an algorithm for detecting patterned minefields for the reason that ...

> ... under the assumption that the minefield has been emplaced such that the mines are laid in a linear pattern, which is a very common mine laying doctrine.

Here we present a method for detecting mines from aerial reconnaissance images in the "field of view" situation. We assume that the images have been preprocessed to produce a collection of points, each one representing the location of a possible mine. Our task is to distinguish between the mines and the false alarms (hereafter referred to as noise or clutter). We develop a flexible model for this type of minefield, namely the sequential placement model. The goal of our approach is to obtain posterior

probabilities of each point being a mine. We estimate these probabilities and the parameters of the model via Markov chain Monte Carlo in a fully Bayesian framework.

The point pattern from Figure 3.1 is shown in Figure 3.2(a) with the mines identified. The results of applying our method to it are shown in Figure 3.2(b). For these data, the method worked well. The rows of mines were all correctly identified. One false row was detected.

In the next section we describe the datasets we analyzed. In Section 3.3 we present the sequential placement model. Section 3.4 outlines the Markov chain Monte Carlo algorithm used, while results from applying it to the datasets are presented in Section 3.5. Section 3.6 contains discussion of the model and possible avenues of future work.

3.2 COBRA Data

The Coastal Battlefield Reconnaissance and Analysis (COBRA) program, developed by the U.S. Marine Corps, is intended to detect minefields in coastal areas before troop deployment. An unmanned aerial vehicle, fitted with multi-spectral video cameras, is flown over the area of interest (see Figure 3.3). The video images are either stored or transmitted for processing and analysis. A more detailed description can be found in Witherspoon, Holloway Jr., Davis, Miller, and Dubey (1995).

Eglin Air Force base and Camp Lejeune

A COBRA image database was obtained from the Navy's Coastal Systems Station, Dahlgren Division, Panama City, Florida. Raw images, ground truth, and the results of their own image processing method were available for two different sites, Eglin Air Force base, Florida and Marine Corps base, Camp Lejeune, North Carolina.

The test site at Eglin was an environment cluttered with clumps of grass and patches of sand. A COBRA image from there is shown in Figure 3.4. The imaged

Figure 3.3: COBRA operational concept.

area is approximately 30 meters across. The test site at Lejeune was a moderately cluttered coastal area (see Figure 3.5).

The image processing technique used (described in Holmes, Schwartz, Seldin, Wright, and Witter 1995), searched for local anomalies in the multi-band images based on a measure of spectral-spatial contrast. This algorithm had been applied to the database, and the top 25 targets for each image were identified.

At each site there were 10 overlapping images available. We aligned the images by eye in order to obtain the relative horizontal and vertical displacements of neighboring images. We then concatenated the top 25 targets from each image into a spatial point pattern on a common scale. Given the typical distance between targets, duplicates (target that were detected in multiple images) were easy to identify and were removed.

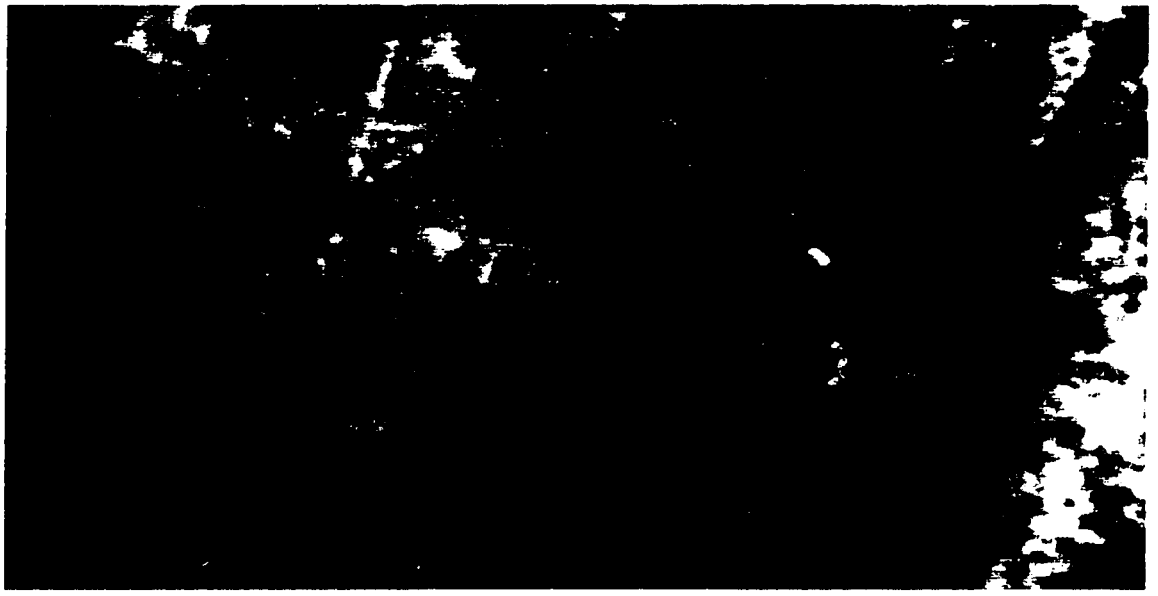


Figure 3.4: COBRA image BB081037, Band #1 , Right Camera - Eglin Air Force base.

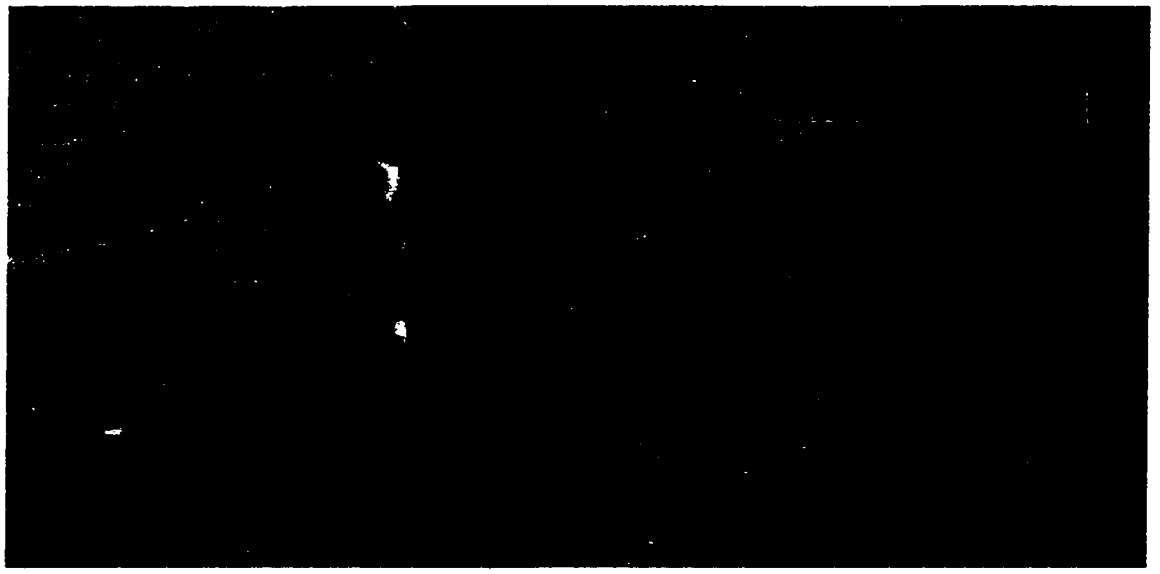


Figure 3.5: COBRA image LB560069, Band #1 , Left Camera - Camp Lejeune.

In the Eglin dataset shown in Figure 3.2(a), the image processing identified 173 points, of which 35 are mines and 138 are noise points. The Lejeune dataset is shown in Figure 3.6(a). Here 168 points were identified, of which 28 were mines and 140 were noise points.

Surf Zone Data

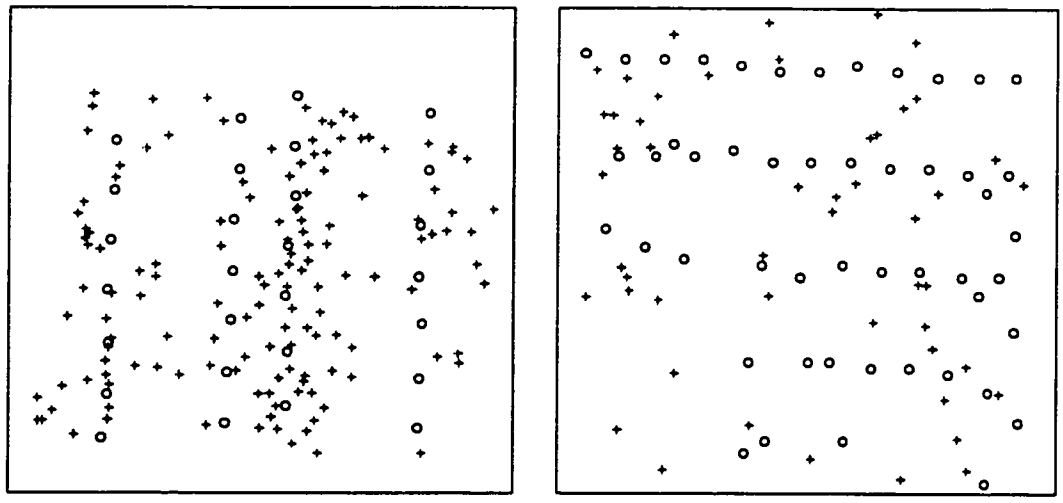
This dataset was produced by processing one spectral band from a multi-spectral camera showing a ground-based scene of mines laid out in a regular pattern on a surf zone (Lake and Keenan 1995; Lake, Sadler, and Casey 1997). The processed image contained 50 points, which they considered to be clutter-free (i.e. all mines). To test their own minefield detection algorithms they added 50 random points. This dataset is shown in Figure 3.6(b). For simplicity, all the point patterns we analyzed have been scaled to the unit square.

The Realism of the COBRA Datasets

The question may be asked, "How well do these datasets represent *real* minefields?" We feel that given the information available to us, the datasets we described above are representative of a *class* of minefields found in the real world. Of course, not all minefields will belong to this class. Past experience, and knowledge of the mine-layer, will determine whether or not this particular class of minefield is the most likely to have been laid.

3.3 A Model for Mine Locations: The Sequential Placement Model

In this section, we present a flexible model for mine locations, the sequential placement model. The basic idea of this model is that the mines are laid sequentially roughly a constant distance apart in approximately parallel rows; earlier we explained why this is a reasonable expectation. The distances between sequential mines, the mean



(a) Camp Lejeune data - Mines shown (o = mines, + = noise).

(b) Surf Zone data - Mines shown (o = mines, + = noise).

Figure 3.6: COBRA datasets.

distances between rows and the direction of the rows are parameters to be inferred.

Throughout this chapter we shall adopt the following notation: Let A be the sample space or study region, and $|A|$ denote the area of this region. Let N be the total number of points, n_0 be the number of noise points, and m be the number of mines. The data (coordinates) of all points are denoted by y.

The n_0 noise points are assumed to be distributed uniformly throughout the study region. This is a simplifying assumption which we have found to work well. It may be though that the noise points have there own underlying structure which could be due to, e.g. the terrain, trees, the surf. Although possibly more difficult, incorporating such knowledge could be beneficial.

Suppose that the minefield consists of K rows, and that the ith row contains n_i

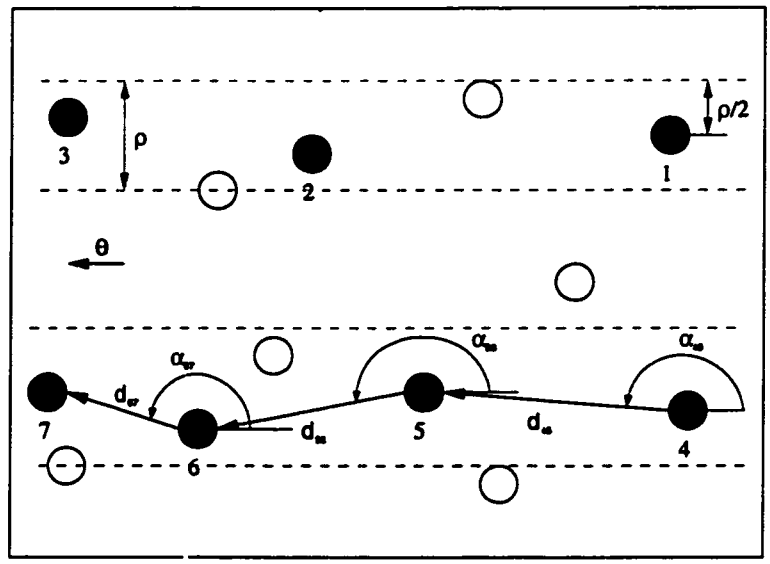


Figure 3.7: The sequential placement model (Solid circles represent mines. Open circles represent noise points. Dashed lines represent bands).

mines. Thus we have,

$$N = n_0 + m = \sum_{i=0}^{K} n_i \qquad \text{and} \qquad m = \sum_{i=1}^{K} n_i.$$

The mines in each row are ordered and the position of each mine depends on the position of the preceding mine. Let d_{ij} be the Euclidean distance from the i^{th} point to the j^{th} point, and let α_{ij} be the corresponding angle, measured in radians, with horizontal right as a baseline (0 radians). We shall use the notation $i \rightarrow j$ if the i^{th} and j^{th} points are both mines and point j is the next mine after point i, in the same row. We assume that these intermine distances and angles are distributed in the following manner:

$$d_{ij} \sim \text{Normal}(\mu, \sigma^2), \text{ if } i \rightarrow j \tag{3.1}$$

$$\alpha_{ij} \sim \text{vonMises}(\theta, \kappa), \text{ if } i \rightarrow j \tag{3.2}$$

A schematic of the sequential placement model is shown in Figure 3.7.

