Algorithms for Estimating Visual Motion Parameters from Ganglion Cell Responses

by

STAVROS VALAVANIS

B.S. Electrical Science and Engineering, 2007

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2008

Author .....
Department of Electrical Engineering and Computer Science
January 17, 2008

Certified by..
John L. Wyatt
Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Algorithms for Estimating Visual Motion Parameters from Ganglion Cell Responses

by

STAVROS VALAVANIS

B.S. Electrical Science and Engineering, 2007

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
January, 2008

Abstract

This thesis is intended to present a specific sub-problem of a larger one we call the "Inverse Problem". We wish to estimate the velocity (speed and direction) of an edge of light which is moving on the photoreceptor layer of a rabbit retinal patch. We make these estimates based solely on the electrical responses measured from the retinal ganglion cells (RGCs). To this end, we develop five novel algorithms. The first four of these are least squares based and the fifth one employs a maximum likelihood approach. We present a sensitivity analysis on the four least squares algorithms. We also develop a novel method for reweighing these least squares algorithms so as to minimize a weighted sum of the variances of our estimates. The fifth algorithm is significantly more complex than the first four as it involves creating cell models through "training"; moreover, it uses the entirety of each cell's response whereas the least squares algorithms use only first order statistics of each cell's response. We present and compare the results of the top performing least squares algorithm with the fifth algorithm on data recorded from a retinal patch. Through simulations, we explore the effects of using a small number of closely "clustered" cells on the performance of these two algorithms.

Thesis Supervisor: John L. Wyatt
Title: Professor

Acknowledgements

There do not exist words to describe the magnificent impact that my mother, Lena Valavani, has had on my life. I am extremely lucky to be her son as she is the best parent that someone could wish for. Her guidance and love have propelled me to where I am today. She has always supported me and has sacrificed a great deal to make me who I am. I have always looked up to and admired her as she has been my role model: she is extremely smart and has a dynamic personality. Moreover, she is loved and respected by everyone around her. However, I think that what I admire most about my mother is that she is young at heart; because of this, we have always had a special relationship unlike any other mother-son relationship I have seen or heard of. As she is a former MIT faculty member, it should come as no surprise that it was she who ignited my passion for mathematics and engineering: as it turns out, the apple does not fall far from the tree! One thing is for sure: I have very big shoes to fill! I would like to take this opportunity to thank her for all that she has been, done, and dreamed for me: I am eternally grateful!

To my grandfather, Stavros Valavanis, grandmother, Victoria Valavani, and uncle, Kostas Valavanis: you left this world too early. I am grateful for all the beautiful childhood memories you have given me; I will cherish them for a lifetime.

To my advisor John Wyatt: You are one of the smartest people I have met. I thank you for both of your roles in my life: advisor and close friend. I have learned a great deal from working with you. Apart from the academic knowledge I have acquired from you, I also have learned a great deal about character. Our interaction has been extremely rewarding. I thank you for all the time you allotted to help me, both in research and personal matters. I am especially grateful for your interest in my research and progress: I will be lucky if future collaborators spend half as much time on working with me! I greatly enjoyed the political and social conversations we had throughout our collaboration. Without your guidance, this thesis would not have been possible. I look forward to future collaboration with you inside and outside of MIT and hope that you will be on my PhD thesis committee!

To Shelley, Steve, and Karl: your expertise and patience have made the experimental work for this thesis possible. I really appreciate how generous you have been with your time. It was very fun working with all of you.

To Adam "chucha" Eisenman: the office is not the same without you! We had so much fun last year, whether it was partying in the office our in Miami, we definitely left our mark everywhere we went! Congratulations on your upcoming marriage with Violulis! I hope to be able to make it-it's going to be great! You are very lucky to have found your soul mate: having seen you and Violulis together, I know that your marriage will be very successful and you will be happy for the rest of your life! I would also like to acknowledge the fact that you wrote chapters 3 and 4 of this thesis! Although the work was joint, you bore the grunt of writing it and I am grateful for that! I think that few collaborations in history have been as successful as ours has been. I learned a lot from you: without you, I would not be as proficient in MATLAB, nor would I have the intuition that I do about how the retina works. You are a great engineer, a magnificent person, and one of the craziest people I have met! I will always be nostalgic of the two semesters in which I shared my office with a crazy guy from Panama!

To Shamim and Jessica: you have both contributed intense amounts to the latter part of this thesis! Even though you were in your first semester in the group, you both jumped on the ball quickly! You are both very smart and it was a joy to work with you both. Shamim, your intuition in Linear Algebra and your numerical optimization prowess are admirable. Jessica, your proficiency in handling data and superior programming skills are remarkable. Thanks to both of you for your help: it was great working with you!

STAVROS VALAVANIS – January, 2007

This Thesis is dedicated to
my beloved mother, *Lena Valavani*

Table of Contents

List of Figures

Chapter 1

Introduction

Wisest is he who knows that he does not know...

-Socrates

When I started looking for a Master's thesis topic, I envisioned working on cutting edge research project which involved rigorous mathematical analysis applied to an interesting and inherently difficult problem whose solution would benefit humanity. I am glad to say that this thesis, which is a branch of the Boston Retinal Implant Project, was the ideal match for my vision, needs, and aspirations. Professor Wyatt offered me the opportunity to work on a pioneering effort whose long term goal is to understand the retinal neural code. The purpose of this thesis is to shed light on the field of retinal neural coding by proposing, simulating, and testing algorithms which estimate motion parameters of optical stimuli given an ensemble of Retinal Ganglion Cell (RGC) responses to these stimuli. We are not asserting that any of these algorithms are used by the brain to decode motion information obtained from the retina. Rather, we seek to demonstrate that information about visual motion which is encoded in RGC spike train responses can be decoded.

The Boston Retinal Implant Project serves as a springboard to study retinal neural coding. Its objective is to restore partial vision in patients with Retinitis Pigmentosa and age related Macular Degeneration. These particular conditions affect the retinal photoreceptor layer, leaving it dysfunctional in most cases. However, these degenerative diseases leave the ganglion cell layer almost entirely functional (Medeiros et al. 2001 [9]). The latter layer is responsible for transmitting the visual information to the brain through the optic nerve. Hence, by electrically stimulating the ganglion cell layer, one could presumably obtain visual perceptions in the brain.

In order to effectively code a retinal implant circuit's stimulating signals, the structure of the stochastic map from optical signals to retinal ganglion cell firing patterns

would have to be understood. If this mapping can be modeled accurately, it will be possible to produce a coding scheme that can be decoded by the brain. It is natural to focus on the retinal ganglion cells as they are the only retinal cells that feed signals to the brain; moreover, this connection is only feed-forward (Grossberg et al. 1997 [7]). Theoretically, this implies that if we were able to replicate the spatio-temporal spiking pattern caused by a specific light pattern in every single ganglion cell of a healthy retina via electrode stimulation, the brain would perceive that specific light pattern. In other words, we would be mimicking the behavior of each RGC's electrical response to a light pattern thus enabling the brain to "decode" the stimulus.

A related problem which helps us understand the aforementioned problem is what we call the "Inverse Problem". The statement of the Inverse Problem is as follows: given a set of spatio-temporal RGC responses, what can we infer about the optical stimulus video that was shown to the retina? In a sense, this process is the inverse of that carried out to stimulate RGCs from a retinal implant. The logic of the Inverse Problem is depicted in Figure 1-1. The aim of this thesis is to develop algorithms which solve the Inverse Problem for a certain family of optical stimuli: moving edges of light. This will be accomplished by suggesting and studying models that aid in finding estimates for the parameters that describe a stimulus in a parameterized set. The overall problem which will be addressed in this thesis is described in section 1.1; a description of typical RGC responses to moving edges of light is given in 1.2; a description of each chapter and the logical flow of this thesis are given in section 1.3.

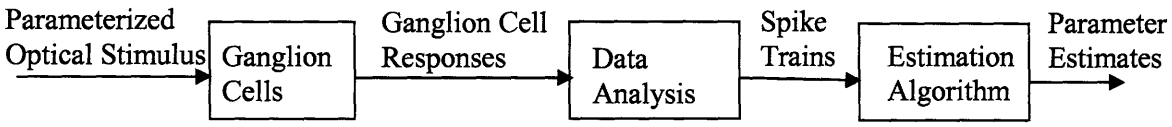

Figure 1-1: This figure depicts the process of estimating visual stimulus parameters from RGC recordings. The optical stimulus elicits responses from the RGCs; these RGC responses are recorded and analyzed so as to obtain a spike train for each RGC; a RGCs spike train signifies the times at which the cell "fired". Estimation algorithms take these spike trains as inputs and output the parameters of interest of the visual stimulus.

1.1 Problem Description

This thesis focuses on one of the simplest statements of an Inverse Problem. We wish to make estimates of the speed and direction of moving edges of light which are presented to a piece of rabbit retina. The speed and direction of these edges are constant throughout their time of motion. These edges can be a dark to light transition (we call these ON edges) or they can be a light to dark transition (we call these OFF edges). Note that the luminance intensity is not altered throughout the time of motion. Moreover, each edge can be wide (we call these curtains) or narrow (we call these bars). In the case of curtains, they are either bright over a dark background or dark over a bright background. In the case of bars, they are bright over a dark background and they are of finite length (i.e. they have a leading and a trailing edge): thus, a single bar causes an ON effect followed by and OFF effect on the retina. Graphical representations of these stimuli are presented in figures 3-3 and 3-4 of chapter 3 of this thesis. For each cell, the occurrence times of the action potentials produced are recorded; the set of all such occurrence times for a cell in response to a stimulus is called a spike train. The algorithms we employ to estimate the speed and direction of a moving edge take as input the spike train response of every cell. There are two types of estimation algorithms we develop: the first type uses only first order statistics of each cell's spike train response whereas the second type uses the entirety of each cell's spike train response.

1.2 RGC Firing Patterns

There are four general categories of retinal ganglion cells which we use in our estimation procedures:

- **OFF cells:** these cells increase their firing rate above their spontaneous firing rate when a light do dark stimulus is presented to them. They also decrease their firing rate below their spontaneous firing rate when a dark to light stimulus is presented

to them. Thus, they fire vigorously when a light to dark edge passes over them; their firing is inhibited when a dark to light edge is passed over them. An example of the firing of such a cell in response to both ON and OFF curtains is given in figure 1-2.

- **ON cells:** these cells increase their firing rate above their spontaneous firing rate when a dark to light stimulus is presented to them. They also decrease their firing rate below their spontaneous firing rate when a light to dark stimulus is presented to them. Thus, they fire vigorously when a dark to light edge passes over them; their firing is inhibited when a light to dark edge is passed over them. An example of the firing of such a cell in response to both ON and OFF curtains is given in figure 1-3.

- **ON-OFF cells:** these cells increase their firing rate above their spontaneous firing rate when presented with either a dark to light or light to dark stimulus. Thus, they fire vigorously in response to both a dark to light edge and a light to dark edge.

1.3 Thesis Layout

This thesis is an amalgamation of the research I pursued jointly with Adam Eisenman in the academic year 2006-2007 and the research I pursued with the important input of Jessica Wu and Shamim Nemati during the fall of 2007. Thus, this thesis is divided into two parts: the first part (chapters 3 and 4) contains work that was done jointly with Adam. The work in these chapters is based on estimation algorithms which use first order statistics of RGC spike train responses: these algorithms are "least squares" algorithms. Adam has graciously allowed me to use these chapter write-ups from his thesis; I give him much kudos for writing them up in a clear and coherent manner. Although the work was done jointly, Adam bore the grunt of writing it up and I thank him for this. Other work which I did jointly with Adam is contained in chapters 5, 6, and 7 of Eisenman, 2007 [1]. However, I omit it from this thesis so as not to be repetitive. The second part (chapters 5, 6, 7, and 8) contains work done during the fall of 2007 in collaboration with

Jessica Wu and Shamim Nemati. The work in these chapters is based on an algorithm which uses the entirety of each cell's spike train response.