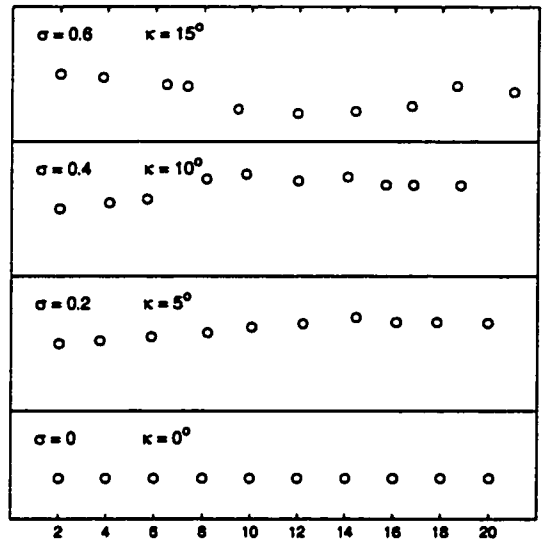


Figure 3.8: Realizations from the sequential placement model - $\mu = 2$, $\theta = 0^o$ (Horizontal right), κ is converted to degrees for convenience.

The parameters μ and σ are the mean and standard deviation of the inter-mine distances, and θ and κ are the mean and standard deviation of the inter-mine angles. To prevent rows from overlapping, we add one constraint. Each row has a "band" of width ρ within which each mine in the row must lie. All the bands have the same orientation θ and the only restriction on the bands is that none of them may overlap. Each band is centered (perpendicular to θ) on the first mine of each row. Finally each row is assumed to have at least three mines. This is a practical constraint to ensure that the model favors linear features.

Figure 3.8 a range of possible minefields that can be generated with this model. The plot shows four rows with 10 mines each. Each row has the same μ and θ, but σ and κ increase from the bottom row to the top row. The bottom row has both standard deviations zero, so that all mines lie on a straight line, and consecutive mines are the same distance apart.

Table 3.1: Parameters of the sequential placement model.

μ	Mean inter-mine distance
σ	Standard deviation of inter-mine distance
θ	Mean inter-mine angle
κ	Standard deviation of inter-mine angle
ρ	Row repulsion radius
λ	Mean number of rows
ν	Mean number of noise points per unit area
ν_0	Mean number of mines per line

Parameters

The parameters of the model are described in the Table 3.1. All the parameters are univariate and shall be denoted by $\Theta = (\mu, \sigma, \theta, \kappa, \rho, \lambda, \nu, \nu_0)$.

Latent Variables

The identities of the mines and of the orderings of the mines within each row shall be given by the latent variable, Z; an $N \times N$ matrix, whose individual entries are given by:

$$Z_{ij} = \begin{cases} 1, & \text{if } i \rightarrow j. \\ 0, & \text{otherwise.} \end{cases}$$

The variable Z can be thought of as a parameter about which we wish to make inference.

Likelihood

The likelihood for the data y and latent variable Z can be written as:

$$P(y, Z \mid \Theta, N, A) = P(y \mid Z, \Theta, N, A)P(Z \mid \Theta, N, A)$$

We condition on the number of points (N) and the area of the study region (A). It is assumed that there is always at least one row of mines present.

The likelihood for the data given Z, $P(y \mid Z, \Theta, N, A)$, given below, is simply a product of the likelihoods of the lengths and the directions of each edge in the minefield, with a term to account for the noise points and a term to ensure that the minefield configuration is allowable. Here Φ_N is the Normal (Gaussian) density function, and Φ_{VM} is the von Mises density function.

$$P(y \mid Z, \Theta, N, A) = \prod_{Z_{ij}=1} \Phi_N(d_{ij} \mid \mu, \sigma)\Phi_{VM}(\alpha_{ij} \mid \theta, \kappa)1(y, Z, \rho)\frac{1}{|A|^{n_0}},$$

$$\text{where } 1(y, Z, \rho) = \begin{cases} 1, & \text{If all mines lie inside the correct bands,} \\ & \quad \text{and no bands overlap.} \\ 0, & \text{otherwise.} \end{cases}$$

The fact that we are looking for roughly linear structures implies that we expect the standard deviation of the inter-mine angles to be relatively small. Consequently we parameterized the von Mises distribution in terms of the standard deviation (κ), rather than the usual concentration parameter. This allowed us to place priors that included the lower bound of κ ($\kappa = 0$), which corresponds to a "perfectly laid" minefield. Thus the von Mises density had the form:

$$\Phi_{VM}(x) = \frac{1}{2\pi I_0(\frac{1}{\kappa^2})}\exp(\frac{1}{\kappa^2}\cos(x - \theta))$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind of order zero.

The likelihood for Z (sometimes called an induced prior since it doesn't depend on the data, y) is factored as follows:

$$\begin{aligned} P(Z \mid \Theta, N, A) &= P(Z \mid \lambda, \nu_0, \nu, N, A) \\ &= P(K, n_0, \ldots, n_K \mid \lambda, \nu, \nu_0, N, A) \\ &= P(n_0, \ldots, n_K \mid K, \nu, \nu_0, N, A) P(K \mid \lambda, N). \end{aligned}$$

This prior is uniform over all minefields with the same number of rows, and same number of mines per row. $P(n_0, \ldots, n_K \mid \cdot)$ is a Multinomial distribution, truncated to ensure that all rows have at least three mines, namely:

$$\begin{aligned} P(n_0, \ldots, n_K \mid K, \nu, \nu_0, N, A) &\propto \binom{N}{n_0 \ldots n_K} \left(\frac{\nu_0 |A|}{\nu_0 |A| + K\nu} \right)^{n_0} \\ &\quad \times \prod_{i=1}^{K} \left(\frac{\nu}{\nu_0 |A| + K\nu} \right)^{n_i} 1_{\mathcal{A}}(n_1, \ldots, n_K) \\ &\propto \binom{N}{n_0 \ldots n_K} \frac{\nu^m \nu_0^{n_0} |A|^{n_0}}{(\nu_0 |A| + K\nu)^N} 1_{\mathcal{A}}(n_1, \ldots, n_K), \end{aligned}$$

where $\mathcal{A} = \{(n_1, \ldots, n_K) : \min\{n_1, \ldots, n_K\} \geq 3\}$.

The prior on the number of rows is a Poisson distribution, truncated such that the number of rows is at least one and no more than N/3, namely

$$P(K \mid \lambda, N) \propto \frac{\lambda^K e^{-\lambda}}{K!} 1_{\mathcal{B}}(K),$$

where $\mathcal{B} = \{K : 1 \leq K \leq \lfloor \frac{N}{3} \rfloor\}$.

3.3.1 Simulation from the Sequential Placement Model

A spatial point process is "any stochastic mechanism which generates a countable set of events in the plane" (Diggle 1983). In order for the sequential placement model to be a valid point process, it must be possible to generate realizations from it. A simple procedure to accomplish this is *rejection sampling*. Using this, a realization from the sequential placement model with parameters Θ can be generated as follows:

1. Sample the number of rows, K, from $P(K \mid \lambda, N)$.

2. Sample the number of mines per row, and the number of noise points, $n_0, \ldots, n_K$ from $P(n_0, \ldots, n_K \mid K, \nu, \nu_0, N, A)$.

3. Sample K points uniformly from A. These are the locations of the first mines in each row (the starting mines).

4. For each value i from 1 to K, draw n_i samples from the inter-mine distance and angle distributions (Equations 3.1 and 3.2). Use these distances and the locations of the starting mines to create the locations the mines.

5. Sample n_0 points uniformly from A. These are the locations of the noise points.

6. Concatenate the locations of the mines and the noise points to form the data, y

7. Calculate the value of the indicator function, $1(y, Z, \rho)$. If this is 1, then stop; else return to step 1

Obviously the probability of acceptance a sample is highly dependent on the location of the starting mines. A hard-core Strauss process (Strauss 1975) with inhibition distance equal to ρ would be a simple improvement to make.

3.4 Bayesian Estimation via Markov Chain Monte Carlo

We now describe a fully Bayesian approach for estimation of the sequential placement model. We use a Markov chain Monte Carlo (MCMC) algorithm to sample from the posterior distribution. See, e.g. Gilks, Richardson, and Spiegelhalter (1996), for an introduction to MCMC methods. The MCMC algorithm used is a Metropolis-Hastings algorithm (Hastings 1970). The prior and proposal distributions we use are described below.

3.4.1 Priors

The prior is decomposed as follows:

$$\begin{aligned}\pi(\Theta \mid N, A) &= \pi(\mu, \sigma, \theta, \kappa, \rho, \lambda, \nu, \nu_0 \mid N, A) \\ &= \pi(\mu)\pi(\sigma)\pi(\theta)\pi(\kappa)\pi(\rho)\pi(\lambda)\pi(\nu)\pi(\nu_0)\end{aligned}$$

This decomposition is chosen mostly for simplicity. All priors have Uniform distributions.

3.4.2 Proposal distributions

For the non-negative-valued univariate parameters (σ, κ, ρ, λ, ν, ν_0), a new value was proposed by multiplying the current value of the parameter by $\exp(U)$ where $U \sim \text{Normal}(0, \tau^2)$. Note that τ can be different for each parameter and should not be too large, as otherwise few moves will be accepted.

For μ a new value was proposed by simply adding a Normal random variable to its current value. A similar move for θ involving the addition of a von Mises random variable can be used. However this leads to the calculation of the acceptance ratio becoming computationally expensive, since it involves the ratio of two modified Bessel functions. In the actual implementation of our MCMC algorithm we proposed a new

value for θ by adding a truncated Normal distribution (with the same mean and variance as the von Mises distribution) to the current value (modulo 2π).

For the edge matrix Z, six different types of proposals are used; Add, Delete, Swap, Grow, Kill and Jump. At any given iteration of the MCMC algorithm, either all the univariate parameters are proposed to be updated (individually, in random order), or one of the edge matrix moves is proposed. Which type of move is proposed depends on a specified probability vector $p_{\text{Prop}} = (p_{\text{Par}}, p_{\text{Add}}, p_{\text{Delete}}, p_{\text{Swap}}, p_{\text{Grow}}, p_{\text{Kill}}, p_{\text{Jump}})$. The different edge proposals are described below and represented schematically in Figure 3.9. The first five moves are designed to explore the current mode of the posterior locally. The Jump move is intended to enable the chain to make large jumps from one posterior mode to another.

Add: An end mine (the first or last mine in a row), point i say, is selected at random. A noise point is then proposed to be added to the end (or beginning) of the row. The probability of proposing noise point j (p_{ij}) is proportional to l_{ij}, where l_{ij} is the likelihood of the new edge created by adding the proposed noise point. That is:

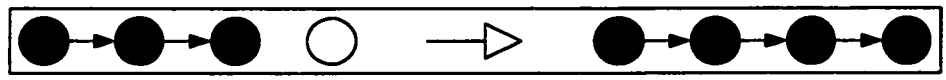
$$p_{ij} \propto l_{ij} = \Phi_N(d_{ij} \mid \mu, \sigma)\Phi_{VM}(\alpha_{ij} \mid \theta, \kappa)$$

Notice how the proposal probabilities depend on the current parameter values. This results in sensible proposals and increases the mixing of the chain.

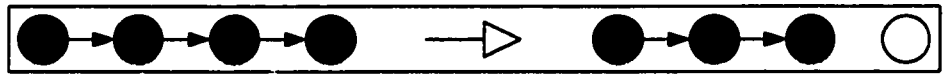
Delete: An end mine is chosen at random. Rows with only 3 mines are ignored. If the move is accepted, the selected end mine is changed to a noise point.

Swap: A mine is chosen at random. A noise point is then proposed to be swapped for this mine. Like *Add*, the probability of proposing a particular noise point is proportional to the likelihood of the new edge(s) that would be created by the replacing the proposed noise point and the selected mine.

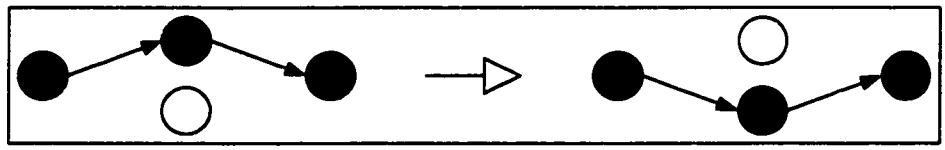
Grow: A noise point is selected at random. Let the location of the noise point be (x', y'). The two noise points closest to $(x' - \mu^* cos\theta^*, y' - \mu^* sin\theta^*)$ and $(x' +$



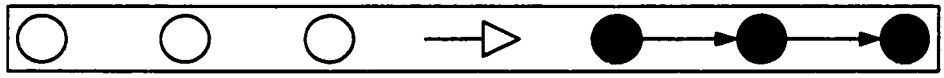
(a) Add



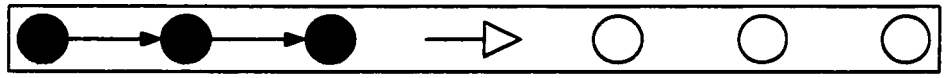
(b) Delete.



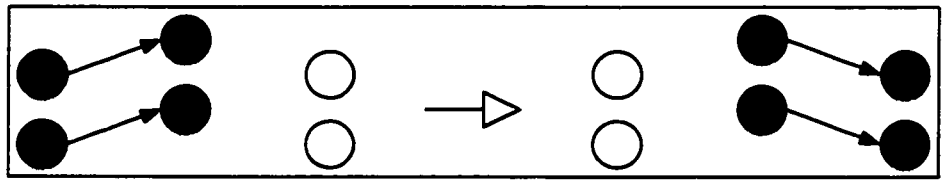
(c) Swap.



(d) Grow.



(e) Kill.



(f) Jump.