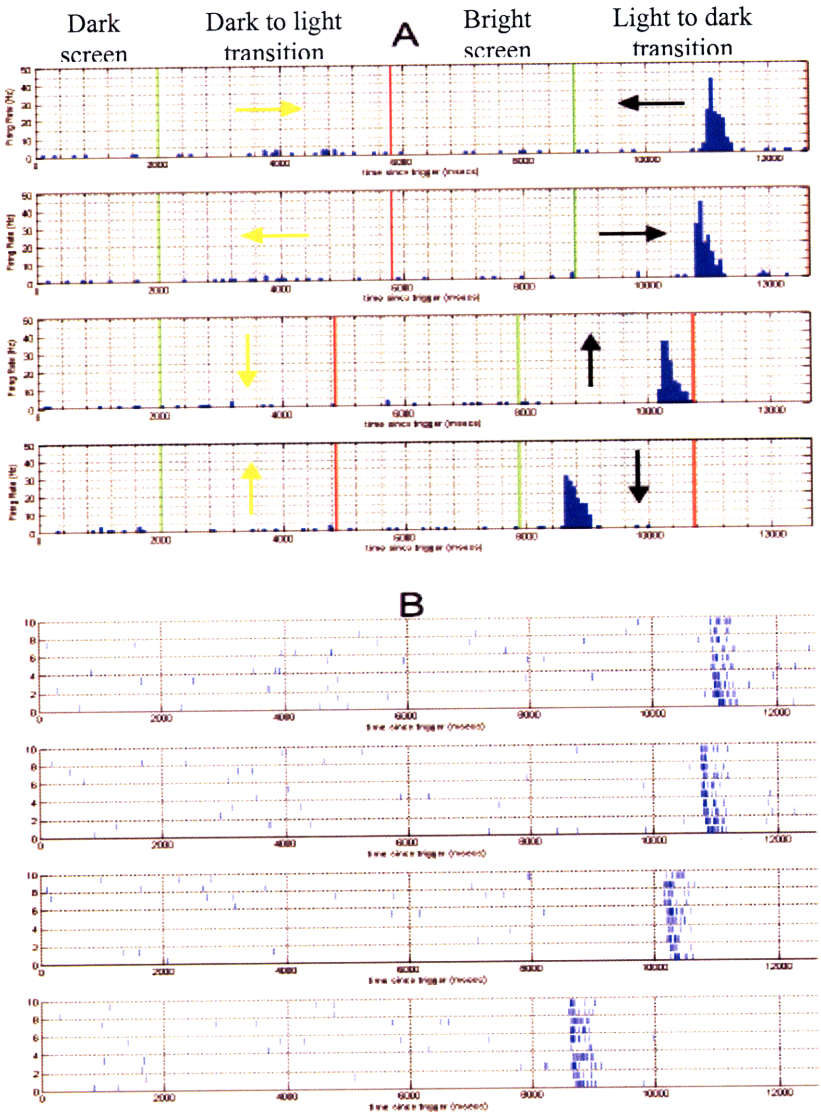

Figure 1-2: This figure depicts the response of an OFF cell to a curtain moving in the 4 cardinal directions. Panel A shows the Peri Stimulus Time Histogram (PSTH) computed over 10 trials of the response of an OFF cell to the motion of an ON curtain in these 4 directions and the motion of an OFF curtain in these 4 directions. Yellow arrows represent the motion of an ON curtain. Black arrows represent the motion of an OFF curtain. The green lines represent the times at which the curtains begin to move in each corresponding direction, while the red lines represent the time at which the curtains stops moving. For example, let's consider the first row of panel A: for the first 2000ms, the screen is dark. A bright curtain enters from the left of the screen at 2000ms and moves rightwards across the screen until 5800ms. Between 5800ms and 8800ms, the screen is covered by the bright curtain. At 8800ms, the bright curtain moves leftwards until it exits the screen at 12600ms. Panel B depicts the cell's response to the same stimulus repeated 10 times. Each spike train corresponds to a single trial of a moving curtain. Each of the four PSTH's of panel A is essentially the average firing of the cell over the ten trials depicted in the corresponding "row" of the plot in panel B.

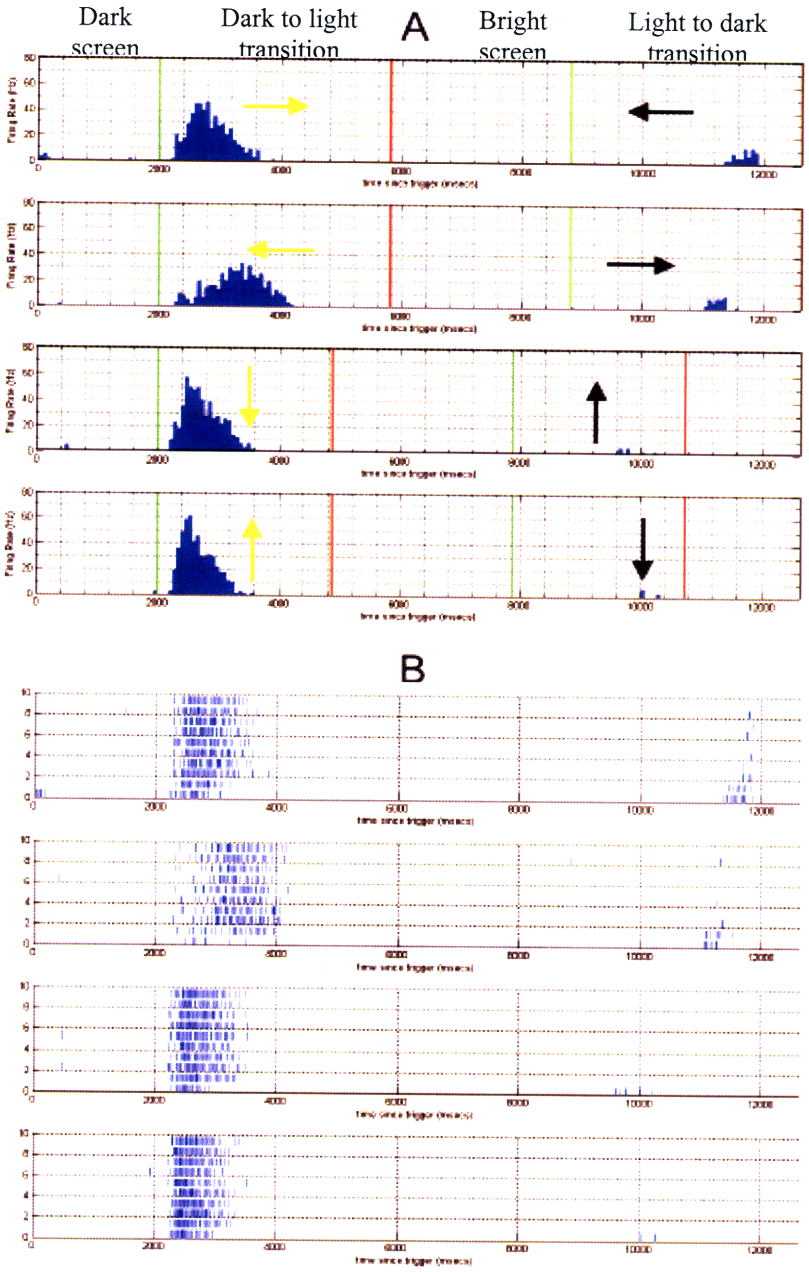

Figure 1-3: This figure depicts the response of an ON cell to a curtain moving in the 4 cardinal directions. Note that these are the same stimuli as those in figure 1-2. Panel A shows the Peri Stimulus Time Histogram (PSTH) computed over 10 trials of the response of an ON cell to the motion of an ON curtain in these 4 directions and the motion of an OFF curtain in these 4 directions. Yellow arrows represent the motion of an ON curtain. Black arrows represent the motion of an OFF curtain. For example, let's consider the first row of panel A: for the first 2000ms, the screen is dark. A bright curtain enters from the left of the screen at 2000ms and moves rightwards across the screen until 5800ms. Between 5800ms and 8800ms, the screen is covered by the bright curtain. At 8800ms, the bright curtain moves leftwards until it exits the screen at 12600ms. The green lines represent the times at which the curtains begin to move in each corresponding direction, while the red lines represent the time at which the curtains stops moving. Panel B depicts the cell's response to the same stimulus repeated 10 times. Each spike train corresponds to a single trial of a moving curtain. Each of the four PSTH's of panel A is essentially the average firing of the cell over the ten trials depicted in the corresponding "row" of the plot in panel B.

More specifically, chapter 3 describes the experimental set up in great detail; Steven Stasheff and Shelley Fried were particularly helpful in teaching us to do the experiments required to generate the data on which we test our algorithms' performance. Chapter 4 contains the mathematical description of our problem. It also contains the derivations of four least squares based estimation algorithms (the "Estimating Velocity Vector Directly" algorithm, the "Global Firing Time Information Algorithm", the "CosCos" algorithm, and the "Newton Raphson Minimization" algorithm). Furthermore, for each algorithm, an analysis is performed of the noise sensitivities of the estimates as a function of the noise in the measured parameters which we input to the algorithms. Simulations and data testing of these algorithms can be found in chapters 5 and 7 of Eisenman, 2007 [1]. Chapter 5 of this thesis presents a novel way of weighing least squares so as to minimize a *weighted sum of the variances* of a set of parameter estimates which we are interested in. At first glance, this chapter may seem like a digression from the rest of this thesis; however, the goal of this work is to find a way to reweigh least squares problems in which we are interested in the accuracy of some estimates more than others. Chapter 6 develops an estimation algorithm which utilizes maximum likelihood estimation: in contrast to the algorithms of chapter 4, this algorithm does not use least squares. For each cell, a model is obtained through "training". The likelihood of each cell's response is then computed and the joint likelihood of all cells' responses is maximized. Chapter 7 compares the results of the "likelihood algorithm" of chapter 6 and the global firing time information algorithm of chapter 4 on real data. Chapter 8 presents and compares simulation results of both the likelihood and global firing time information algorithms. It explores the effect of using a small number of cells as well as the effect of using cells which are clustered closely in each algorithm. Chapter 9 reviews the conclusions of this thesis and proposes further work to be done.

Chapter 2

Survey of the Current Literature

A survey of the current literature on the subjects of the Inverse Problem and Likelihood Methods is presented in this chapter. It served as a guideline outlining methods that had been used in the past and what had already been accomplished. It also provided insight into what could be done differently to advance the field. Section 2.1 describes work that is relevant to understanding retinal ganglion cell differentiation; section 2.2 presents work relevant to the Inverse Problem; section 2.3 surveys how Likelihood Methods are currently being used in neural decoding.

2.1 Ganglion Cell Types and Functions

Mammalian retinal ganglion cell morphological types are relatively well defined (Rockhill et al. 2002 [9]); Rockhill found 11 types of morphologically distinct RGCs. Moreover, RGCs have distinct physiological characteristics: some respond maximally to an onset of bright light (ON cells), some respond maximally to an offset (OFF cells), and some respond to both onsets and offsets (ON-OFF cells). Moreover, within each of the 3 groups described above, there exist subgroups: within each group, some cells are more latent than others in their responses and some cells' responses have a higher duration than others. Devries et al. 1997 [5] conducted anatomic and physiological studies on the mammalian retina and showed that the receptive fields of several types of ganglion cells tile the retinal surface. Carcieri et al. 2003 [4] clustered mouse retinal ganglion cells into physiological types as follows: they project a flashing light onto a cell's receptive field in such a way that it responds maximally. They used a cluster analysis approach from which they found that mouse ganglion cells clustered into several groups based on 3 standard response parameters: *1*) response latency, *2*) response duration, and *3*) relative amplitude of the ON and OFF responses.

2.2 The Inverse Problem

The problem of decoding the neural code of the retina has been studied in the past (Stanley et al. 1997 [11], Frechette et al. 2005 [6], and Guillory et al. 2006 [8]). The traditional way of estimating the stimulus that was shown to the retina has been to first characterize the response properties of each cell in the ensemble using an optimal (in the mean-square sense) LTI filter and then filter the subsequent response of these cells using each of their filters to obtain a visual stimulus estimate. Stanley et al. 1997 [11] used a linear decoding technique to reconstruct spatiotemporal visual inputs from ensemble responses in the Lateral Geniculate Nucleus (LGN) of the cat. From the activity of 177 cells, they reconstructed natural scenes with recognizable moving objects. The quality of reconstruction depended on the number of cells. For each point in space, the quality of their reconstruction began to saturate at six to eight pairs of ON and OFF cells which approached the estimated coverage factor in the LGN of the cat. The method above however gives no information about how the geometrical layout of the cells used affects the reconstruction. Moreover, the decoding method assumes that all information is coded in the firing rates of neurons.

Frechette et al. 2005 [6] examine how speed of movement is encoded in the population activity of magnocellular-projecting parasol retinal ganglion cells (RGCs) in macaque monkey retina. They adopt a different approach as they record responses at the level of the retina; in this manner, they take advantage of the cells' geometrical layout in space because these positions roughly correspond to the location of the cells' receptive fields. They propose a model which, by taking into account the delay between the responses of pairs of cells, gives an estimate of the speed of a moving curtain. In essence, they solve the problem of estimating a curtain's speed given that its direction of motion is known. They pair cells up and estimate each cell pair's time difference of responses by finding the peak of the cross-correlation of the two cells' spike train responses. Given that they know the cells' locations, they are able to estimate speed. They conclude that temporal structure in spike trains provides more precise speed estimates than time-varying firing rates; moreover, correlated activity between RGCs has little effect on speed estimates. The Inverse Problem which is proposed and solved through various

algorithms in this thesis is an extension of the one presented in Frechette et al. 2005 [6] because we seek to estimate both the speed and direction of a moving edge simultaneously: the fact that our problem is two-dimensional makes it much more complicated than the one in Frechette et al. 2005 [6].

Guillory et al. 2006 [8] infer the color of a stimulus, given the activities of 18 retinal ganglion cells. They use a point-process framework consisting of an inhomogeneous Poisson model of neural firing combined with a refractory renewal period following each spike. The instantaneous rate function is taken to be the smoothed PSTH for each cell. They decode via a likelihood framework; the inclusion of the refractory behavior of neurons into their model only marginally (less than 1 percent) improved the decoding performance.

2.3 Likelihood Methods

Likelihood methods have become popular in neural spike train decoding over the last 10-15 years. Their main advantage is that they are extremely generalizeable. Brown et al. 1998 [2] tackles the problem of predicting the position of a freely foraging rat based on the ensemble firing patterns of place cells recorded from the CA1 region of its hippocampus. They develop a two-stage statistical paradigm for neural spike train decoding: in the first stage, they model place cell spiking activity as an inhomogeneous Poisson process whose instantaneous rate is a function of the animal's position in space and phase of its theta rhythm. In the second stage, they use a Bayesian statistical paradigm to derive a nonlinear recursive causal filter algorithm for predicting the position of the animal from the place cell ensemble firing patterns. Of interest to us was their model for place cells: the inhomogeneous Poisson rate parameter is position dependent and modeled as a Gaussian function defined as:

$$\lambda(t) = \exp\left\{\alpha - \frac{1}{2}(\mathbf{x}(t) - \boldsymbol{\mu})^T W^{-1}(\mathbf{x}(t) - \boldsymbol{\mu})\right\} \tag{2.3.1}$$

where α is the place cell firing intensity parameter, $\mathbf{x}(t) = [x_1(t), x_2(t)]^T$ is the vector denoting the animal's position at time t, $\boldsymbol{\mu} = [\mu_1, \mu_2]^T$ is the vector whose coordinates are the location of the place field center, and $W = \begin{bmatrix} \sigma_{x_1}^2 & 0 \\ 0 & \sigma_{x_2}^2 \end{bmatrix}$ is a scale matrix whose scale parameters in the x_1 and x_2 directions are $\sigma_{x_1}^2$ and $\sigma_{x_2}^2$ respectively. They set the off-diagonal terms to 0 because they claim that those parameters are statistically indistinguishable from 0. In chapter 6, we model the receptive fields of retinal ganglion cells with a variant of the above model for place cells. However, our decoding stage is non-Bayesian in nature as we obtain maximum likelihood estimates for our speed and direction parameters.

Truccolo et al. 2004 [12] derive a point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. Of interest to us is the statistical framework they propose which is based on the point process likelihood function: with this framework, they relate a neuron's spiking probability to three typical covariates: the neuron's own spiking history, concurrent ensemble activity, and extrinsic covariates such as stimuli or behavior. Their framework uses parametric models of the conditional intensity function to define a neuron's spiking probability in terms of the covariates. Their point process framework provides a flexible, computationally efficient approach for maximum likelihood estimation, goodness-of-fit assessment, residual analysis, model selection, and neural decoding. Of interest to us is their definition of the conditional intensity function, which is exactly the inhomogeneous Poisson rate. In chapter 6, we use an adaptation of their method in deriving the joint likelihood function of a set of spike train responses.

2.4 Acknowledgements for Chapters 3 and 4

Before continuing, I would like to remind the reader that the next two chapters, chapter 3 (Experimental Procedures) and chapter 4 (Theoretical Developments on Least Squares Algorithm) were joint work with Adam Eisenman but they were written up in Adam's thesis; he has graciously allowed me to use them in this thesis and I thank him for this! Of course, Professor John Wyatt has agreed to this as well.

Chapter 3

Experimental Procedures

The experiments that we found necessary for the completion of this thesis took place in the Cellular Neurobiology Laboratory (Masland Lab) at Massachusetts General Hospital (MGH) under the supervision of neurophysiologists Steven Stasheff, MD, PhD, Shelley Fried, PhD, and Karl Farrow, PhD. More specifically, Dr. Shelley Fried performed the surgery and dissection of the retinal piece, and Dr. Steven Stasheff mounted the retinal piece onto his multi-electrode array (MEA) set-up. Dr. Karl Farrow provided help with the system setup and debugging.

This chapter commences with a description of the procedures that took place in order to prepare the rabbit retinal tissue on which we ran experiments. Next, we describe the MEA set-up and its interface with the retinal piece. Subsequently, we give a description of the visual stimuli that were presented to the retinal piece along with the optical machinery required to perform the presentation task. Lastly, we explicate the procedures for assigning spike times to each cell from which the MEA recorded electrical activity.

With the purpose of having multiple trials on which to test our analysis, we performed experiments on different days; each day on a retinal patch coming from a different rabbit. On any given experimental day, we chose to run a subset of the experiments described in this chapter.

3.1 Tissue Preparation

New Zealand white rabbits of either sex (3-5 kg) were anesthetized with xylazine (5-10 mg/kg) and ketamine (30-100 mg/kg) to the point that the corneal reflex was abolished. The animal was enucleated, the globe hemisected, and the vitreous removed. The animal was killed with an overdose of ketamine, according to a protocol approved by the Subcommittee on Research Animal Care of the Massachusetts General Hospital. Under infrared illumination to minimize exposure to visible light, using a dissecting microscope (Leica Microsystems, Inc., Bannockburn, IL) with infrared image intensifiers (BE Meyers, Inc., Redmond, WA), the retina was dissected from the retinal pigmentary epithelium. Next, it was placed ganglion cell layer down onto a multi-electrode recording array (10 μm in diameter circular contacts spaced 200 μm apart; Multichannel Systems, Reutlingen, Germany) in a recording chamber attached to a microscope stage, and superfused at 2.5-3.5 mL/min with warm (33-37°C) Ames' medium. Subsequently, the retina was allowed to sit in the dark for approximately one hour so as to become less hyperactive and "settle down"; once we decided that the retina was not hyperactive, we proceeded with our experiments.

3.2 Multi-electrode Recordings

A square (1.4mm side-length) MEA (seen in Figure 3-1 with 4 corner electrodes not present; 10μm electrode diameter, spaced 200μm apart) followed by a 60-channel amplifier (Multi-channel Systems, Reutlingen, Germany) mounted on a microscope stage (Zeiss Axioplan, Göttingen, Germany) interfaced with digital sampling hardware and software (Bionic Technologies, Inc., Salt Lake City, UT) for recording and analyzing spike trains from each of the electrodes in the array. Digitized data initially were streamed onto the computer's hard drive and further analyzed offline. After transfer of the retina to the recording chamber, recordings were allowed to stabilize for at least one hour, as evidenced by stable action potential amplitudes, number of cells recorded, frequency of spontaneous firing, and consistency of light-evoked responses.

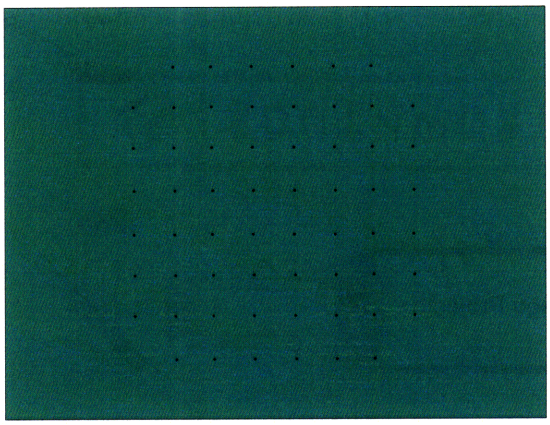

Figure 3-1: MEA and image projection area drawn to scale. The MEA's side has a length of 1.4mm. Each electrode has a 10μm diameter. The shape of the image Projection Area is rectangular (height 2.038mm, width 2.718mm)

The MEA recording system samples waveforms at 30 kHz. If a digitized waveform exceeds a user-defined threshold, it is stored in memory along with its occurrence time. These thresholds (one for each channel/electrode) are set in such a way so as to minimize the recording of events other than action potentials. In this manner, only action potentials and their corresponding occurrence times are stored in memory; faulty waveforms are discarded.

3.3 Visual Stimulation

In experiments with light stimulation, a miniature computer monitor (Lucivid, MicroBrightField, Colchester, VT) projected visual stimuli through a 5× objective; these were focused onto the photoreceptor layer of the retina with the help of a mirror (depicted in Figure 3-2). Luminance was calibrated via commercial software (VisionWorks, Vision Research Graphics, Durham, NH), using a photometer (Minolta, Ramsey, NJ) and photodiode placed in the tissue plane. The refresh rate of the monitor was 66 Hz. The same software controlled and recorded stimulus parameters, passing synchronization pulses to the data acquisition computer via a parallel interface with 10 μsec precision. The purpose of these synchronization pulses was to give us an indication of when the spikes occurred relative to what occurred on the image

plane.