Figure 3.9: Schematic representing the result of accepting the different proposal moves for Z. Current minefields are shown on the left and the result of accepting the move is shown on the right. Mines are represented by solid circles while the noise points are represented by open circles. Edges are represented by directed arrows.

$\mu cos\theta, y' + \mu sin\theta)$, where μ^* and θ^* are the current values of μ and θ respectively, are then selected. These three mines are proposed to form a new row.

Kill: A row with exactly 3 mines is selected at random from all rows with exactly 3 mines, if any. If the move is accepted all three mines in this row will be changed to noise points. This move is not proposed if there is only one row in the minefield.

Jump: The Jump move proposes new model parameters and "grows" a new minefield based on these parameters. This move allows the Markov chain to make large jumps throughout the parameter space. The details of the move can be found in Appendix A.

3.4.3 MCMC Implementation

The MCMC algorithm was run on the three datasets. In each case, the chain was run for 1.6 million iterations. This large number was needed due to the slow mixing of the chain. Only every 400^{th} iteration was recorded to reduce storage space. Convergence of the chain to the equilibrium distribution was monitored by running multiple chains from different starting points; it was observed that they converged to the same solution. More formally, the Gelman-Rubin R statistic (Gelman and Rubin 1992) did not indicate non-convergence.

The `gibbsit` software (Raftery and Lewis 1992, 1996), was used to assess the number of iterations needed to estimate the posterior quantiles of interest after reaching the region of high posterior probability; it verified that the number of iterations we used was sufficient for this purpose. Note that convergence and adequacy of the number of iterations are different issues. Convergence is achieved once the chain has reached the region of high posterior probability for the first time; estimation accuracy depends on the number of iterations after convergence has been achieved.

The chain in each case was initialized by drawing the univariate parameters from the priors and creating an initial minefield with a Grow move. Thus the chain started with one row consisting of three mines.

Table 3.2: Priors used for MCMC algorithm - The priors on κ and θ are given in degrees for convenience.

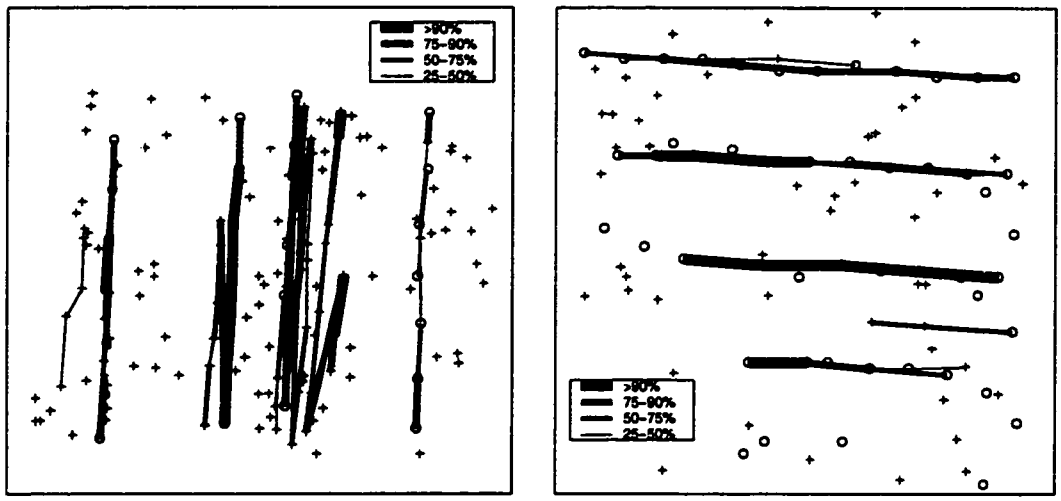
Parameter	Prior
μ	(0.06,0.12)
σ	(0.0,0.04)
θ (Eglin & Surf Zone)	$(135^o, 225^o)$
θ (Lejeune)	$(45^o, 135^o)$
κ	$(0, 1.5^o)$
ρ	(0.08,0.12)
λ	(2,4)
ν	(10,15)
ν_0	(30,150)

For the chain to mix well the priors used had to be somewhat informative. We attempted to create priors for each dataset that might correspond to knowledge that an expert would have. In particular we assumed that the terrain of the imaged region would provide some prior knowledge about the direction of the minefield. Thus we used priors on θ with width 90^o in each case. Note that a totally uninformative prior on θ would have width 180^o, since a given minefield would be equally well modeled by θ or $\theta + 180^o$. The priors for each dataset were the same (except for θ) and are given in Table 3.2. For all simulations $p_{\text{Prop}} = (0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.10)$.

3.5 Results

Edge Detection

For each dataset we plotted the posterior probabilities of each edge over the point pattern (Figures 3.2(b), 3.10(a) and 3.10(b)). We can see that our approach, for the most part, has been successful in identifying the mines.



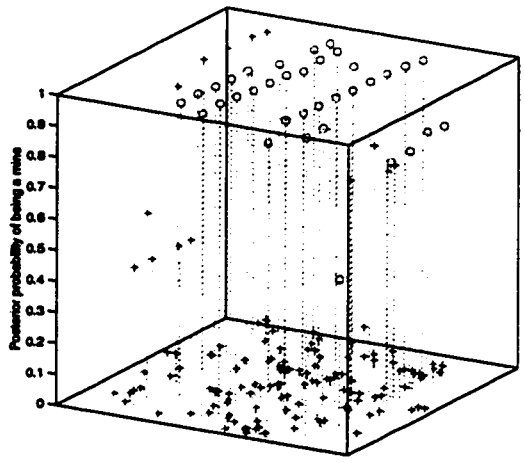
(a) Edge Classification - Camp Lejeune. The width of the edges are proportional to their posterior probabilities.

(b) Edge Classification - Surf Zone. The width of the edges are proportional to their posterior probabilities.

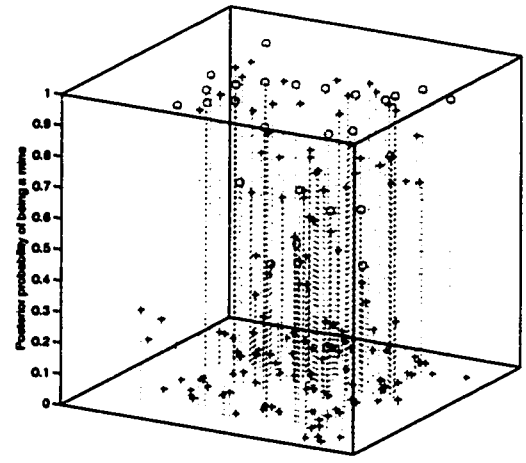
Figure 3.10: MCMC results - Camp Lejeune and Surf Zone datasets.

Mine Detection

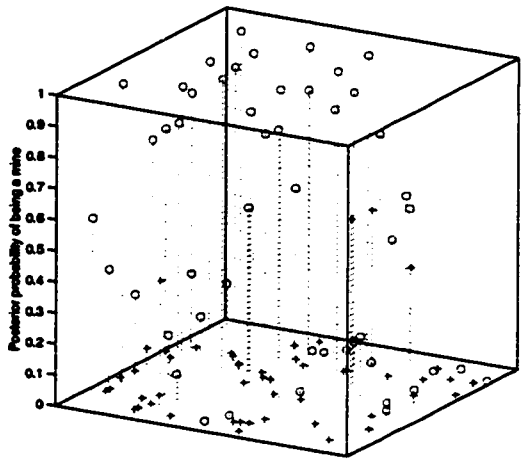
We plotted the posterior probabilities that each point was a mine for each dataset (Figure 3.11(a), 3.11(b),3.11(c)). There is a clear distinction between the mines and the noise for the Eglin dataset. In the Lejeune dataset there are more false positives,



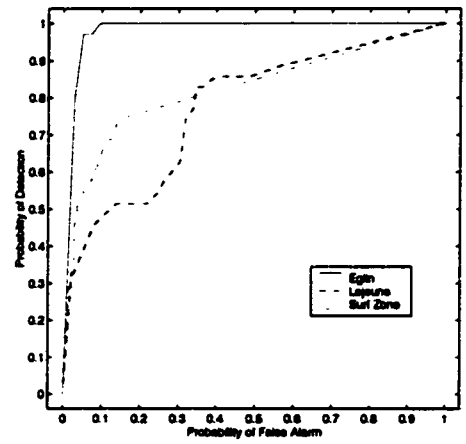
(a) Mine classification - Eglin Air Force base (∘ = Mines, + = Noise).



(b) Mine classification - Camp Lejeune (∘ = Mines, + = Noise).



(c) Mine classification - Surf Zone (∘ = Mines, + = Noise).



(d) Receiver Operating Characteristic (ROC) curves.

Figure 3.11: MCMC results - Mine classification.

Table 3.3: Detection and false positive rates (for a 50% threshold) for each dataset.

Dataset	Detection Rate	False Positive Rate
Eglin	97	7
Lejeune	79	30
Surf Zone	54	4

Table 3.4: Detection and false positive rates (for a 20% threshold) for each dataset.

Dataset	Detection Rate	False Positive Rate
Eglin	100	10
Lejeune	100	42
Surf Zone	66	8

which is to be expected given the higher intensity of noise points around the two center rows. The detection rate for the Surf Zone data is not as high as the others, due to the high variability of the inter-mine angles. However the four rows with appreciable structure are detected, and the false positive rate is low.

These plots are summarized by Receiver Operating Characteristic (ROC) curves in Figure 3.11(d) and tables of detection and false positive rates (Tables 3.3 and 3.4). The ROC curves are created by plotting the detection rate versus the false alarm rate for many different thresholds of the posterior probabilities that each point is a mine. The ideal curve will be vertical from the origin to (0,1) and then horizontal. Again, we can see that the results are very good for the Eglin data while moderately good for the noisy Lejeune and Surf Zone data.

Overall, it is difficult to assess the quality of the results since the cost of a false positive versus a false negative is unclear. However, given the inherent noise level in the datasets, we feel the results are as good as could be expected.

3.6 Discussion

We have presented a model for representing the locations of mines in a minefield characterized by nearly parallel rows of roughly equally spaced mines, in the presence of noise. We also outlined a method for fitting this model via Markov chain Monte Carlo. When applied to real data with a substantial amount of noise, the discrimination between the mines and the noise ranged from excellent (Eglin) to moderately good (Lejeune and Surf Zone).

MCMC methods are computationally intensive. However, our simulations ran reasonably quickly. For instance the Eglin dataset took about 2000 CPU seconds for 1.2 million iterations on a DEC Alpha workstation. The time required is $O(n)$, where n is the total number of points.

Prior selection is an important issue. In practice, there will often be considerable prior information available. For instance, the terrain of the study region will be known, and may influence the orientation of the minefield and the spacing of the mines. The results shown here assume prior information has narrowed the possible orientation angles to a 90^o span. This is rather broad, and in practice more expert information might well be available which would improve the algorithm's performance.

One could choose prior distributions of a completely different form than the one we used. For example, the truncated Poisson distribution for the number of rows may not be suitable for a dataset with a large number of rows, since the mean = variance assumption might be too strong. A uniform distribution with large support may be more suitable.

The Jump proposal move is crucial to the mixing of the chain. While the accep-

tance rate of the Jump move was low, it was usually sufficient to allow the chain to find the dominant mode of the posterior. However, when the chain was run with less prior information (particularly on θ) the chain had difficulty moving between different posterior modes. A more sophisticated technique like simulated annealing or simulated tempering (Kirkpatrick, Gelatt Jr., and Vecchi 1983; Marinari and Parisi 1992; Geyer and Thompson 1995) might help to improve the mixing.

We believe our method to be fairly robust to departures about the assumptions we made concerning the distributions of the inter-mine distances and angles; provided that the variability of these distributions is not large. Our model as it stands in its current form is not robust to departures from the assumption of the linearity and parallel nature of the rows. This is an area for future improvement of the model.

Here we have focused on the post-image-processing phase of mine detection. Improvements in mine detection results could also be achieved by improving the image processing algorithm itself. The MM-MNF algorithm (Banerji and Goutsias 1995; Braga-Neto and Goutsias 1998) has been successfully applied to COBRA images. The false alarm rate in their processed images is much lower than in the point patterns we considered, which were obtained with a widely used algorithm. Clutter does remain, however, so the issues we discuss here are still germane.

Our approach is to model the output of an image processing algorithm as a point process. In doing so, we combine two ideas used in the minefield literature; modeling approximate linearity and regularity of the minefield, and using a Bayesian framework to obtain posterior probabilities of each point being a mine.

Cressie and Lawson (1997, 1998) also fitted a Bayesian model via MCMC; in their case it was a hierarchical point process model. The dataset they analyzed (in Cressie and Lawson 1998) was a subset of the Lejeune dataset. The linear structure of the minefield was not obvious in this smaller dataset, however the mines were more inhibited than the noise points. Their suggested implementation of the model aimed to detect minefields displaying this local inhibition, although their model was

extremely flexible, and one could envision adapting it to detect linear structures.

A different way of detecting approximate linearity and regularity was developed by Lake and Keenan (1995) and Lake, Sadler, and Casey (1997). They developed a two-stage algorithm. First, approximately collinear points are identified using a variant of the Hough Transform (Hough 1962), and then the spacing of the mines is estimated using a modified Euclidean algorithm. Muise and Smith (1995) developed an algorithm for minefield detection that also exploits linear patterns. A critical difference between this approach and ours is that ours is based on the conceptually appealing framework of an explicit statistical model. This leads directly to good estimation methods using established statistical principles, and allows one to understand when the approach is likely to work well and when it is not. It also suggests ways of improving the method's performance in different situations, by modifying the model so as to exploit background knowledge more effectively.