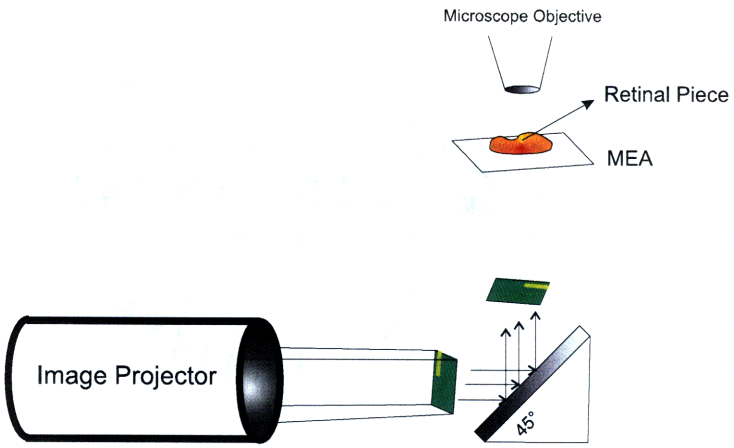

Figure 3-2: The image, produced by the image projector is deflected off of a flat mirror which is inclined at 45°. The image passes through the transparent (the effect of the 10 μm non-transparent electrodes is very small) MEA, and focuses on the photoreceptor layer of the retina. The MEA makes contact with the RGCs.

We stimulate the retinal piece with various stimuli which were crafted using commercially available software (VisionWorks, Vision Research Graphics, Durham, NH). The projected images are pixelated with 800×600 resolution. The following is a description of the stimuli which we presented.

3.3.1 Bright and Dark Curtains Moving at Various Speeds and Directions

A curtain is a moving edge of light (bright over a dark background *or* dark over a bright background) which progressively covers the projection area. The edge of light moves at a constant speed and direction. For the sake of clarity we describe the sequence of events which define an ON curtain: 1) The background is dark, 2) An edge (which separates dark from bright) comes onto the projection area, 3) The portion on the bright side of the edge grows bigger and bigger until the projection area is completely bright. An OFF curtain is defined similarly, except that the background is initially bright and ends up being dark.

During a given experiment day we ran ON and OFF curtains in 4, 8, or 16 different

directions. The angles at which we run the curtains are evenly spaced over the range of 360°. For example, motion in 16 directions occurs at 0°, ±22.5°, ±45°, ±67.5°, ±90° ±112.5°, ±135°, ±157.5°, and 180°. The curtains were designed to move along two axes (horizontal and vertical). However, to obtain more than 4 directions we rotated the projected image accordingly (seen in 3-3). On a given day we run the curtains at a subset of the following speeds: 300, 357, 600, 714, 1200, 1428, 2400, and 2856 μm/sec. We repeated motion of each curtain at every contrast (ON or OFF), speed, and direction 10 times. This is done so that we can do statistical analysis of the cell firing patterns. We wait at least 2 seconds between the end of one curtain and the beginning of the next curtain motion. Figure 3-3 depicts motion of ON and OFF curtains in all 16 directions.

3.3.2 Finite Length Thin Bars Moving at Various Speeds and Directions

Bright rectangular bars which are narrow compared to the size of the projection area (height 2.038mm, width 2.718mm) were moved over a darker background. The contrast between the bright bars and the darker background was the same as the contrast described for the curtains. The dimensions of the bars were either 300μm × 900μm, or 357μm × 1071μm, depending on the experiment. A single bar causes ON and OFF effects due to its leading and trailing edges, respectively. These bars were moved across the retinal piece at various directions and speeds. The bar stimuli were prepared in such a way that the whole projection area would be swept by the moving bars.

Similar to the curtain stimuli, the bar stimuli were designed so that all motion occurred along two axes (horizontal and vertical). The projected image would be rotated accordingly (as was seen in the previous section, for curtains) depending on the angle at which we wanted to move the bars. We ran the bars in 4, 8, or 16 directions depending on the experimental day. For example, motion in 16 directions occurs at 0°, ±22.5°, ±45°, ±67.5°, ±90° ±112.5°, ±135°, ±157.5°, and 180°. On a

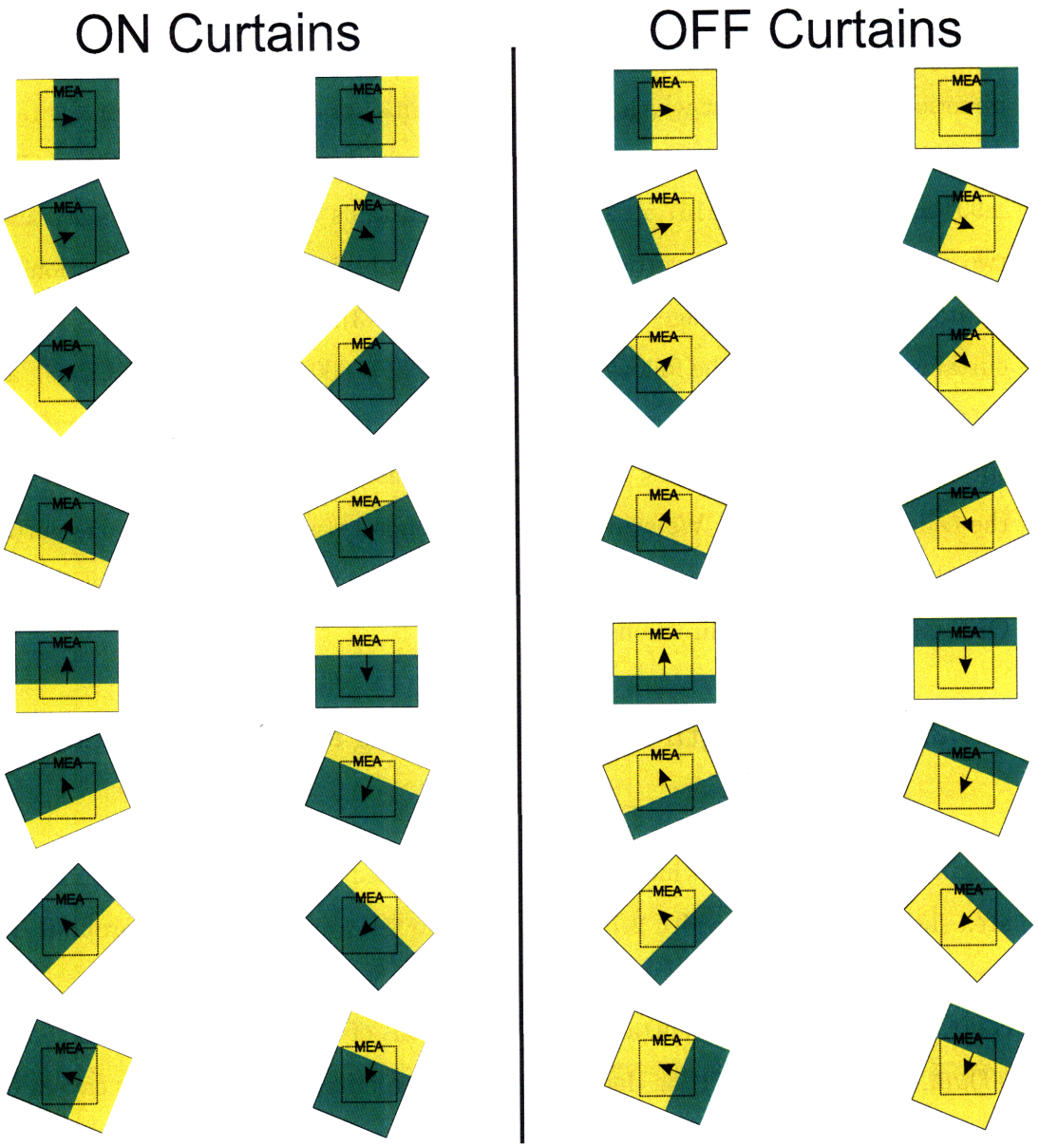

Figure 3-3: The MEA is always fixed. The projected image is rotated with respect to the MEA (by rotating the projection device) to obtain the effect of curtain motion in various directions. The left half of the figure depicts motion of a bright curtain over a darker background (ON effect). The right half depicts motion of a dark curtain over a bright background (OFF effect). The directions of motion are 0°, 180°, ±22.5°, ±45°, ±67.5°, ±90° ±112.5°, ±135°, and ±157.5°. In addition, these curtains are moved at various speeds, as described above. The order in which the different directions, speeds and contrasts (ON or OFF curtain) were shown to the retinal piece vary from one experimental day to another, but is explained in subsection 3.3.3

given day we run the bars at a subset of the following speeds: 300, 357, 600, 714, 1200, 1428, 2400, and 2856 μm/sec. We repeated motion of each bar stimulus (given speed and direction) 5 times.

We ran the bar stimuli with the purpose of estimating the speed and direction of the moving bars. In addition, as mentioned in Chapter 1, we noticed that DS cells lose much of their directional selectivity when stimulating them with curtains; therefore, it was helpful to detect the presence of such cells using moving bars. We make the bars overlap (by half a bar width) to obtain better resolution when locating DS cells. Figure 3-4 depicts the bar stimuli for motion along both axes (horizontal and vertical).

3.3.3 Visual Stimulation Protocol

As soon as the retinal firing had "settled down," we recorded 10 minutes of spontaneous activity. These recordings were used after the experiment to check for any patterns of recognizable noise in the spontaneous firing patters of the RGCs.

Next, we proceeded with the retinal visual stimulation in one of two possible ways:

1. We ran curtains (at various speeds) and then various speeds of bars (10 and 5 times respectively) in 4 directions. If on the given experimental day we ran more than 4 directions, we then rotated the projector and ran the curtains and then the bars at the same speeds, 10 and 5 times respectively, in 4 new directions. We proceeded with this protocol until we had finished with all the directions that were run on a given experimental day.

2. We ran a set of curtains (at a single speed) and various speeds of bars in 4 directions, 1 time each. We then rotated the projector and ran the same set of stimuli in 4 new directions. We then rotated back to the original projector position, and started over. This was done 5 times to obtain 5 repetitions of bars moving at various speeds in 8 directions and a set of curtains moving at a single speed in 8 directions.

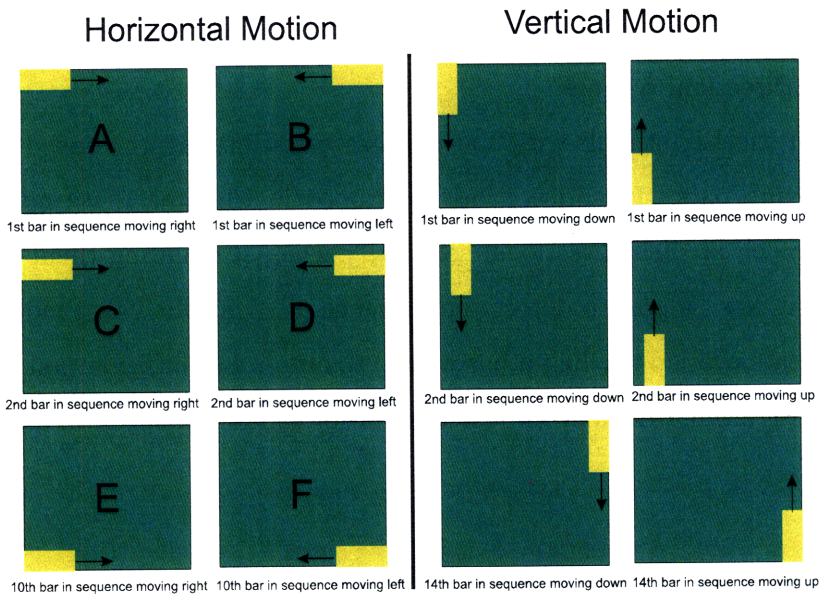

Figure 3-4: **Horizontal Motion**: The left half of the figure depicts the manner and the order in which horizontal bars sweep the projection area. In words, the order of events is as follows: A) First bar comes into the projection area along the top of the screen, this bar keeps moving to the right at a constant speed until its back edge reaches the end of the screen (now the screen has no bright elements on it), 3 waiting seconds pass with no motion, B) The same bar comes back into the projection area and moves along the same line (in the opposite direction) until what is now its back edge reaches the end of the screen (the screen has no bright elements on it), 3 waiting seconds pass with no motion, C) The next bar moves along a line half a bar width under the line of motion of the first bar, D,E,F) This is repeated until all 10 bars have moved back and forth. This way the screen is more than swept (the bars overlap by a half bar width). **Vertical Motion**: The right half of the figure depicts a scenario analogous to the one described for horizontal motion. In this picture we see that the bars move along the vertical axis. Due to the rectangularity of the screen, we need to move 14 bars instead of 10. The bars overlapped by half a bar width also, and more than swept the screen. The order in which the different directions, and speeds were shown to the retinal piece vary from one experimental day to another, but is explained in subsection 3.3.3

The purpose of running protocol #1 is to leave as little time as possible between repetitive runs of each stimulus. However, if protocol #1 is run, there are big gaps between the time in which a set of bars was run in one direction and some of the other directions. This is problematic because the state of the retinal tissue is not constant over time. Therefore, protocol #2 is necessary to make reliable DS polar plots[1].

3.4 Spike Waveform Analysis

Action potential (spike) waveforms accepted for further analysis were at least 60 μV in amplitude and greater than 1.85 times the RMS of the background signal. To distinguish responses from different cells that might appear on the same electrode, PowerNap, a component of the data acquisition software (Bionic Technologies, Inc., Salt Lake City, UT), was used for supervised automated sorting of action potential profiles according to a principle components analysis (PCA) paradigm. For each electrode, the software displays all of the waveforms recorded in a window of length 1 msec. Each of these waveforms is decomposed into its first three principal components and placed as a point in three-dimensional space. Principal components are the eigenvectors computed from the correlation matrix of all the action potentials recorded at each electrode [?]. We are able to view all three two-dimensional projections of each waveform in the space defined by the first three principle components (Figure 3-5 shows the projection onto their space defined by the first two components).

The individual waveforms were partitioned iteratively into 1-5 clusters according to an automated K-means paradigm [?], an algorithm used to minimize the total intra-cluster variance. With the help of the K-means algorithm, followed by further manual assignment of waveforms to specific clusters, we try to: 1) Maximize the similarity among waveforms within a cluster; 2) Minimize the degree of overlap between clusters, and 3) Maximize the distance between cluster centers and edges. In cases where an optimal solution was not immediately distinguished on this basis, the data initially

[1]DS polar plots give a measure of how much a DS cell fired for motion over all experimented directions.

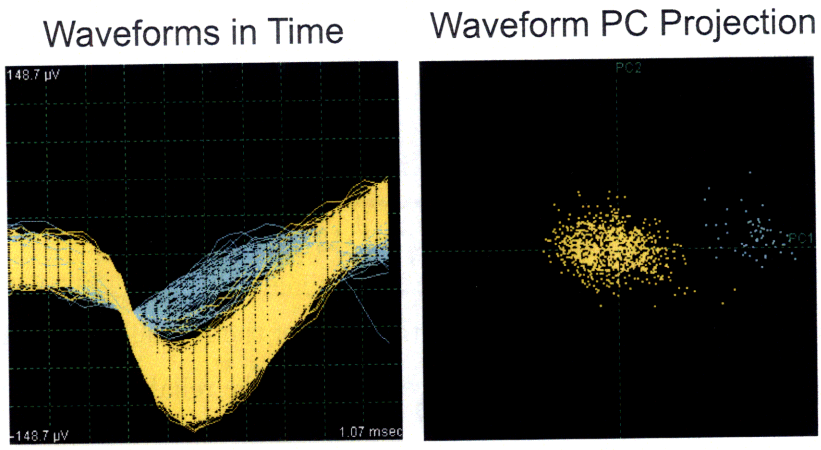

Figure 3-5: On the left of the figure, we see the samples of action potential waveforms coming from two different cells recorded on the same electrode. On the right, we see the projection of each of these transformed vectors onto the 2-D space defined by the first two principal components of the data on the electrode. It can be seen that the yellow and white clusters are gracefully separate. In the time domain, it can be seen that waveforms which were clustered together look very similar.

was segregated into a greater number of clusters than seemed the likely final solution. This was followed by subsequent analysis of the corresponding spike trains (described below), to determine which of these signals were generated by the same or distinct sources. In the cases with broad and overlapping clusters, individual waveforms were considered outliers and excluded if their projected point in PC space was distant from the closest cluster's center by greater than 2.5-4.0 times the standard deviation of the data within that cluster. Appropriate assignment of individual waveforms to distinct cells was confirmed further by analysis of the corresponding spike trains. Inter-spike interval (ISI) histograms were computed for each spike train by measuring the intervals between spikes in the train for all possible spike pairs within a candidate cluster, and then distributing these values in bins of 0.2 msec width. ISI histograms from accepted data demonstrated a refractory period of at least 1 msec (typically 2-5 msec) and did not reflect any of the following patterns of recognizable noise: 60 Hz, very high frequency ($>$ 10 kHz) transients, or waveforms distinct from those of extracellular action potentials (e.g. sinusoidal oscillations).

Once the spike sorting for a particular experiment (e.g. curtains in a certain direction, moving at a certain speed) was done, the results were used as a basis to

sort the rest of the experimental data files. For example, if a cell with a particular action potential shape on electrode 55 was named unit 1, then it was verified that for every other data file, unit 1 on electrode 55 had the same action potential shape. This assures us that when we refer to the firing of a particular cell across two different experiments (e.g. curtains in a given direction at two different speeds), we know that we are referring to the same cell.

Chapter 4

Theoretical Developments on Least Squares Algorithms

Initially, we are interested in estimating the speed (v) and direction (θ) at which a curtain of light is moving. The curtain moves at a constant speed and direction during the time of motion. We wish to make these estimates solely by using the times at which each cell in the ensemble fires action potentials. To do this, we model each cell's location as a point in the plane representing the cell's receptive field (RF) center. We imagine these cells as sensors which respond instantaneously to changes in brightness. ON cells react to dark-then-bright changes, OFF cells react to bright-then-dark changes and ON-OFF cells react to both types of brightness changes.

Given N such cells in the plane, we number them 1 through N and obtain noisy measurements of each cell's RF center location. We denote cell i's RF center location by (x_i, y_i). We also obtain noisy measurements of the time at which each cell fired relative to the beginning of the recording interval and denote cell i's firing time by t_i[1].

In what follows, we present mathematical relationships between the parameters we obtain from neural recordings (cell locations and firing times) and the speed and direction of the moving edge. Subsequently, we study how the noise in each parameter

[1]A real cell generally fires multiple action potentials when an edge of light passes over its receptive field. However, for simplicity of analysis, we model the cell as a sensor that fires at a single point in time (when the edge is crossing over it's RF center).

affects our beliefs about what the speed and direction are. We do this by restricting ourselves to a specific cell location set-up which is analytically tractable. Next, we discuss possibilities of how to make the desired estimates by merging the information that each cell contributes.

We find that when we observe the response of cells to motion of a thin bar (the thickness of which is on the order of a cell's receptive field diameter), estimates of speed and direction become much noisier. Due to this reason, we look for information coming from DS cell firing, as DS cells have strong opinions about the direction in which a thin bar is moving. We study the performance of algorithms that estimate speed and direction of a thin moving bar under two scenarios: 1) The cells used are all non-DS, 2) The cells used are a mix of DS and non-DS.

4.1 Equations Relating v, θ, (x_i, y_i)'s, and t_i's

4.1.1 Extracting Information by Pairing Cells

One option is to make estimates of v and θ based on pairwise information. To do this, we draw a vector that points from cell i to cell j if $t_i < t_j$. We do this for all $\binom{n}{2}$ cell pairings. We number the cell pairs using an index $k = \{i, j\}$. We denote the magnitude of such a vector by d_k and the angle by θ_k. In addition, for each cell pair k, we define:

$$\Delta t_k \triangleq |t_i - t_j| \tag{4.1}$$

For clarity, in Figure 4-1, we present a depiction of a bright edge moving to the right over a dark background. In this picture, we draw the edge velocity vector (defined to be orthogonal to the line defined by the moving edge, pointing in the direction of motion) in both polar and rectangular coordinates. In polar coordinates, the vector is denoted (v, θ), in rectangular coordinates, (u, w).

Now, given perfect measurements, we have that for cell pair k:

$$v = \frac{d_k}{\Delta t_k} \cos(\theta - \theta_k) \tag{4.2}$$

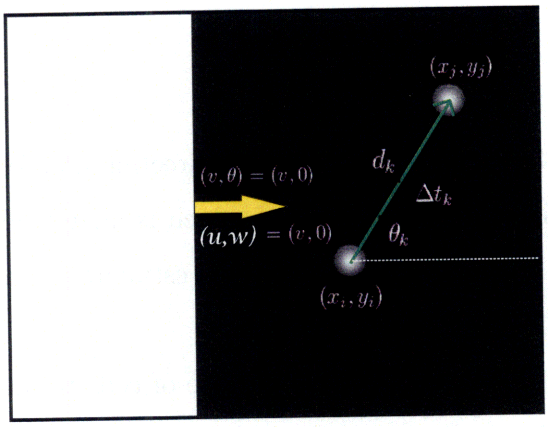

Figure 4-1: Depiction of cell pair, vector from cell i to cell j, time between cell i and cell j, distance between cell i and cell j, and edge velocity vector.

If the measurements of d_k, θ_k and Δt_k were exact, we could find v and θ exactly using only three cells by forming two distinct pairs, which give us two equations. Note that if the measurements are noisy and we wish to have two equations that involve v and θ, where the errors in the measurements in one equation are independent from the errors in the other equation, we require 4 cells. In general, for this to be the case (independence in measurement errors between equations), given N cells, we can only form at most $\lfloor \frac{N}{2} \rfloor$ pairs, though there are many ways to do so.

Alternatively, we can rewrite 4.2 in a form which relates the velocity vector $\underline{v} = (u, w)$ to the measured parameters. Let cells i and j form cell pair k, and let $\mathbf{p}_k$ be the vector which points from cell i to cell j. That is, $\mathbf{p}_k = (x_j, y_j) - (x_i, y_i)$. Then we have that:

$$\mathbf{p}_k \cdot \frac{(u, w)}{\sqrt{u^2 + w^2}} = \Delta t_k \sqrt{u^2 + w^2} \tag{4.3}$$

We see this because $\mathbf{p}_k \cdot \frac{(u,w)}{\sqrt{u^2+w^2}}$ is the distance which the curtain must traverse between cell i and cell j, where $\cdot$ represents the dot product operation.

4.1.2 Extracting Information by Looking at Ensemble Response

We now shift our point of view and wish to extract information about the velocity vector from the response of the cell ensemble as a whole. To do this, we wish to find an equation which relates the velocity vector to each cell's measured parameters. It is not enough to pay attention to one of these equations alone when solving for $\mathbf{v}$, however by using these equations jointly we will be able to find an estimate of $\mathbf{v}$. For the moment, we only present the equation, and not the estimation problem. The ideas and equations which follow in this subsection were presented to us by Prof. Berthold Horn.