We have assumed throughout that a minefield is present in the region imaged, and that the analyst's task is to detect the mines. Another important question is to determine whether the region surveyed is part of a minefield or not. A fairly straightforward solution to this problem is available by casting it as one of comparing the minefield model that we have developed with a model for clutter only, such as a homogeneous Poisson process. This can be done by computing the Bayes factor (Kass and Raftery 1995) for one model against the other. There are now several effective ways of computing Bayes factors from MCMC output e.g. Raftery 1996; DiCiccio, Kass, Raftery, and Wasserman 1997 and references therein.

Chapter 4

IMPORTANCE SAMPLING AND THE HOUGH TRANSFORM

4.1 Introduction

The Hough transform is a well known technique for detecting parametric curves in images. We place a particular group of Hough transforms (probabilistic Hough transforms) in the framework of importance sampling. This framework suggests an area where probabilistic Hough transforms can be improved: by specifying a target distribution and weighting the sampled parameters accordingly to make identification of curves easier. We investigate the use of clustering techniques to simultaneously identify multiple curves in the image. We also use probabilistic arguments to develop stopping conditions for the Hough transform. We apply our method to both simulated and real data, and compare its performance against two standard Hough transforms. We also apply our method to the COBRA datasets of Chapter 3.

4.1.1 Notation and Definitions

The Hough transform (Hough 1962), hereafter HT, is typically used to detect object boundaries in images. Before applying the HT to a particular image, the image must be transformed by an edge detection algorithm into a binary edge detected image. A edge detection algorithm assigns each pixel to be a *foreground pixel* (*edge pixel*), or a *background pixel*. In all the examples presented here the foreground pixels are black and the background pixels are white. We will use the term *points* to refer to foreground pixels and the term *pixels* to refer to foreground and background pixels.

We define N to be the total number of points.

In order to detect instances of a particular parametric curve in an image, one must decide upon a parameterization of the curve. We will denote the curve parameterization by Θ. The dimension of curve will be denoted by p. For instance, if a line is parameterized in Cartesian coordinates; $p = 2$, and $\Theta = \{m, c\}$, where m and c are the slope and intercept of the line, respectively.

Several of the HTs we discuss utilize an *accumulator array* to detect curves. An accumulator array, denoted by $\mathcal{A}$, is a grid defined on Θ. Each cell of $\mathcal{A}$ is used to store votes for curves whose parameters are contained within the cell.

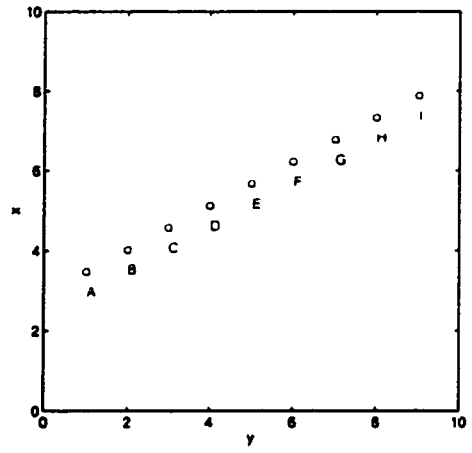
4.1.2 The Standard Hough transform

The HT is a popular method for detecting parametric curves (lines, circles, ellipses etc) in binary image data. By means of transforming the image space into a parameter space, the HT reduces the task of detecting global patterns in the image space to finding local peaks in the parameter space. Good introductions to the HT and comprehensive surveys of the literature can be found in Illingworth and Kittler (1988) and Leavers (1993).

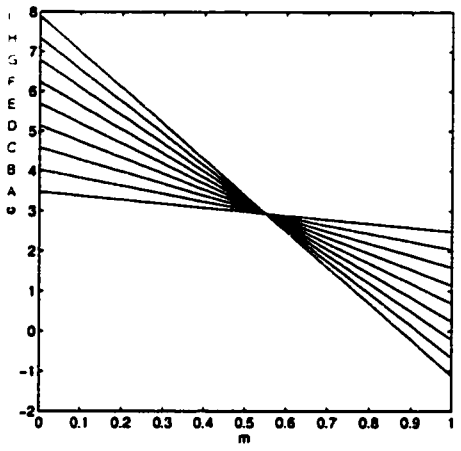
The HT is most easily described by the simple example of detecting the parameters of a straight line that a set of collinear points lie on. Consider the plot of y versus x in Figure 4.1(a). We will refer to this as the image space. Every line that passes through a point in the image space (excluding the case where the line is vertical) can be parameterized as:

$$y = mx + c. \tag{4.1}$$

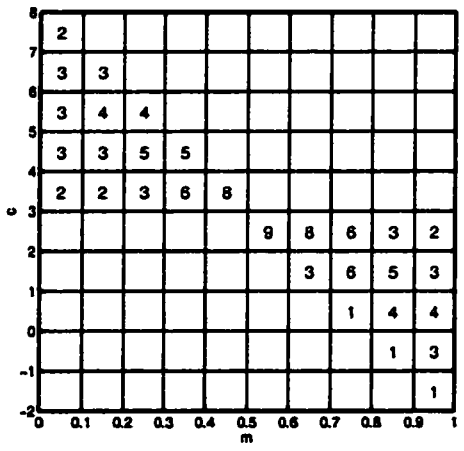
Our parameter space in this case is $\Theta = \{m, c\}$. By rearranging Equation 4.1 in terms of m and c, we can see that all lines that pass through a particular point (x', y') form a line in the parameter space. That is:



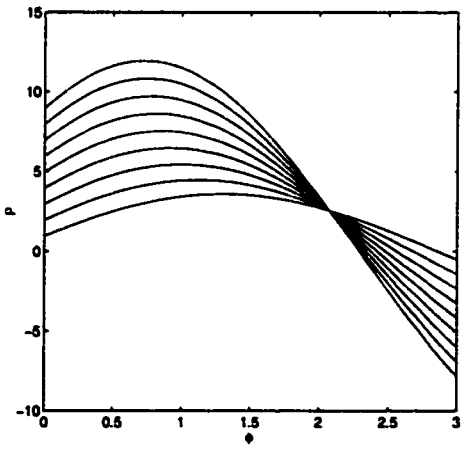
(a) Image space



(b) Parameter space corresponding to $\{m, c\}$ parameterization



(c) Accumulator array corresponding to $\{m, c\}$ parameterization



(d) Parameter space corresponding to $\{\rho, \phi\}$ parameterization

Figure 4.1: Line detection via the standard Hough transform.

$$c = -x'm + y'.$$

Moreover, the lines corresponding to points which are collinear in the image space, will intersect at the same place in the parameter space (see Figure 4.1(b)). Thus the problem of finding the parameters of the line that the points lie on is reduced to detecting the intersection of the lines in the parameter space. Real images, of course, are more complex. The image may contain multiples lines, lines that do exist in the image may not be perfectly straight, and there may be many points not associated with any lines (noise points). In this situation, the goal is to identify regions of high intensity in the parameter space. In practice, the parameter space is discretized into the accumulator array, $\mathcal{A}$, and the number of curves that pass through each cell of the array is recorded (see Figure 4.1(c)). A simple thresholding of this grid will return likely parameter values of the curves present in the image.

There is a problem parameterizing a line via its slope and intercept; namely, that the parameter space is unbounded. Duda and Hart (1972) proposed parameterizing straight lines by the length, ρ, and orientation, ϕ, of the normal to the line from origin of the image. When using this parameterization each point in the image space corresponds to a sinusoid, not a line, in the parameter space (see Figure 4.1(d)). A detailed investigation of the nature of curve parameterization for the HT is given in Yuen and Ma (1997).

4.1.3 The Randomized Hough transform

The main problems of the standard HT (SHT) are its computation time and large storage requirements. The long computation times are caused by the fact that the HT increments the cells in the accumulator array corresponding to *all* curves that pass through *all* points in the image. Thus, much of the computation time is spent storing votes for curves with very little support from the data. The storage requirements

of the standard Hough transform are a bigger problem though, since the size of the accumulator array is exponential in the number of parameters. When the number of parameters is greater than two, the storage space requirements become excessive.

A new class of HTs called *probabilistic Hough transforms* (PHTs) attempted to address these problems by using *random sampling* of edge points, and *many-to-one mapping* of edge points from the image space to the parameter space. The simplest PHT is the Randomized HT (RHT) due to Xu, Oja, and Kultanen (1990). Below we outline the RHT algorithm for the general case of detecting all instances of a particular p-dimensional curve in a binary image.

1. Create the set E of all edge points in the binary edge detected image.

2. Select p points, $(e_1, \ldots, e_p)$, at random from E.

3. Solve for the parameters (θ) of the curve in the image space, defined by the selected points.

4. Increment the appropriate cell, $\mathcal{A}(\theta)$, in an accumulator array.

5. If the $\mathcal{A}(\theta)$ equals a predefined threshold t, then the curve parameterized by θ is detected. When this happens the points that lie on the curve are removed from E and the accumulator array is re-initialized.

6. Regardless of whether a curve is detected, check whether or not a stopping condition is satisfied. If it isn't, return to step 2.

The two main ingredients in this algorithm (and all PHTs) are a *sampling mechanism* (steps 2-3) and a *peak detection method* (steps 4-5).

The sampling mechanism defines a sampling distribution on the continuous parameter space, Θ. However, the sampling mechanism is discrete in nature, so we can

view the sampling mechanism as defining a discrete sample space $\Omega \subseteq \Theta$, and a sampling distribution (probability mass function), $g(\cdot)$, on Ω. In the RHT, the sampling mechanism is the selection of p points at random. Thus, if there are $K = \binom{N}{p}$ p-tuples of points and θ_i is the parameter for the i^{th} p-tuple, $\Omega = \{\theta_1, \ldots, \theta_K\}$. The sampling distribution can be written as:

$$g(\theta^*) = \frac{1}{K} \sum_{i=1}^{K} 1_{\{\theta^* = \theta_i\}} \qquad \theta^* \in \Omega.$$

If all the $\theta_i's$ are unique then $g(\cdot)$ is an uniform distribution on Ω, i.e.

$$g(\theta^*) = \frac{1}{K} \qquad \theta^* \in \Omega.$$

Curves that are present in the image should correspond to regions in Θ that have relatively large mass. The peak detection method of the RHT achieves this by grouping sampled parameter values into cells in the accumulator array and comparing with a threshold.

Applying the RHT is not straightforward if the curve is nonlinear with respect to its parameters, or in other words when selecting p points does not necessarily uniquely define a p-dimensional curve. Ellipses are an example of a curve that is nonlinear with respect to its parameters. A solution to the problem of detecting ellipses within the framework of PHTs has been explored by McLaughlin (1998) (based on work by Yuen, Illingworth, and Kittler (1989) in a non-PHT framework). In his method, three points are selected at a time. Estimates of the tangents to the (possible) ellipse at these points are then calculated and used to interpolate the center of the ellipse. Given the location of the ellipse center the remaining parameters are easily determined.

A greater problem affecting the RHT, is that its performance is poor when the image is noisy or complex. New PHTs have been proposed to increase performance of the RHT. The simplest modification to the RHT is the RHT with point distance criterion (RHT_D). In this HT, at each iteration, one point is sampled at random and then $p - 1$ points are sampled uniformly from all points that are within certain a

distance (greater than d_{min} and less than d_{max}) of the first point. Sensible choices of d_{min} and d_{max} lead to an increase in the probability of sampling points on a curve present in the image.

Comparisons of various probabilistic (and non-probabilistic) HTs can be found in Kälviäinen, Hirvonen, Xu, and Oja (1995) and Kälviäinen and Hirvonen (1997). These methods include the RHT_D mentioned above, the Dynamic RHT (DRHT), the Random Window RHT, the Window RHT, the Connective RHT, and the Dynamic Combinatorial HT. Most of these modifications to the RHT attempt to improve the sampling distribution in various ways so that it is more peaked around the parameters corresponding to the curves in the image, thus making the subsequent process of peak detection easier. In some cases the distinction between the sampling mechanism and peak detection method is blurred, e.g. in the DRHT where curves are detected by an iterative process involving two RHTs.

While all of these methods differ in the sampling distribution employed they all share two common traits: (i) they all detect curves sequentially, and (ii) curves are detected when the number of votes in a cell in the accumulator array exceeds a certain threshold, or in other words, when a curve has been sampled a certain number of times. We see these as areas where improvements can be made. When detecting curves sequentially, every time a curve is detected the accumulator array is reset, thus discarding other curves that have already been accumulated. When accumulating a curve in the accumulator array, typically no effort is made to assess the "quality" of the entire curve. A practical consequence of this is that a curve must be sampled many times to be detected.

In our approach we will use a criterion for judging the "quality" of a curve, and introduce a technique that allows detection of multiple curves simultaneously. The form of this criterion is suggested by considering the following question: "What distribution would I *ideally* wish to sample from?" If we can *define* an ideal sampling distribution and obtain a sample from $g(\cdot)$, then we can obtain a sample from the

ideal distribution via the technique of importance sampling.

In the following sections we introduce importance sampling and then show how it relates to the HT. We present a simple rule-of-thumb for determining how many samples from the sampling distribution are needed and suggest using clustering techniques to simultaneously identify curves. Results from applying these ideas to both simulated and real data are shown, and our method is compared to two standard HTs.

4.2 Importance Sampling

Importance sampling is a technique that can be useful in two situations: (i) when a sample from a particular *target* distribution is desired but simulation from that distribution is not straightforward, and (ii) when sampling from this distribution is possible but produces estimates of some quantity of interest with high variance. We will discuss importance sampling with regard to the first situation, and we will restrict our attention to the case where the target distribution is discrete and only known up to a scale.

Consider a discrete random variable, θ, with probability mass function $\{\pi(\theta) : \theta \in \Omega\}$ where Ω is the sample space of θ. We wish to obtain a sample $\theta_1, \ldots, \theta_T$ from $\pi(\cdot)$. We know the form of $\pi(\cdot)$ up to a scale, i.e.:

$$\pi(\theta) = f(\theta)/c, \quad \theta \in \Omega,$$

where the form of $f(\cdot)$ is known but the normalizing constant c is unknown. We now assume that there exists a probability mass function $g(\cdot)$ (the sampling distribution or importance sampling function) defined on Ω from which we can draw a sample, $\theta_1, \ldots, \theta_T$. Each observation in this sample is then weighted as follows:

$$w_i = \frac{f(\theta_i)/g(\theta_i)}{\sum_{j=1}^{T}\{f(\theta_j)/g(\theta_j)\}}.$$

These weights are called the *importance weights*. If a random (unweighted) sample from the target distribution is required (e.g. for plotting purposes), then *sample/importance resampling* (SIR) (Rubin 1987) can be employed. Here we simply resample the sampled parameters with probabilities proportional to their importance weights. If one is trying to estimate a quantity from the target distribution, the resampling should be performed without replacement (i.e once a parameter is resampled, it cannot be resampled again). This is to prevent observations with large weights from dominating the resulting estimate. However in our case our goal is to detect the location of the peaks of the target distribution. Thus oversampling the peaks of the distribution is not a concern, and if resampling is performed, it should be done with replacement.

4.3 *Placing the Hough Transform in an Importance Sampling Framework*

In this section we outline an algorithm for a PHT, which we call the ISHT, that can be viewed in an importance sampling framework. The algorithm to detect curves, parameterized by Θ, in a binary edge detected image proceeds as follows.

1. Create the set E of all edge points in the binary edge detected image.

2. Define a target distribution $f(\theta|E)$. This is a function that measures the "quality" or "goodness-of-fit" of the curve given by θ, to the edge points E. For convenience we shall write $f(\theta)$.

3. Obtain a random sample, or *batch*, of parameters, $\{\theta_1, \ldots, \theta_{T^*}\}$, of size T^* from a sampling distribution $g(\cdot)$.

4. For each sampled parameter, θ_i, in the batch, calculate its importance weight as:

$$w_i = \frac{f(\theta_i)}{\sum_{j=1}^{T} f(\theta_j)}.$$

Note: For simplicity we have assumed that $g(\cdot)$ is approximately uniform and thus does not appear in the above equation. See Sections 4.3.1 and 4.5 for discussion of the implications of this assumption.

5. Run a peak detection method on the weighted sample of parameters to identify the curve parameters. Remove the points corresponding to each curve identified from E.

6. Check whether or not certain stopping conditions have been satisfied. If not return to step 3.

The four main components of this algorithm are: a sampling distribution, a target distribution, a peak detection method, and suitable stopping conditions,. Each of these components are discussed below in greater detail.

4.3.1 Importance Sampling Distribution

Once a sampling mechanism is chosen it defines the importance sampling distribution, $g(\cdot)$. As with all PHTs, the sampling mechanism is vital to an efficient HT. If an image is moderately complex (e.g. several curves present) then the simplest sampling mechanism (sampling p points at random from the entire image) will rarely sample curve parameters. The easiest modification to this sampling mechanism is to use the point distance criterion (see Section 4.1.3). Surprisingly, even though this sampling mechanism is easy to implement, exact calculation of $g(\cdot)$ is difficult. This is also true of more complex sampling mechanisms. In these cases we make the simplifying assumption that the sampling distribution is approximately uniform and therefore

cancels from the calculation of the importance weights (i.e $w_i \propto f(\theta_i)$). We discuss this assumption and its relationship to our importance sampling framework in Section 4.5.

4.3.2 Target Distribution

The selection of a good target distribution is crucial to the success of the HT. A good target distribution will have a large concentration of its mass on the parameters corresponding to curves present in the image. The target distributions we consider are of the form:

$$f(\theta_i) = \beta(\theta_i) \times \alpha(\theta_i).$$

where $\beta(\theta_i)$ is a measure of the number of points associated with the curve given by θ_i, and $\alpha(\theta_i)$ is a measure of the goodness of fit of these points to the curve.

We considered several functions for $\beta(\cdot)$.

1. $\beta(\theta_i) = n_i \times 1_{\{n_i > t_n\}}$, where n_i is the number of points that lie on the curve given by θ_i, and t_n is a predefined threshold. If $\alpha(\cdot) \equiv 1$, then this can be thought of as the target distribution corresponding to the SHT.

2. $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j)$, where r_j is the minimum distance from the j^{th} point to the curve given by θ_i, and $W(r_j)$ is a weighting function.

This weighting function can be interpreted as a fuzzy membership function with *membership radius* R if $W(0) = 1$, $W(r)$ decreases monotonically from 0 to R, and $W(r) = 0$ for $r \geq R$ (Han, Koczy, and Poston 1994). Natural choices for $W(r)$ are:

1. Step-function:

$$W(r) = \begin{cases} 1, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

2. Gaussian:
$$W(r) = \begin{cases} e^{-r^2/\sigma^2}, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

3. Linear:
$$W(r) = \begin{cases} 1 - r/R, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

Note that for values of R between $\frac{1}{2}$ and $\frac{1}{\sqrt{2}}$, if one uses the Step weighting function then $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j) \approx n_i$, the number of points that lie on the curve .

The various goodness-of-fit criteria we considered were:

1. $\alpha(\theta_i) = (n_i/c_i)^G$, where n_i is defined as above and where c_i is the number of *pixels* that lie on the curve. G is a positive number. For $G = 1$, this is the proportion of pixels on the curve that are points.

2. $\alpha(\theta_i) = (p_i/c_i)^G$. Each point within distance membership radius R of the curve is projected onto the curve. The number of these projected points is p_i and c_i is defined as above.

3. $\alpha(\theta_i) = (p_i/c_i)^G \times 1_{\{(p_i/c_i)^G > t_\alpha\}}$, for some threshold t_α.

All of the above criteria are between 0 and 1. If $G > 1$ then it can be thought of as a penalty term, penalizing "bad" curves. However if $G < 1$ then the goodness-of-fit of each curve is boosted. The first goodness-of-fit criteria listed is very strict. It favors only "perfect curves" whereas the second criteria allows variation around the curve. The last criteria gives zero weight to curves that do not exceed a certain goodness-of-fit threshold.

Obviously there is a great variety of target distributions that can be used. The application will play a large role in determining which target distributions are suitable.

4.3.3 Peak Detection in the Parameter Space

Given a sample of parameters from the sampling distribution and their importance weights, a peak detection method must be used to determine the number of curves, and their parameters. This simultaneous detection of peaks in the parameter space can be thought of as a clustering problem. We wish to partition the sampled parameters into spatially compact groups, where each group represents possible parameters for a particular curve in the image. The center of each group will be our estimate of the parameters for that curve. Of course some sampled parameters do not correspond to any curve in the image. These parameters can be thought of as noise. The number of these sorts of parameters in the sample will depend on the efficiency of the sampling distribution. However, these parameters should all have low importance weights. They can be removed via thresholding the sample based on the importance weights. Once these noise parameters are removed, the remaining sample of parameters can be clustered using any one of numerous clustering methods. These range from the simple k-means clustering (Hartigan 1975) (for a range of values of k) to more complex hierarchical and model-based methods, e.g. M-Clust (Banfield and Raftery 1993; Fraley and Raftery 1998).

In this chapter, we shall use a method similar in spirit to the peak detection method used in the RHT. First, we threshold the sampled parameters based on their importance weights. The remaining parameters are assumed to be associated with a curve in the image. We then place each parameter in a cell of an accumulator array (as in the RHT) and run a connected components algorithm on the non-empty cells in the array. The parameters contained in the cells associated with a particular connected component are assumed to be associated with one curve in the image. We estimate the parameters of this curve by taking a weighted average of all the parameters associated with the component (where the weights used are the importance weights, w_i). Choosing the cell size appropriately is necessary for the curves to be detected

accurately. This requirement is common to all PHTs, though.

4.3.4 Stopping Conditions

One advantage of our approach over other PHTs is that the we do not have to sample the points that lie on a curve many times in order to detect the curve. This is because our algorithm incorporates a measure of the quality of each curve. As a result of this, simple probabilistic arguments can be used to develop stopping conditions for the HT.

Our notation will be as follows. Let b be the number of batches currently processed, $T = bT^*$ be the total number of parameters sampled, and b_0 be the current number of consecutive batches processed without detecting a single curve. Let the current number of points in the edge set, E, be N_b.

We propose two different stopping conditions to determine if a sufficient number of parameters have been sampled from $g(\cdot)$. One condition is to stop the HT if the total number of parameters sampled, T, exceeds some threshold, T_{max}. An alternative rule is to stop the HT if the number of consecutive batches processed without detecting a single curve, b_0, is greater than some threshold, b_0^{max}. Of course if the number of remaining edge points becomes zero (or undesirably low) then sampling should stop.

Stopping Condition 1: Total number of parameters sampled

The total number of parameters sampled, T, will be sufficient if the probability of sampling all curves present in the image at least once is high. We can write down this probability if we have some idea of the number and size of the curves in the image. For example consider an image containing M p-dimensional curves of n points each. Let $p_{\{n,N\}}$ be the probability of sampling p points all from a n point curve, given that the image contains a total of N points. The probability of having sampled all curves after T samples is:

$$Pr = 1 - \sum_{i=1}^{M} (-1)^{i-1} \binom{M}{i} \left(1 - i \cdot p_{\{n,N\}} \right)^{T}.$$

This probability can be plotted as a function of T and can then be used to gauge what would constitute a reasonable sample. Derivation of this probability can be found in Appendix B.

If the sampling mechanism is simply sampling p points at random then

$$p_{\{n,N\}} = \frac{\binom{n}{p}}{\binom{N}{p}}.$$

If one uses the point distance criterion then this probability becomes

$$p_{\{n,N\}} = \frac{n}{N} \times \frac{\binom{n'-1}{p-1}}{\binom{N'-1}{p-1}}.$$

Here n' is the expected number of points that would typically lie on a given curve that would satisfy the point distance criterion given that the first point selected lies on that given curve. Similarly N' is the total number of points that typically satisfy the point distance criterion given selection of the first point. This condition (provided the guesses for M, n, and N are good) will result in a conservative stopping condition. This is because the above calculation does not take into account the fact that, as curves are detected and removed from the image, the probability of detecting the remaining curves increases.

Stopping Condition 2: Number of batches processed since last detection

Consider the number of consecutive batches processed without detecting a single curve, b_0. If this number is not zero, then we should cease sampling if the probability of not having sampled a curve over these $b_0 T^*$ samples, given that one remains, is low. Call this probability δ. This can be calculated as follows:

$$\delta = (1 - p_{\{n,N_b\}})^{T}.$$

Sampling should stop if δ is low enough, (less than a predefined threshold δ_0 say), or alternatively if:

$$b_0^{max} = \frac{1}{T^*} \frac{\log \delta_0}{\log(1 - p_{\{n,N_b\}})} \leq b_0.$$

This stopping condition is preferable to the first one, since it uses more information about the current state of the HT (e.g. how many curves have been detected, which edge points remain) and should result in a smaller sample size being needed. Discussion of the selection of the batch size is given can be found in Section 4.5.

4.4 Examples

In this section we show some results obtained by applying our methods to both real and simulated data. We compared our method (ISHT) to the randomized Hough transform with point distance criterion (RHT_D) and the standard Hough transform (SHT). Admittedly, these are the two simplest HTs currently available, however the goal of this section is not to demonstrate beyond any doubt the superiority of our approach over any other, but rather to stress the need for measuring the "quality" of sampled curves, the feasibility of detected multiple curves, and the usefulness of probabilistic stopping rules.

4.4.1 Simulated Data

We simulated a 256 × 256 binary image, containing 10 lines (see Figure 4.2). In this image we randomly changed the color of each pixel with probability 0.01 The total number of points in the image is 2809, while the total number of points associated with lines is 2192. However, the probability of sampling a pair of points at random both from the same line is only about 0.08.

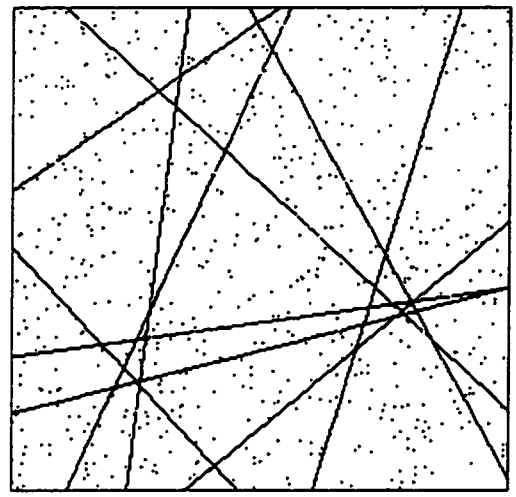


Figure 4.2: Simulated 256 × 256 binary image containing 10 lines and 1% speckle noise.

ISHT

We chose the sampling distribution corresponding to the sampling mechanism of the RHT with point distance criterion with $d_{min} = 1$, and $d_{max} = 50$, i.e. the two pixels selected must be within 50 pixels of each other. The components of our target distribution were given by:

- $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j)$, where $W(\cdot)$ is the Gaussian weighting function with $R = 2$, and $\sigma = 2$.

- $\alpha(\theta_i) = (p_i/c_i) \times 1_{\{(p_i/c_i)>0.5\}}$

This target distribution allows points within 2 pixels of the line to be associated with the curve, but penalizes curves that do not have at least half as many points near the line as pixels on the line. We used a peak detection method with a importance weight threshold of zero. The cell size of the accumulator array (in (ρ, θ) notation)

was $1.8 \times \frac{2\pi}{100}$. The batch size was 50, and the HT was stopped if two consecutive batches failed to detect any curves.

RHT_D

We set the threshold of the RHT_D to be 10 points, and used the same point distance criterion as the ISHT above. We ceased sampling when the number of points remaining in the image was less than 30% of the original total. The cell size of the accumulator array was $1.8 \times \frac{2\pi}{100}$. When removing detected curves from the image we removed all points within a 3 pixel radius of the curves.