The equation for the set of points (x, y) of a line orthogonal to a vector (u, w) at a distance ρ (positive in the direction in which (u, w) points) from the origin is:

$$(x, y) \cdot \frac{(u, w)}{\sqrt{u^2 + w^2}} = \rho \tag{4.4}$$

Further, the perpendicular distance d from an arbitrary point (x', y') in the plane to the line above is:

$$d = (x', y') \cdot \frac{(u, w)}{\sqrt{u^2 + w^2}} - \rho \tag{4.5}$$

where $d > 0$ if (x', y') is displaced from the line by a positive multiple of (u, w)

Now, if the edge moving with velocity (u, w) perpendicular to the edge crosses the origin at time T, then at time t the distance of the line from the origin is $\rho = v(t - T)$, where $v = \sqrt{u^2 + w^2}$. The perpendicular distance $d(t)$ from a point (x', y') in the plane to the nearest point on the moving edge at time t is:

$$d(t) = (x', y') \cdot \frac{(u, w)}{\sqrt{u^2 + w^2}} - \sqrt{u^2 + w^2}(t - T) \tag{4.6}$$

where $d(t) > 0$ until the edge crosses (x', y') and negative thereafter.

If we let (x_i, y_i) be particular points (e.g., receptive field center locations), and t_i be the estimated crossing time, when the edge crosses (x_i, y_i), then (absent measurement

errors) $d(t_i) = 0$.

4.2 Variance in v and θ Estimates

We now focus on Equation 4.2 to understand how v and θ change as we vary the measured parameters from their true values. In other words, we wish to understand how v and θ vary from their true values as a function of variations in the number of cell pairs used, the spatial extent in which the cells are located, the amount of noise in the cell position measurements, and the amount of noise in the firing time measurements.

The following set-up, suggested by Prof. Wyatt, is a bit artificial, but it captures a lot of the qualities we wish to understand. We assume that we have $2N$ cells uniformly placed on a circumference of radius R. N pairs of cells are formed by pairing each cell up with the cell exactly opposite to it on the circumference. We wish to find approximately how the squared error in v and θ vary with these two parameters. Figure 4-2 depicts our set-up.

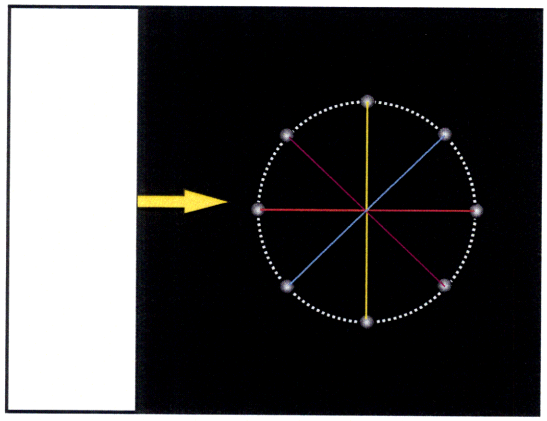

Figure 4-2: $2N$ Cells uniformly placed on a circumference (Here $N = 4$). Each cell is paired up with the cell which is a diameter across from it.

From 4.2 we have that:

$$d_k \cos(\theta_k - \theta) = \Delta t_k v \tag{4.7}$$

for perfect measurements d_k, θ_k, Δt_k and a curtain moving at a speed v and angle θ. Again, the angle of motion is defined to be the angle of the vector pointing in the direction of motion.

We define the residual f_k to be:

$$f_k \triangleq d_k \cos(\theta_k - \theta) - \Delta t_k v \tag{4.8}$$

We note that $f_k = 0$ for perfect measurements of d_k, θ_k, and Δt_k. We perturb the measured parameters $d_k, \theta_k, \Delta t_k$ to incorporate small parameter errors δ_{d_k} in $d_k^{(noisy)} = d_k^{(true)} + \delta_{d_k}$, δ_{θ_k} in $\theta_k^{(noisy)} = \theta_k^{(true)} + \delta_{\theta_k}$, and $\delta_{\Delta t_k}$ in $\Delta t_k^{(noisy)} = \Delta t_k^{(true)} + \delta_{\Delta t_k}$, respectively. From now on we do not use the (noisy) label, and assume that we refer to noisy parameters. For *small perturbations* (i.e., $\frac{\delta_{\theta_k}}{\theta_k} \ll 1$) in these parameters, we obtain the following linear approximation:

$$f_k^{(noisy)} \approx f_k^{(true)} + \frac{Df_k}{D\boldsymbol{\lambda}} \tag{4.9}$$

where $\frac{Df_k}{D\boldsymbol{\lambda}}$ represents taking the first derivative of f_k defined in 4.8 (with respect to all parameters, organized in a vector $\boldsymbol{\lambda}$) evaluated at the true values of the parameters. Since $f_k^{(true)} = 0$ we see that:

$$f_k^{(noisy)} \approx \frac{Df_k}{D\boldsymbol{\lambda}} \tag{4.10}$$

Our objective is to choose v and θ to minimize the sum of squares of the f_k's. In the following text we perform this linearization to show how v and θ change as we vary the measured parameters from their true values. Linearizing 4.7 gives:

$$\cos(\theta_k - \theta)\delta_{d_k} - d_k \sin(\theta_k - \theta)[\delta_{\theta_k} - \delta_\theta] = \Delta t_k \delta_v + v\delta_{\Delta t_k} \tag{4.11}$$

where θ_k, θ, d_k, v, and Δt_k are the real (noise-free) values of those parameters, and the δ's are small deviations in the corresponding parameters.

We reorganize this equation to obtain:

$$d_k \sin(\theta_k - \theta)\delta_\theta - \Delta t_k \delta_v = v\delta_{\Delta t_k} + d_k \sin(\theta_k - \theta)\delta_{\theta_k} - \cos(\theta_k - \theta)\delta_{d_k} \tag{4.12}$$

As we have N cell pairs, we wish to set up a system of equations:

$$\begin{bmatrix} -\Delta t_1 & d_1 \sin(\theta_1 - \theta) \\ \vdots & \vdots \\ -\Delta t_N & d_N \sin(\theta_N - \theta) \end{bmatrix} \begin{bmatrix} \delta_v \\ \delta_\theta \end{bmatrix} = \boldsymbol{\beta} \cdot \begin{bmatrix} \delta_{\Delta t_1} \\ \delta_{\theta_1} \\ \delta_{d_1} \\ \vdots \\ \delta_{\Delta t_N} \\ \delta_{\theta_N} \\ \delta_{d_N} \end{bmatrix} \tag{4.13}$$

where $\boldsymbol{\beta}$ is a $N \times 3N$ matrix composed of N rows. The ith row of β is composed of $3 \cdot (i-1)$ zeros followed by the row vector $\begin{bmatrix} v & d_i \sin(\theta_i - \theta) & -\cos(\theta_i - \theta) \end{bmatrix}$, then followed by $3 \cdot (N - i)$ zeros.

Since N is presumed to be greater than 2, this system is overdetermined. We can find the Least Squares solution for $\begin{bmatrix} \delta_v \\ \delta_\theta \end{bmatrix}$ to be:

$$\begin{bmatrix} \delta_v \\ \delta_\theta \end{bmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \tag{4.14}$$

where $\mathbf{A} = \begin{bmatrix} -\Delta t_1 & d_1 \sin(\theta_1 - \theta) \\ \vdots & \vdots \\ -\Delta t_N & d_N \sin(\theta_N - \theta) \end{bmatrix}$, and $\mathbf{b} = \boldsymbol{\beta} \cdot \begin{bmatrix} \delta_{t_1} \\ \delta_{\theta_1} \\ \delta_{d_1} \\ \vdots \\ \delta_{t_N} \\ \delta_{\theta_N} \\ \delta_{d_N} \end{bmatrix}$

Now,

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \sum_{i=1}^{N} \Delta t_i^2 & -\sum_{i=1}^{N} \Delta t_i d_i \sin(\theta_i - \theta) \\ -\sum_{i=1}^{N} \Delta t_i d_i \sin(\theta_i - \theta) & \sum_{i=1}^{N} d_i^2 \sin^2(\theta_i - \theta) \end{bmatrix} \tag{4.15}$$

We can write Δt_i as $\frac{d_i \cos(\theta_i - \theta)}{v}$ and observe that $d_i = 2R, \forall i$. We then rewrite $\mathbf{A}^T\mathbf{A}$ as:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \frac{4R^2}{v^2} \sum_{i=1}^{N} \cos^2(\theta_i - \theta) & -\frac{2R^2}{v} \sum_{i=1}^{N} \sin(2\theta_i - 2\theta) \\ -\frac{2R^2}{v} \sum_{i=1}^{N} \sin(2\theta_i - 2\theta) & 4R^2 \sum_{i=1}^{N} \sin^2(\theta_i - \theta) \end{bmatrix} \tag{4.16}$$

We now write $\mathbf{b}$ as $\mathbf{b} = \boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}}$, and then compute $\mathbf{A}^T\boldsymbol{\beta}$ below:

$$\underbrace{\mathbf{A}^T\boldsymbol{\beta}}_{2\text{x}3N} = \begin{bmatrix} -\Delta t_1 v & -\Delta t_1 d_1 \sin(\theta_1 - \theta) & \Delta t_1 \cos(\theta_1 - \theta) & \dots & -\Delta t_N v & -\Delta t_N d_N \sin(\theta_N - \theta) & \Delta t_N \cos(\theta_N - \theta) \\ d_1 v \sin(\theta_1 - \theta) & d_1^2 \sin^2(\theta_1 - \theta) & -\frac{d_1}{2} \sin(2\theta_1 - 2\theta) & \dots & d_N v \sin(\theta_N - \theta) & d_N^2 \sin^2(\theta_N - \theta) & -\frac{d_N}{2} \sin(2\theta_N - 2\theta) \end{bmatrix} \tag{4.17}$$

We now compute $\mathbf{A}^T\boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}}$ to be:

$$\underbrace{\mathbf{A}^T\boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}}}_{2\text{X}1} = \begin{bmatrix} -v \sum_{i=1}^{N} \Delta t_i \delta_{\Delta t_i} - \sum_{i=1}^{N} \Delta t_i d_i \sin(\theta_i - \theta)\delta_{\theta_i} + \sum_{i=1}^{N} \Delta t_i \cos(\theta_i - \theta)\delta_{d_i} \\ v \sum_{i=1}^{N} d_i \sin(\theta_i - \theta)\delta_{\Delta t_i} + \sum_{i=1}^{N} d_i^2 \sin^2(\theta_i - \theta)\delta_{\theta_i} - \sum_{i=1}^{N} \frac{d_i}{2} \sin(2\theta_i - 2\theta)\delta_{d_i} \end{bmatrix} \tag{4.18}$$

By replacing Δt_i with $\frac{d_i \cos(\theta_i - \theta)}{v}$ and each d_i with $d_i = 2R$ once again, we then rewrite $\mathbf{A}^T\boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}}$ as:

$$\mathbf{A}^T\boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}} = \begin{bmatrix} -2R \sum_{i=1}^{N} \cos(\theta_i - \theta)\delta_{\Delta t_i} - \frac{2R^2}{v} \sum_{i=1}^{N} \sin(2\theta_i - 2\theta)\delta_{\theta_i} + \frac{2R}{v} \sum_{i=1}^{N} \cos^2(\theta_i - \theta)\delta_{d_i} \\ 2Rv \sum_{i=1}^{N} \sin(\theta_i - \theta)\delta_{\Delta t_i} + 4R^2 \sum_{i=1}^{N} \sin^2(\theta_i - \theta)\delta_{\theta_i} - R \sum_{i=1}^{N} \sin(2\theta_i - 2\theta)\delta_{d_i} \end{bmatrix} \tag{4.19}$$

Now, we would like to rewrite $\mathbf{A}^T\mathbf{A}$ by evaluating the sums it contains. However, it is unclear what each θ_i should be. By assumption, the line segments that connect the cells that form a pair cut the circle into equal pieces. Nonetheless, θ_i depends on which cell in pair i the curtain hits first. Because the direction in which the curtain is moving is a variable, it is not clear what value to assign to each θ_i. It is worth noting that each θ_i can take one of two possible values, each of which are 180° away from each other. It turns out that for all the sums that we will evaluate, it does not matter which of those two values θ_i takes. Therefore, we will let $\theta_i = \frac{2\pi}{N}i$ from now on.

$$\sum_{i=1}^{N} \sin(2\theta_i - 2\theta) = \frac{1}{2j}\sum_{i=1}^{N} e^{j2(\theta_i-\theta)} - \frac{1}{2j}\sum_{i=1}^{N} e^{-j2(\theta_i-\theta)}$$

Now,

$$\sum_{i=1}^{N} e^{j2\theta_i} = \sum_{i=1}^{N} e^{j2\cdot\frac{2\pi}{N}i} = \sum_{i=1}^{N} e^{j\frac{4\pi}{N}i} = \frac{1 - e^{j\frac{4\pi}{N}(N+1)}}{1 - e^{j\frac{4\pi}{N}}} - 1 = 0$$

$$\implies \sum_{i=1}^{N} e^{-j2\theta_i} = 0 \quad \implies \boxed{\sum_{i=1}^{N} \sin(2\theta_i - 2\theta) = 0}$$

It is easy to see that

$$\boxed{\sum_{i=1}^{N} \cos(2\theta_i - 2\theta) = 0}$$

as well, by following very similar steps.

Let's also evaluate:

$$\boxed{\sum_{i=1}^{N} \sin^2(\theta_i - \theta) = \sum_{i=1}^{N} \frac{1 - \cos(2\theta_i - 2\theta)}{2} = \frac{N}{2} - \frac{1}{2}\sum_{i=1}^{N} \cos(2\theta_i - 2\theta) = \frac{N}{2}}$$

Similarly, we see that

$$\boxed{\sum_{i=1}^{N} \cos^2(\theta_i - \theta) = \sum_{i=1}^{N} \frac{1 + \cos(2\theta_i - 2\theta)}{2} = \frac{N}{2}}$$

Now we can write:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \frac{2NR^2}{v^2} & 0 \\ 0 & 2NR^2 \end{bmatrix} \Longrightarrow \boxed{(A^TA)^{-1} = \begin{bmatrix} \frac{v^2}{2NR^2} & 0 \\ 0 & \frac{1}{2NR^2} \end{bmatrix}} \tag{4.20}$$

We now find that:

$$\begin{bmatrix} \delta_v \\ \delta_\theta \end{bmatrix} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\boldsymbol{\beta}\underline{\boldsymbol{\delta\lambda}} \tag{4.21}$$

$$= \begin{bmatrix} \frac{v^2}{2NR^2} & 0 \\ 0 & \frac{1}{2NR^2} \end{bmatrix} \begin{bmatrix} -2R\sum_{i=1}^{N}\cos(\theta_i-\theta)\delta_{\Delta t_i} - \frac{2R^2}{v}\sum_{i=1}^{N}\sin(2\theta_i-2\theta)\delta_{\theta_i} + \frac{2R}{v}\sum_{i=1}^{N}\cos^2(\theta_i-\theta)\delta_{d_i} \\ 2Rv\sum_{i=1}^{N}\sin(\theta_i-\theta)\delta_{\Delta t_i} + 4R^2\sum_{i=1}^{N}\sin^2(\theta_i-\theta)\delta_{\theta_i} - R\sum_{i=1}^{N}\sin(2\theta_i-2\theta)\delta_{d_i} \end{bmatrix} \tag{4.22}$$

$$= \frac{1}{N}\begin{bmatrix} -\frac{v^2}{R}\sum_{i=1}^{N}\cos(\theta_i-\theta)\delta_{\Delta t_i} - v\sum_{i=1}^{N}\sin(2\theta_i-2\theta)\delta_{\theta_i} + \frac{v}{R}\sum_{i=1}^{N}\cos^2(\theta_i-\theta)\delta_{d_i} \\ \frac{v}{R}\sum_{i=1}^{N}\sin(\theta_i-\theta)\delta_{\Delta t_i} + 2\sum_{i=1}^{N}\sin^2(\theta_i-\theta)\delta_{\theta_i} - \frac{1}{2R}\sum_{i=1}^{N}\sin(2\theta_i-2\theta)\delta_{d_i} \end{bmatrix} \tag{4.23}$$

Before we proceed, let's evaluate the following sums which we will make use of in calculating the variances of δ_v and δ_θ:

$$\boxed{\sum_{i=1}^{N}\sin^2(2\theta_i-2\theta) = \sum_{i=1}^{N}\frac{1-\cos(4\theta_i-4\theta)}{2} = \frac{N}{2} - \frac{1}{2}\sum_{i=1}^{N}\cos(4\theta_i-4\theta) = \frac{N}{2}}$$

where the last equality is established because $\sum_{i=1}^{N}\cos(4\theta_i-4\theta) = 0$ by a similar calculation to the one done to find that $\sum_{i=1}^{N}\cos(2\theta_i-2\theta) = 0$. By a very similar calculation it can be shown that $\sum_{i=1}^{N}\cos^2(2\theta_i-2\theta) = \frac{N}{2}$ as well.

Furthermore,

$$\sum_{i=1}^{N} \cos^4(\theta_i - \theta) = \sum_{i=1}^{N} \left(\frac{1 + \cos(2\theta_i - 2\theta)}{2} \right) \left(\frac{1 + \cos(2\theta_i - 2\theta)}{2} \right)$$

$$= \boxed{\sum_{i=1}^{N} \left(\frac{1}{4} + \frac{1}{2}\cos(2\theta_i - 2\theta) + \frac{1}{4}\cos^2(2\theta_i - 2\theta) \right) = \frac{3N}{8}}$$

And by a very similar argument it can be shown that $\sum_{i=1}^{N} \sin^4(\theta_i - \theta) = \frac{3N}{8}$ as well.

Now we wish to find the variance of δ_v and δ_θ. To do this, we must first look at the variances in the measured parameters. We assume that the noise in each coordinate of each cell position is additive and zero-mean with variance σ_p^2. Furthermore, the noise in each coordinate of a particular cell is independent of the noise in the other coordinate, and independent of the noise in each of the coordinates of the other cell in the pair. We also assume that the noise in Δt_k is additive, zero-mean and has variance $\sigma_{\Delta t}^2$. The noise in each Δt_k, denoted $\delta_{\Delta t_k}$, is independent of the noise in the position measurements of all the cells.

Before we proceed, let us find an approximation of the variance of the noise in θ_k, δ_{θ_k}, as a function of σ_p^2. We have that

$$\theta_k = \tan^{-1}\left(\frac{\Delta y}{\Delta x} \right) \triangleq g_\theta \tag{4.24}$$

where Δy is the measured vertical distance separating the cells in pair k, and Δx is the horizontal distance.

Then, to first order, the error in θ_k, δ_{θ_k}, is

$$\delta_{\theta_k} \approx \left. \frac{\partial g_\theta}{\partial \Delta y} \right|_{\Delta y, \Delta x} \delta_{\Delta y} + \left. \frac{\partial g_\theta}{\partial \Delta x} \right|_{\Delta y, \Delta x} \delta_{\Delta x} \tag{4.25}$$

for small errors $\delta_{\Delta y}$ in Δy and $\delta_{\Delta x}$ in Δx.

So we have:

$$\delta_{\theta_k} \approx \frac{\Delta x \delta_{\Delta y} - \Delta y \delta_{\Delta x}}{\Delta x^2 + \Delta y^2} \tag{4.26}$$

Now, since $\delta_{\Delta y}$ and $\delta_{\Delta x}$ are independent by assumption, we find the variance in δ_{θ_k} to be:

$$\sigma^2_{\theta_k} \approx \frac{2\sigma_p^2 \Delta x^2 + 2\sigma_p^2 \Delta y^2}{(\Delta x^2 + \Delta y^2)^2} \tag{4.27}$$

4.27 follows from 4.26 because the variances $\delta_{\Delta y}$ and $\delta_{\Delta x}$ are each twice the variance of the noise in each cell's position coordinates[2]. We see that 4.27 reduces to:

$$\boxed{\sigma^2_{\theta_k} \approx \frac{2\sigma_p^2}{\Delta x^2 + \Delta y^2} = \frac{2\sigma_p^2}{d_k^2}} \tag{4.28}$$

Now, let us also find the variance of δ_{d_k} as a function of σ_p^2. We know that

$$d_k = \sqrt{\Delta x^2 + \Delta y^2} \tag{4.29}$$

By proceeding as we did in equation 4.25, to first order:

$$\delta_{d_k} \approx \frac{1}{2}(\Delta x^2 + \Delta y^2)^{-1/2}[2\Delta x \delta_{\Delta x} + 2\Delta y \delta_{\Delta y}] = \frac{\Delta x \delta_{\Delta x} + \Delta y \delta_{\Delta y}}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{4.30}$$

By the independence of $\delta_{\Delta y}$ and $\delta_{\Delta x}$ we see that:

$$\boxed{\sigma^2_{d_k} \approx \frac{\Delta x^2}{d_k^2} \cdot 2\sigma_p^2 + \frac{\Delta y^2}{d_k^2} \cdot 2\sigma_p^2 = 2\sigma_p^2 \cdot \frac{\Delta x^2 + \Delta y^2}{d_k^2} = 2\sigma_p^2} \tag{4.31}$$

We now find it necessary to calculate any possible non-zero covariance that could exist between δ_{d_k} and δ_{θ_k}. We will use this in computing $Var(\delta_v)$ and $Var(\delta_\theta)$.

[2]This is because $\delta_{\Delta x}$ and $\delta_{\Delta y}$ are each the subtraction of two position coordinates which have noise of variance σ_p^2 and are independent.

$$Cov(\delta_{d_k}, \delta_{\theta_k}) = E[\delta_{d_k}\delta_{\theta_k}] \qquad \text{since } \delta_{d_k} \text{ and } \delta_{\theta_k} \text{ are zero-mean} \tag{4.32}$$

$$\approx E\left[\left(\frac{\Delta x\delta_{\Delta x} + \Delta y\delta_{\Delta y}}{d_k}\right)\left(\frac{\Delta x\delta_{\Delta y} - \Delta y\delta_{\Delta x}}{\Delta x^2 + \Delta y^2}\right)\right] \tag{4.33}$$

$$= E\left[\frac{(\Delta x^2 - \Delta y^2)\delta_{\Delta x}\delta_{\Delta y} + \Delta x\Delta y(\delta_{\Delta y}^2 - \delta_{\Delta x}^2)}{d_k^3}\right] = 0 \tag{4.34}$$

where the last *approximate equality* follows from equations 4.26 and 4.30. In the last equality, the $\delta_{\Delta x}$ and $\delta_{\Delta y}$ cross term disappears by independence and because each of them is zero-mean. The last equality holds because $E[\delta_{\Delta x}^2] = E[\delta_{\Delta y}^2]$.

δ_{θ_k} and δ_{d_k} are therefore, uncorrelated.