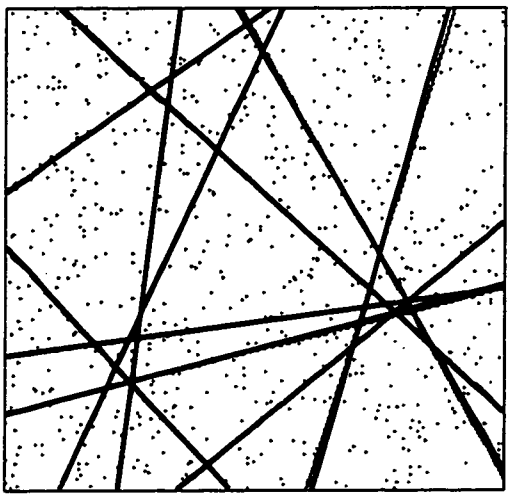
SHT

The SHT threshold was set 130 points. The cell size of the accumulator array was $3.6 \times \frac{\pi}{100}$. After thresholding the accumulator array (all cells below the threshold were set to zero), we ran a connected components algorithm on the non-empty cells. The cell with the most votes inside each connected component was identified as corresponding to a curve present in the image. The parameters of the curve were given by the parameters of the center of the cell. This curve identification procedure is similar to our ISHT clustering algorithm.

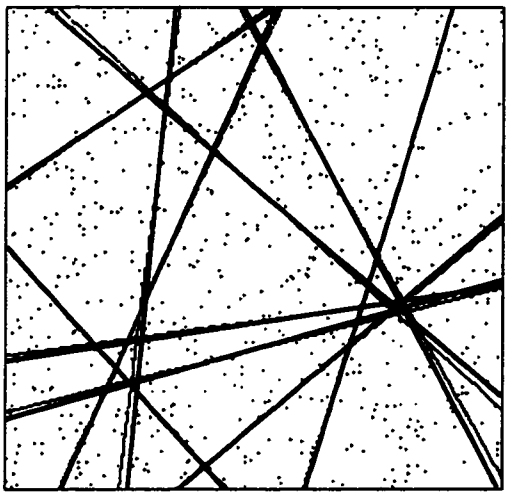
Results

The lines detected by each method are shown in Figure 4.3. Each method detected all 10 lines in the image with no false positives, however the accuracy and the speed of each method in this example varied considerably. Table 4.1 shows the run times of each method, and the mean squared errors for ρ and θ. The parameter estimates are for each method are plotted versus the true values in the Figure 4.4.

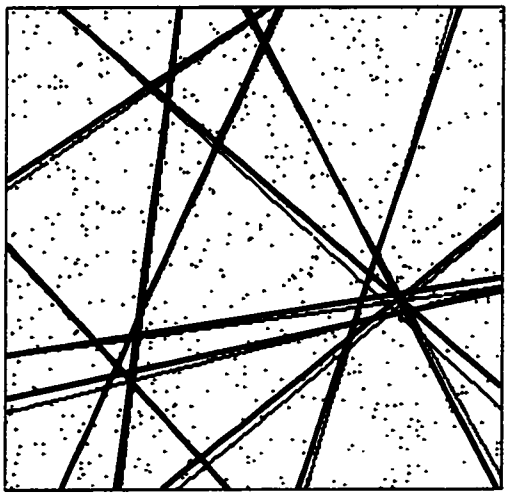
The SHT was the fastest method by far, approximately 3 times faster than the ISHT, which was an order of magnitude faster than the RHT_D. This result is not



(a) ISHT.

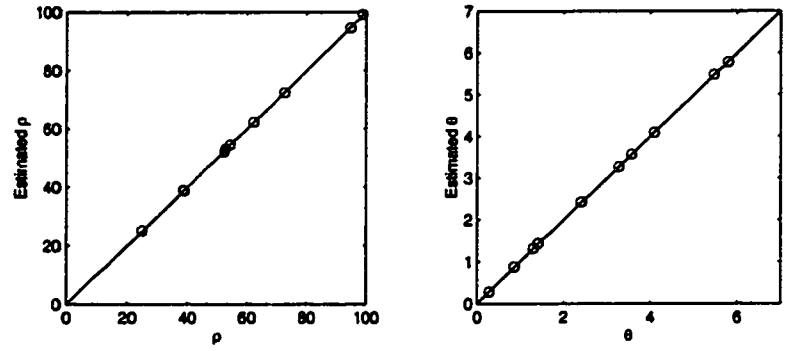


(b) RHT_D.

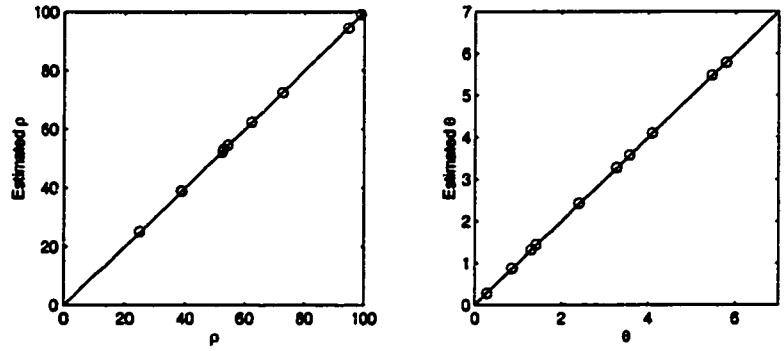


(c) SHT.

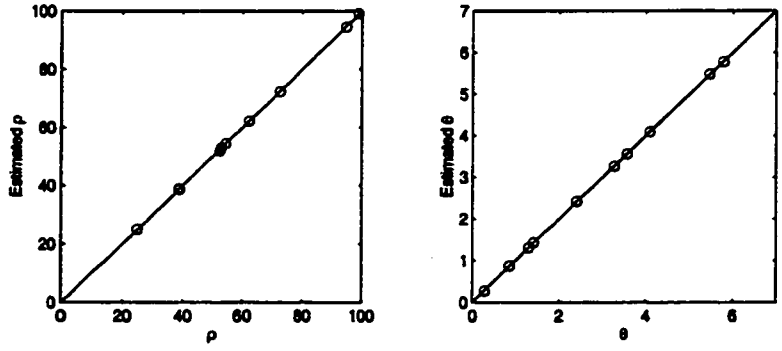
Figure 4.3: Simulated data : Lines detected.



(a) ISHT.



(b) RHT_D.



(c) SHT.

Figure 4.4: Simulated Data: Parameter estimates of ρ and θ for each HT, plotted versus their true values. A 45° line is added to each plot for ease of comparison.

too surprising given that the dimensionality of the parameter space is low (high dimensions adversely affect the SHT more than the RHT), and the complexity of the image was high (the complexity of the image affects the RHT more than the SHT).

Table 4.1: Simulated Data: Run times, and mean squared errors for each HT.

	ISHT	RHT_D	SHT
Run Time (seconds)	4.73	44.64	1.770
MSE of ρ	0.061	0.362	5.293
MSE of θ ($\times 10^5$)	9.02	40.91	45.65

In terms of accuracy, the ISHT was clearly the best. The MSEs for θ was 4 times lower than either one of the other methods, while the MSE for ρ was nearly an order of magnitude better for the ISHT than for the RHT_D, and two orders of magnitude better than the SHT.

4.4.2 Stopping Conditions

For this example we used the stopping condition based on the number of consecutive empty batches. Specifically, we stopped the ISHT when we had two empty batches, each of size 50 parameters. The final image, after the detected lines had been removed, contained 692 points. It would be interesting to calculate the probability of not detecting a line, given that one existed, in these remaining 100 (50 $\times$ 2) samples. Suppose, for simplicity, that our sampling mechanism, had been that of the RHT, and the undetected line contained 100 points. In this case, the probability of sampling two points on this line is:

$$p_{\{n,N_b\}} = \binom{n}{2} / \binom{N}{2}$$

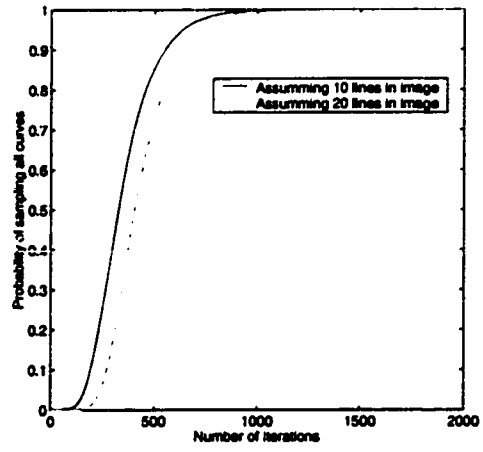


Figure 4.5: Simulated Data: Probability of sampling all lines in the image by a given iteration, assuming there are 10 or 20 lines in the image.

$$\begin{aligned} &\approx (n/N)^2 \\ &= (100/692)^2 \\ &= 0.0209. \end{aligned}$$

Therefore the probability of not detecting the curve is:

$$\begin{aligned} Pr &= (1 - p_{\{n,N_b\}})^{100} \\ &= (1 - 0.0209)^{100} \\ &= 0.12. \end{aligned}$$

This probability is low, although not that low. It suggests that maybe we should run our HT longer. The alternative stopping condition, based on the total number of parameters sampled, is more conservative. Figure 4.5 shows the probability of sampling all curves in the image for a given total sample size, assuming there are either 10 or 20 lines in the image (where each line is of size 256 points). The probability for

both curves is high when the number of parameters sampled is approximately 750. In this example we sampled 250 parameters in total, and we can see that the probability is only approximately 0.5. Since we had detected all the curves in the image after 250 samples, it appears that this stopping condition can be too conservative; however, it is a good rule of thumb for establishing an upper bound for the number of parameters sampled.

4.4.3 Real Data

Figure 4.6 shows a 265×272 image of blood cells. The edge detected image, obtained via a Sobel transform, is shown in Figure 4.7. The task of identifying each blood cell is not straightforward. There are numerous blood cells in the image. All are roughly circular, although not perfectly so, and several are occluded, either by other blood cells or by the boundary of the image.

ISHT

We chose a batch size of 200 samples. The sampling mechanism and target distribution are the same as in the previous example. Again we clustered the sampled parameters after each batch using our peak detection method. The cell size used was (in (r, a, b) notation) $(0.8 \times 11 \times 11)$ and the sample was thresholded based on an importance weight of 45. The ISHT was stopped if 4 consecutive batches detected no curves.

RHT_D

We set the threshold of the RHT_D to be 10 points, and used the same point distance criterion as the ISHT above. The size of a cell in the accumulator array was $(0.8 \times 5 \times 5)$

As with the previous example, we ceased sampling when the number of points remaining in the image was less than 30% of the original total, and we used a 3 pixel

Figure 4.6: Light microscope image of red blood cells.

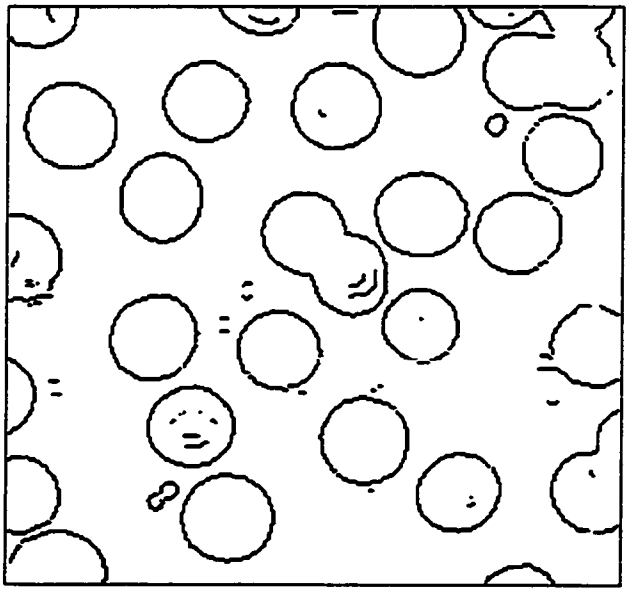


Figure 4.7: Blood cells - Edge detected image.

radius when determining which points should be removed after a curve had been detected.

SHT

In order to alleviate the severe storage requirements of the SHT when detecting circles we employed the Gerig-Klein modification to the HT (Gerig and Klein 1986). This modified HT allows only one circle to be detected at each given center. The reduces the accumulator array from a three-dimensional array to two two-dimensional arrays. The cost is the inability to detect concentric circles. Of course in this example that is a reasonable assumption.

Results

The final classifications for each method are shown in Figure 4.8. We considered 26 of the blood cells in the image to detectable. The number of cells correctly detected, along with the run times, the number of false positives, and the number of duplicates are shown in Table 4.2.

Table 4.2: Blood Cell Analysis.

	ISHT	RHT_D	SHT
Run time	44.90	263.6	138.9
Cells detected	21	23	20
Cells undetected	5	3	6
False positives	0	13	0
Duplicates	0	2	5

The RHT_D detected the most blood cells, but also detected 13 false positives. The SHT detected 20 blood cells, but also detected 5 duplicates. Our method has

detected the 21 of the 26 blood cells, detected no false positives or duplicates and had by far the shortest run time. We can see by visually comparing the images in Figure 4.8 that the circles detected by the ISHT fit the blood cells better. The ISHT did not detect several of the cells that are not fully within the image, nor did detect the two partially occluded cells that are fully contained within the image. However, with suitable modifications of the target distribution these cells could conceivably be detected.

4.4.4 Applying the ISHT to the COBRA datasets

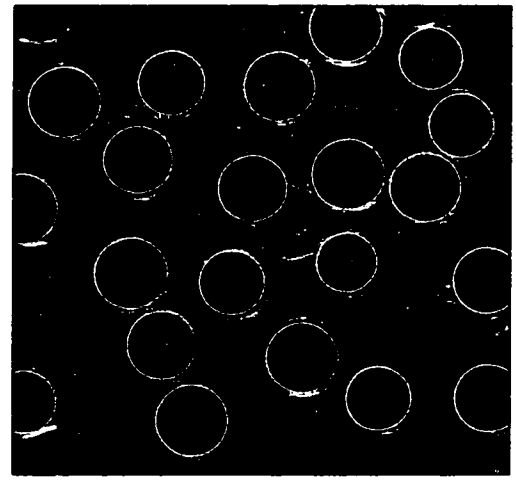
We applied the ISHT to the COBRA datasets of chapter 3. Given the great computational demands of the MCMC approach we used, it would be of great interest to see if our Hough transform approach would be successful. We discretized the COBRA point processes into 100×100 binary images. The ISHT for each COBRA dataset was initialized in the same way. We used the sampling mechanism corresponding to the standard RHT. The target distribution was given by:

- $\beta(\theta_i) = \sum_{j=1}^{N} W(2)$, where W is the step function.
- $\alpha(\theta_i) = 1_{\{(\sum_{j=1}^{N} W(2))>10\}}$

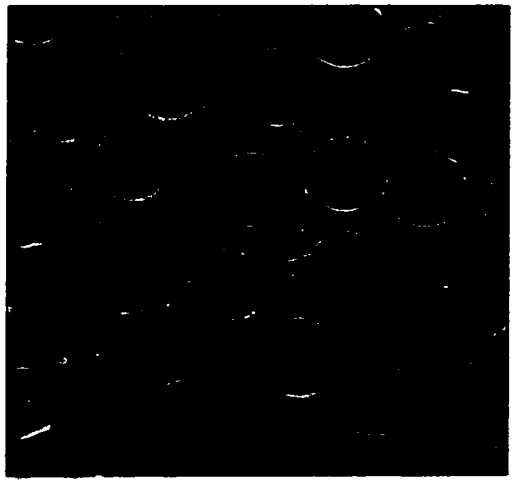
In other words we counted the number of points within 2 pixels of the curve and used a threshold of 10 points. The cell size of the clustering algorithm was $1 \times \frac{2\pi}{100}$.

The final classifications for each dataset are shown in Figure 4.9. In general, the ISHT was unsuccessful at detecting the rows of mines. Two rows of mines are detecting in the Surf Zone dataset however most of the mines are missed. Although it appears that the ISHT is unsuited to this type of dataset, it may that a different target distribution, one that takes into account the distances between consecutive points, may be more successful. This is one possible area for future research.

(a) ISHT - All circles corresponding to sampled parameters with positive importance weight.



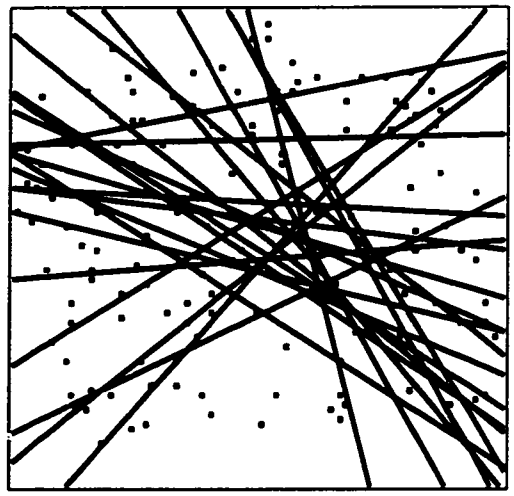
(b) ISHT - Circles detected.



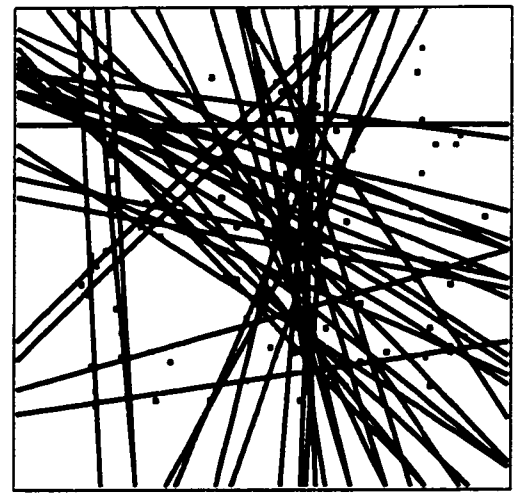
(c) RHT_D - Circles detected.

(d) SHT - Circles detected.

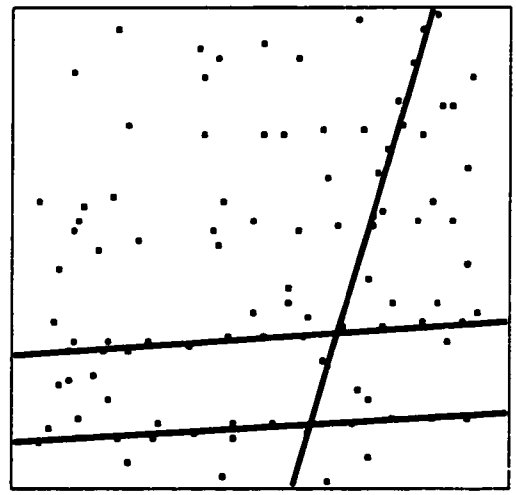
Figure 4.8: Blood Cells: Circles Detected.



(a) Eglin.



(b) Lejeune.



(c) Surf Zone.

Figure 4.9: COBRA data. Line detected.

4.5 Discussion

We have found importance sampling to be a natural framework with which to design, view, and understand probabilistic Hough transforms. We feel that the notion of a target distribution, which defines an importance weight associated with every sampled parameter, is fundamental to the successful implementation of a PHT. We investigated the feasibility of detecting multiple curves from a given batch of sampled parameters. We found that simple clustering methods provided good identification of curve parameters, provided that the sampled parameters not associated with curves present in the image were filtered out. The removal of these "noise" parameters was achieved via a simple thresholding of the importance weights. Probabilistic arguments can be used to determine reasonable stopping conditions for PHTs.

The question may be asked "Is our approach really importance sampling?" Typically when performing importance sampling, the goal is to obtain a sample from the target distribution, either to estimate the target distribution (perhaps because it is only known up to a constant), or to estimate the expectation of some function with respect to the target distribution. However, we are only interested in the location of the peaks of the target distribution. In light of this, the approximation we make in assuming that $g(\cdot)$ is uniformly distributed on Ω has little impact on the detection of curves in the image. In contrast, making such an approximation when calculating the expectation of a function could lead to undesirable properties in the resulting estimate, such as bias. In spite of this distinction between the typical importance sampling application and the problem of curve detection, we found that the importance sampling framework provided a natural way to incorporate a goodness of fit measure, through weights on the sampled parameters.

As with most data processing algorithms, their successful implementation often depends on correctly setting several tuning parameters. We feel that, depending on the given image, the tuning parameters of the ISHT are reasonably intuitive to set.

One parameter, the batch size (if one is employing the second stopping condition) can affect the efficiency and accuracy of the HT. If the batch size is too large, the increased efficiency of sampling from a smaller pool of points is lost. Conversely, if the batch size is too small curves may be detected after only being sampled once, thus increasing the bias in the parameter estimates. One way to possibly avoid this problem is to use a small batch size, and employ an algorithm similar to the DRHT. That is, for each detected curve, select all the points near the curve and perform a refined HT on these points. This technique can be thought of as adaptively sampling the parameter space, and it is of course not the only conceivable approach to take.

The importance sampling framework is broad enough to incorporate many of the beneficial features of other existing PHTs. For example, we used a very simple sampling distribution, whereas one could use a more sophisticated sampling distribution, e.g. one based on the Connective RHT. In addition, the identification of curves could be improved by incorporating an adaptive sampling mechanism, such as in the Dynamic RHT. The main computational burden of our approach over other PHTs is the evaluation of the target distribution. This should be an $O(N)$ operation, where N is the total number of points in the image. We feel that the benefit of evaluating the quality of a curve far outweighs the cost of its computation. In summary, we feel that importance sampling provides a framework in which different PHTs can be understood, and which will be useful in guiding the design of better curve detection methods.

Chapter 5

DISCUSSION

In this chapter we summarize the dissertation and outline possible directions for future investigation.

5.1 Summary of Dissertation

In Chapter 2 we addressed the problem of classifying mixtures of spatial point processes. We proposed using partial Bayes factors to solve this problem, and demonstrated their applicability to this problem via a simulation study. We discussed the merits and drawbacks of this approach with respect to the supervised classifier advocated by Raghavan, Goel, and Ghosh (1998).

In Chapter 3 we considered the problem of detecting individual mines from a processed aerial image of a possible minefield. We proposed a stochastic model for the location of the mines in the minefield and fitted this model via Markov chain Monte Carlo.

Chapter 4 introduced the Hough transform. We showed how viewing it in an importance sampling framework could lead to improved understanding of its properties and suggested areas for improvement. These improvements included: defining a target distribution to assess the quality of sampled curves; using probabilistic stopping rules to determine the required number of curve parameters to be sampled; and a simple clustering algorithm to allow the detection of several curves from a sample of curve parameters.

5.2 Future Research

To conclude this dissertation, ideas for possible future research are outlined.

5.2.1 Classification of Spatial Point Patterns

The approach of Raghavan, Goel, and Ghosh (1998) naturally allows the best summary statistic(s) for each class of spatial point process to be used to classify it. It would be nice if there was a similar mechanism to choose the appropriate summary statistic when using partial Bayes Factors.

Currently, our use of partial Bayes factors has been limited to the problem of classifying spatial point processes. It would be interesting to extend this methodology to other applications.

5.2.2 Mine Detection

One drawback of the MCMC algorithm we propose in Chapter 3 is its slow mixing. A new MCMC algorithm with better mixing properties might have the following form:

- Generate a number of parallel lines in the study region.
- Identify evenly spaced points that are close to each line.

The difficulty with this algorithm is determining how to assign points to be mines given the line configuration.

5.2.3 The Hough transform

In our implementation of the Hough transform in Chapter 4, we used very simple sampling distributions. A more complex sampling distribution (e.g. one based on the Connective Randomized Hough transform), may give noticeable improvements in both speed and performance of the Hough transform.

A second area for improvement would be in the actual parameter estimation of the detected curves. Any bias in the estimated curve parameters could be reduced by incorporating an adaptive sampling mechanism, such as the one used in the Dynamic Randomized Hough transform.

BIBLIOGRAPHY

Allard, D. and C. Fraley (1997). Non-parametric maximum likelihood estimation of features in spatial point processes using Voronoï tessellation. *Journal of the American Statistical Association 92*, 1485–1493.

Banerji, A. and J. Goutsias (1995). Detection of minelike targets using grayscale morphological image reconstruction. In A. C. Dubey, I. Cindrich, J.M.Ralston, and K.Rigano (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets*, Volume 2496 of *Proceedings of SPIE*, pp. 836–849. 17-21 April, Orlando, Florida.

Banfield, J. D. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–822.

Bartlett, M. S. (1964). The spectral analysis of two-dimensional point processes. *Biometrika 51*, 299–311.

Braga-Neto, U. M. and J. Goutsias (1998). On detecting mines and minelike objects in highly cluttered multispectral aerial images by means of mathematical morphology. In A. C. Dubey, J. F. Harvey, and J. T. Broach (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets III*, Volume 3392 of *Proceedings of SPIE*, pp. 987–999. 13-17 April, Orlando, Florida.

Byers, S. D. and A. E. Raftery (1997). Bayesian estimation and segmentation of spatial point processes using Voronoï tilings. Technical Report 326, Department of Statistics, University of Washington.

Byers, S. D. and A. E. Raftery (1998). Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical*

Association 93, 557–584.

Cressie, N. and A. B. Lawson (1997). Models and inference for clustering of locations of mines and mine-like objects. In A. C. Dubey and R. L. Barnard (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets II*, Volume 3079 of *Proceedings of SPIE*, pp. 519–530. 21-24 April, Orlando, Florida.

Cressie, N. and A. B. Lawson (1998). Bayesian hierarchical analysis of minefield data. In A. C. Dubey, J. F. Harvey, and J. T. Broach (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets III*, Volume 3392 of *Proceedings of SPIE*, pp. 930–940. 13-17 April, Orlando, Florida.

Cressie, N. A. C. (1993). *Statistics for spatial data.* New York: John Wiley & Sons. Revised edition.

Dasgupta, A. and A. E. Raftery (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association 93*, 294–302.

DiCiccio, T. J., R. E. Kass, A. E. Raftery, and L. Wasserman (1997). Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association 92*(439), 903–915.

Diggle, P. J. (1983). *Statistical analysis of spatial point patterns.* London: Academic Press.

Diggle, P. J., T. Fiksel, P. Grabarnik, Y. Ogata, D. Stoyan, and M. Tanemura (1994). On parameter estimation for pairwise interaction point processes. *International Statistical Review 62*(1), 99–117.

Duda, R. O. and P. E. Hart (1972). Use of the Hough transform to detect lines and curves in pictures. *Communications of the Association for Computing Machin-*

ery : Graphics Image Processing 15(1), 11–15.

Fraley, C. and A. E. Raftery (1998). How many clusters? Which clustering method? - Answers via model-based cluster analysis. Technical Report 329, Department of Statistics, University of Washington.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*, 457–511. With discussion.

Gerig, G. and F. Klein (1986). Fast contour identification through efficient Hough transform and simplified interpretation strategy. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pp. 498–500. Paris.

Geyer, C. J. and E. A. Thompson (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association 90*(431), 909–920.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (Eds.) (1996). *Markov Chain Monte Carlo in practice*. London: Chapman & Hall.

Han, J. H., L. T. Koczy, and T. Poston (1994). Fuzzy Hough transform. *Pattern Recognition Letters 15*(7), 649–658.

Hartigan, J. A. (1975). *Clustering algorithms*. New York: John Wiley & Sons. Wiley Series in Probability and Mathematical Statistics.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109.

Holmes, Q. A., C. R. Schwartz, J. H. Seldin, J. A. Wright, and L. J. Witter (1995). Adaptive multispectral CFAR detection of land mines. In A. C. Dubey, I. Cindrich, J. M. Ralston, and K. Rigano (Eds.), *Detection and Technologies for Mines and Minelike Targets*, Volume 2496 of *Proceedings of SPIE*, pp. 421–432. 17-21 April, Orlando, Florida.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

Illingworth, J. and J. Kittler (1988). A survey of the Hough transform. *Computer Vision Graphics and Image Processing 44*(1), 87–116.

Isham, V. (1984). Multitype Markov point processes: some approximations. *Proceedings of the Royal Society of London, Series A 391*(1800), 39–53.

Jeffreys, H. (1961). *Theory of probability* (Third ed.). Oxford: Clarendon Press.

Kälviäinen, H. and P. Hirvonen (1997). An extension to the randomized Hough transform exploiting connectivity. *Pattern Recognition Letters 18*(1), 77–85.

Kälviäinen, H., P. Hirvonen, L. Xu, and E. Oja (1995). Probabilistic and non-probabilistic Hough transforms - overview and comparisons. *Image and Vision Computing 13*(4), 239–252.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

Kelly, F. P. and B. D. Ripley (1976). A note on Strauss's model for clustering. *Biometrika 63*(2), 357–360.

Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

Lake, D. E. and D. M. Keenan (1995). Identifying minefields in clutter via collinearity and regularity detection. In A. C. Dubey, I. Cindrich, J. M. Ralston, and K. Rigano (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets*, Volume 2496 of *Proceedings of SPIE*, pp. 519–530. 17-21 April, Orlando, Florida.

Lake, D. E., B. Sadler, and S. Casey (1997). Detecting regularity in minefields using collinearity and a modified Euclidean algorithm. In A. C. Dubey and R. Barnard

(Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets II*, Volume 3079 of *Proceedings of SPIE*, pp. 508–518. 21-24 April, Orlando, Florida.

Leavers, V. (1993). Which Hough transform? A survey of Hough transform methods. *CVGIP - Image Understanding 58*(2), 250–264.

Marinari, E. and G. Parisi (1992). Simulated tempering - a new Monte-Carlo scheme. *Europhysics Letters 19*(6), 451–458.

McLaughlin, R. A. (1998). Randomized Hough Transform: Improved ellipse detection with comparison. *Pattern Recognition Letters 19*(3-4), 299–305.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *Journal of Chemistry and Physics 21*, 1087–1091.

Muise, R. R. and C. M. Smith (1992). Nonparametric minefield detection and localization. Technical Report CSS-TM-591-91, Naval Surface Warfare Center, Coastal Systems Station.

Muise, R. R. and C. M. Smith (1995). A linear density algorithm for patterned minefield detection. In A. C. Dubey, I. Cindrich, J. M. Ralston, and K. Rigano (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets*, Volume 2496 of *Proceedings of SPIE*, pp. 586–593. 17-21 April, Orlando, Florida.

Okabe, A., B. Boots, and K. Sugihara (1992). *Spatial tesselations: Concepts and applications of Voronoï diagrams*. Chichester: John Wiley & Sons.

Raftery, A. E. (1996). Hypothesis testing and model selection. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in practice*, London, pp. 163–188. Chapman & Hall.

Raftery, A. E. and S. M. Lewis (1992). How many iterations in the Gibbs sampler? In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (Eds.), *Bayesian Statistics 4*, Oxford, pp. 765–776. Oxford University Press.

Raftery, A. E. and S. M. Lewis (1996). Implementing MCMC. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in practice*, pp. 115–130. Chapman & Hall.

Raghavan, N., P. K. Goel, and S. Ghosh (1998). Computer experiments for the classification of spatial point processes. Technical Report 600, Department of Statistics, Ohio State University.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability 13*(2), 255–266.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society, Series B 39*(2), 172–212. With discussion.

Rubin, D. B. (1987). Comment on 'The calculation of posterior distributions by data augmentation' by Tanner and Wong. *Journal of the American Statistical Association 82*(398), 543–546.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman & Hall.

Smith, A. F. M. and G. O. Roberts (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Series B 55*(1), 3–23. With discussion.

Stanford, D. and A. E. Raftery (1997). Principal curve clustering with noise. Technical Report 317, Department of Statistics, University of Washington.

Stoyan, D. and W. S. Kendall (1995). *Stochastic geometry and its applications* (2nd ed.). Chichester: John Wiley & Sons Ltd.

Strauss, D. J. (1975). A model for clustering. *Biometrika 62*(2), 467–475.

Witherspoon, N. H., J. H. Holloway Jr., K. S. Davis, R. W. Miller, and A. C. Dubey (1995). The Coastal Battlefield Reconnaissance and Analysis (COBRA) program for minefield detection. In A. C. Dubey, I. Cindrich, J. M. Ralston, and K. Rigano (Eds.), *Detection and Remediation Technologies for Mines and Minelike Targets*, Volume 2496 of *Proceedings of SPIE*, pp. 500–508. 17-21 April, Orlando, Florida.

Xu, L., E. Oja, and P. Kultanen (1990). A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters 11*(5), 331–338.

Yuen, H. K., J. Illingworth, and J. Kittler (1989). Detecting partially occluded ellipses using the Hough transform. *Image and Vision Computing 8*(1), 71–77.

Yuen, S. Y. and C. H. Ma (1997). An investigation of the nature of the parameterization for the Hough transform. *Pattern Recognition 30*(6), 1009–1040.

Appendix A

JUMP PROPOSAL MOVE DETAILS

The Jump move allows the Markov chain to move large distances in the parameter space in a single iteration. The steps involved in the Jump move are:

1. Propose new univariate parameters.

2. Propose a new number of rows and mines per row (K' and $n'_0, \ldots, n'_{K'}$).

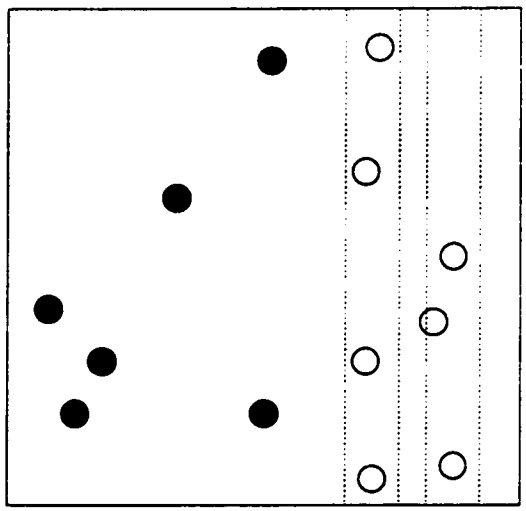
3. Propose new start mines.

4. Grow new rows.

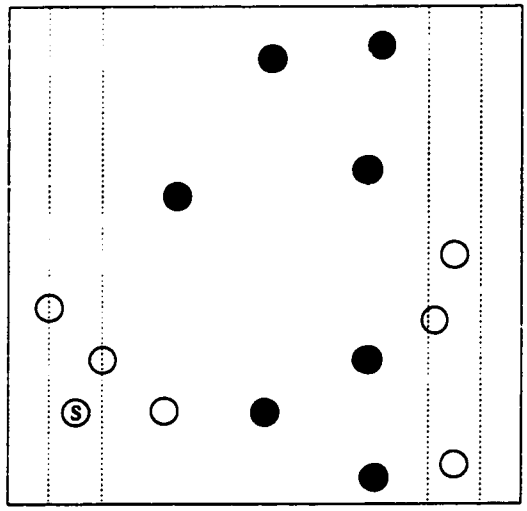
In our simulations for steps 1 and 2 above, we only proposed new values for the minefield orientation parameter θ. The proposed value θ' was drawn from the prior on θ. When proposing new start mines, it is important to ensure that the proposed mines are valid; i.e. that the bands of width ρ centered on the start mines in direction θ do not overlap. A secondary concern is that the calculation of the probability of the proposed mines not be too computationally expensive.

Our approach is to first project all points onto a line perpendicular to θ', and sort each point by its distance *along* this line. Next, associate a *selection probability* p_i to each point. We based p_i on the distance to the projection line. If this distance is small (in the lower quartile of all such distances) we set $p_i = 0.9$, else we set $p_i = 0.1$. This is to encourage starting mines to be near the "front" of the data.

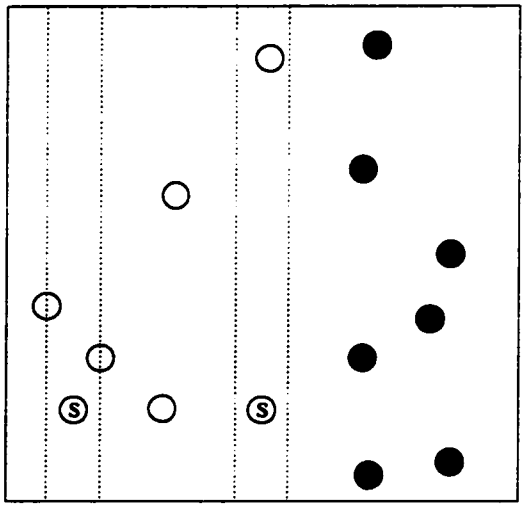
We now select K' mines sequentially from left to right along the projection line. To do this, we first place $K'-1$ non-overlapping bands as far to the right as possible while



(a) Before selection of the first start mine.



(b) First start mine selected.



(c) Second start mine selected.

Figure A.1: Selection of the first two start mines : $K' = 3$. Candidate points are represented by solid circles. S denotes start mines.

being centered on a point in the direction θ' as in Figure A.1(a). We then select one of the points to the left of the bands (the solid circles in the Figure) with probability proportional to p_i. This point is the first start mine. We now move the leftmost band and place it on this point (Figure A.1(b)). This creates a new set of candidate points from which the next start mine can be selected (Figure A.1(c)). This procedure is repeated until K' start mines are selected. Once the start mines have been selected, the remaining mines in each row are added sequentially with probability proportional to the likelihood of edge created, as in the Add move.

Appendix B

CALCULATION OF THE PROBABILITY OF DETECTING ALL CURVES

Given M events, $A_0, \ldots, A_{M-1}$, the probability of the union of all of these events can be expressed in the following way:

$$\begin{aligned} P(\cup_{i=1}^{M} A_i) &= P(A_1) + \ldots + P(A_M) \\ &\quad +(-1)^1\{\sum_{i<j} P(A_i \cap A_j)\} \\ &\quad +(-1)^2\{\sum_{i<j<k} P(A_i \cap A_j \cap A_k)\} \\ &\quad \vdots \\ &\quad +(-1)^{M-1}\{P(\cap_{i=1}^{M} A_i)).\} \end{aligned}$$

Now consider an image containing M p-dimensional curves of n points each. Assume a sampling distribution, $g(\cdot)$, exists that enables one to select p points from the N total points in the image.

Now consider a sampling scheme where, at each draw, the p points are sampled from the N total points using the sampling distribution. If, at a particular draw, all p points are sampled from the same curve, we shall consider that curve to have been sampled. Define $p_{\{n,N\}}$ to be the probability of sampling a curve at a particular draw (we assume for simplicity this is the same for all curves).

Let A_i^t be the event that the i^{th} curve has *not* been sampled after t draws of

p points from the sampling distribution. Since each curve has n points and each sampled parameter is drawn independently from $g(\cdot)$, we can write:

$$P(A_i^t) = (1 - p_{\{n,N\}})^t, \text{ for } i = 1, \ldots, M.$$

Similarly $P(A_i^t \cap A_j^t)$ can be calculated as follows:

$$\begin{aligned} P(A_i^t \cap A_j^t) &= (P(A_i^1 \cap A_j^1))^t \\ &= (1 - P((A_i^1 \cap A_j^1)^c))^t \\ &= (1 - P((A_i^1)^c \cup (A_j^1)^c))^t \\ &= (1 - (P((A_i^1)^c) + P((A_j^1)^c) - P((A_i^1)^c \cap (A_j^1)^c)))^t \\ &= (1 - p_{\{n,N\}} - p_{\{n,N\}} + 0)^t \\ &= (1 - 2 \cdot p_{\{n,N\}})^t, \text{ for all } i = 1, \ldots, M, j \neq i. \end{aligned}$$

The term $P((A_i^1)^c \cap (A_j^1)^c)$ is equal to zero since $(A_i^1)^c$ and $(A_j^1)^c)$ are mutually exclusive (one cannot sample two curves at the same time). Similarly, the probability of the intersection of all events can be written as:

$$P(\cap_{i=1}^M A_i^t) = (1 - M \cdot p_{\{n,N\}})^t.$$

Substituting these expressions into the first equation we find that the probabilty of *not* sampling all curves at least once after t samples have been drawn from $g(\cdot)$ is:

$$P(\cup_{i=1}^M A_i^t) = \sum_{i=1}^{M} (-1)^{i-1} \binom{M}{i} \left(1 - i \cdot p_{\{n,N\}} \right)^t.$$

The probability of having sampled all curves after t samples is simply this quantity subtracted from 1.

VITA

Daniel Walsh was born on June 28th, 1972, in New Plymouth, New Zealand. He attended Francis Douglas Memorial High School. In 1994 he received a Bachelor of Science Honors Degree with first class honors from the University of Auckland, with a major in Statistics. He commenced graduate studies in the Department of Statistics at the University of Washington in the fall of 1995. He received a M.S. in Statistics in 1997, and a Ph.D. in 2000.