Now we have that the $\delta_{\Delta t_i}$'s, δ_{θ_i}'s, δ_{d_i}'s are all uncorrelated with each other. In addition, all $\delta_{\Delta t_i}$'s have a common variance $\sigma_{\Delta t}^2$, δ_{θ_i}'s have a common variance $\sigma_{\theta_k}^2$, and δ_{d_k}'s have a common variance σ_d^2. Now, from 4.21,

$$Var(\delta_v) = \sigma_v^2 \approx \frac{1}{N^2}\left(\frac{v^4}{R^2}\sum_{i=1}^{N}\cos^2(\theta_i - \theta)\sigma_{\Delta t}^2 + v^2\sum_{i=1}^{N}\sin^2(2\theta_i - 2\theta)\sigma_{\theta_k}^2 + \frac{v^2}{R^2}\sum_{i=1}^{N}\cos^4(\theta_i - \theta)\sigma_d^2\right) \tag{4.35}$$

$$= \frac{v^2}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{1}{2}\sigma_{\theta_k}^2 + \frac{3}{8R^2}\sigma_d^2\right) = \frac{v^2}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{\sigma_p^2}{4R^2} + \frac{3}{4R^2}\sigma_p^2\right) \tag{4.36}$$

$$= \boxed{\frac{v^2}{2NR^2}\left(v^2\sigma_{\Delta t}^2 + 2\sigma_p^2\right)} \tag{4.37}$$

where σ_p^2 is the variance in the x and y coordinates of each cell's position. The penultimate equality follows from 4.27.

In addition, from 4.21 we see that,

$$Var(\delta_\theta) = \sigma_\theta^2 \approx \frac{1}{N^2}\left(\frac{v^2}{R^2}\sum_{i=1}^{N}\sin^2(\theta_i-\theta)\sigma_{\Delta t}^2 + 4\sum_{i=1}^{N} sin^4(\theta_i-\theta)\sigma_{\theta_k}^2 + \frac{1}{4R^2}\sum_{i=1}^{N} sin^2(2\theta_i-2\theta)\sigma_d^2\right) \tag{4.38}$$

$$= \frac{1}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{3}{2}\sigma_{\theta_k}^2 + \frac{1}{8R^2}\sigma_d^2\right) \tag{4.39}$$

$$= \frac{1}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{3\sigma_p^2}{4R^2} + \frac{1}{4R^2}\sigma_p^2\right) \tag{4.40}$$

$$= \boxed{\frac{1}{2NR^2}\left(v^2\sigma_{\Delta t}^2 + 2\sigma_p^2\right)} \tag{4.41}$$

4.3 Variances of the Residuals

We are interested in finding the variance of

$$f_k \triangleq d_k\cos(\theta_k-\theta) - \Delta t_k v \tag{4.42}$$

as a function of the variances of the noise in the position measurements of the cells that form pair k and the noise in the measurement of the time between the moment the curtain hits the first cell and the moment it hits the other cell in the pair. Presumably, finding these variances will be useful towards assigning each equation an optimal weight when estimating v and θ.

We find that, to first order, the change in f_k, denoted δ_{f_k}, due to noise in our measured parameters is

$$\delta_{f_k} \approx \cos(\theta_k-\theta)\delta_{d_k} - d_k\sin(\theta_k-\theta)\delta_{\theta_k} - v\delta_{\Delta t_k} \tag{4.43}$$

Let us first note that, to first order, δ_{f_k} is zero-mean, just as δ_{d_k}, δ_{θ_k}, and $\delta_{\Delta t_k}$ are zero-mean. If we are to find the variance in δ_{f_k} as a function of v and θ, the parameters that we are trying to estimate, how can we plug in for v and θ to find the variances? To do so, we assume that by first weighing the equations equally, we are

able to find estimates which are close enough to the true values. Then, we can plug these estimates back in to obtain the approximate variance of each residual.

Since δ_{θ_k} and δ_{d_k} are functions of the noise in each cell position coordinate, each of which is independent of $\delta_{\Delta t_k}$, to find the variance of δ_{f_k} we must consider a possible non-zero covariance between δ_{d_k} and δ_{θ_k}. We have shown in the previous section that this covariance is zero.

Then, from 4.43, the variance $\sigma_{f_k}^2$ becomes:

$$\boxed{\sigma_{f_k}^2 \approx \cos^2(\theta_k - \theta) \cdot 2\sigma_p^2 + d_k^2 \sin^2(\theta_k - \theta) \cdot \frac{2\sigma_p^2}{d_k^2} + v^2\sigma_{\Delta t}^2 = 2\sigma_p^2 + v^2\sigma_{\Delta t}^2} \tag{4.44}$$

Therefore, we notice that to first order, the variance in δ_{f_k} does not depend on the measured values of d_k, θ_k, or Δt_k.

Note: A similar procedure can be carried out by expanding δ_{f_k} to second order and assuming that the additive noises are Gaussian. Doing so, we obtain:

$$\sigma_{f_k}^2 \approx 2\sigma_p^2 + 2v^2\sigma_{\Delta t}^2 + \frac{\cos^2(\theta_k - \theta) \cdot 8\sigma_p^4}{d_k^2} \tag{4.45}$$

4.4 Estimating Curtain Motion Parameters

In this section we present algorithms for estimating the speed and direction of a moving curtain using ON, OFF, and ON-OFF cells. First, we attempt to estimate the velocity vector directly by using information from two-pairings (an arrangement of cells which provides two equations). Next, we attempt to estimate this vector by using global firing time information. Lastly, we estimate the speed and direction of the moving curtain by pairing cells up and taking advantage of the many equations that arise. For each algorithm we provide a sensitivity analysis, i.e., we analyze how the noise in the measured parameters, speed of the curtain, number of cells, and their radial extent affect the estimates of the velocity vector.

4.4.1 Estimating Velocity Vector Directly

Since we are trying to estimate the velocity vector which describes the motion of a curtain, naturally one would like to estimate this vector directly by obtaining equations that involve the cell position and firing time parameters. As mentioned earlier, two cells do not provide information to estimate these unknowns. However, a two-pairing of cells does.

A two-pairing is a selection of a subset (of cardinality 3 or 4) from the set of all cells, such that if the subset is of size 3, we form two cell pairs within the subset by picking one cell which will be the only cell (out of the 3) which is a member of both pairs. If the subset is of size 4, we form two cell pairs within the subset by pairing each cell in the subset with only one other cell. For completeness, let us count all possible ways to make different two-pairings given a set of N cells.

Given a set of N cells, we can choose a pair of cells in $\binom{N}{2}$ distinct ways, i.e., there are $\binom{N}{2}$ distinct pairs that can be formed. Given these $\binom{N}{2}$ pairs, we make a two-pairing by choosing two out of all the possible pairs[3]. This can be done in $\binom{\binom{N}{2}}{2}$ ways. Therefore, given a set of N cells, there are $T_N \triangleq \binom{\binom{N}{2}}{2}$ possible two-pairings.

From each such two-pairing we get a pair of equations:

$$\mathbf{p}_1 \cdot \frac{(u,w)}{\sqrt{u^2+w^2}} = \Delta t_1 \sqrt{u^2+w^2} \qquad \mathbf{p}_2 \cdot \frac{(u,w)}{\sqrt{u^2+w^2}} = \Delta t_2 \sqrt{u^2+w^2} \tag{4.46}$$

where each $\mathbf{p}_i$ is the vector drawn from the cell which fires first to the cell which fires second, and each Δt_i is the time between the moments when the two cells in the pair fire. Note that each equation arises because the component of $\mathbf{p}_i$ in the direction of (u,w) is equal to the time that it takes the edge to get from one cell to the other,

[3]That is, out of the set of possible pairs chosen from the set of all cells, we choose two elements. The two chosen pairs can have either no elements in common or a single element in common. If they have no element in common, they form a two-pairing of cardinality 4; otherwise, they form a two-pairing of cardinality 3.

multiplied by the speed of the edge. We can rewrite this pair of equations as:

$$\mathbf{p}_1 \cdot \frac{(u,w)}{u^2+w^2} = \Delta t_1 \qquad \mathbf{p}_2 \cdot \frac{(u,w)}{u^2+w^2} = \Delta t_2 \tag{4.47}$$

We now invoke Prof. Horn's reflection trick and let

$$u' = \tfrac{u}{u^2+w^2} \qquad w' = \tfrac{v}{u^2+w^2}$$

that is, we reflect (u,w) into the unit circle (if it is outside of it, otherwise we reflect to the outside) and let (u',w') be the new coordinates. By doing so we obtain two equations which are linear in u' and w' which we express as an easily solvable matrix system:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \cdot \begin{bmatrix} u' \\ w' \end{bmatrix} = \begin{bmatrix} \Delta t_1 \\ \Delta t_2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} u' \\ w' \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}^{-1} \begin{bmatrix} \Delta t_1 \\ \Delta t_2 \end{bmatrix} \tag{4.48}$$

where (a_i, b_i) are the coordinates of $\mathbf{p}_i$. Once we find (u',w') we can transform back to (u,w) by reflecting back to outside of the unit circle (or inside, if we had previously reflected outside).[4] That is,

$$u = \tfrac{u'}{u'^2+w'^2} \qquad w = \tfrac{w'}{u'^2+w'^2}$$

We note that each two-pairing gives us *estimates* of (u',w') because the cell locations and Δt's are noisy. We would like to find an overall estimate by putting all the two-pairing information together. We choose to minimize

$$\sum_{k=1}^{T_N} \left\| \begin{bmatrix} \hat{u}'_k \\ \hat{w}'_k \end{bmatrix} - \begin{bmatrix} \hat{u}' \\ \hat{w}' \end{bmatrix} \right\|^2 = \sum_{k=1}^{T_N} (\hat{u'}_k - \hat{u}')^2 + \sum_{k=1}^{T_N} (\hat{w'}_k - \hat{w}')^2 \tag{4.49}$$

where $\hat{u'}_k$ and $\hat{w'}_k$ are the solution of the estimates of u' and w' from each of the T_N two-pairings, and $\hat{u}'$ and $\hat{w}'$ are the overall estimates we obtain by minimizing the above sum. It is easy to see that this sum is minimized when:

[4]The matrix can not be inverted if the vector (a_1, b_1) is a multiple of (a_2, b_2) (i.e., if the 3 or 4 cells in the two-pairing lie on a line). In this case we don't have enough information to solve for (u,w) using this particular two-pairing.

$$\hat{u'} = \frac{1}{T_N}\sum_{k=1}^{T_N} \hat{u'}_k \qquad \hat{w'} = \frac{1}{T_N}\sum_{k=1}^{T_N} \hat{w'}_k \tag{4.50}$$

This approach seems to minimize a quantity that makes natural sense but seems very prone to be affected by outliers. It seems plausible to reduce the effect of the outliers but still use the information they provide by weighing the estimates differently. The weights could be assigned according to how sensitive the estimates are to noise in the particular positions and firing times of cells in the two-pairing.

Estimates' Noise Sensitivity

We would like to understand how the estimates in u' and w' vary from their true values as a function of the cell positions, the cell firing times, noise in the cell positions, and the noise in the cell firing times. Given the variances of the estimates, we have the option of reweighing the terms that go into the sums in 4.114.

For simplicity, we assume that each two-pairing is made up of 4 cells. Then, by assumption, the noise in the measured parameters of one pair is independent of the noise in the measured parameters of the other pair.

Given a particular two-pairing for which the matrix in 4.48 is invertible, we find u' and w':

$$u' = \frac{1}{a_1 b_2 - a_2 b_1}(b_2 \Delta t_1 - b_1 \Delta t_2) \qquad w' = \frac{1}{a_1 b_2 - a_2 b_1}(-a_2 \Delta t_1 + a_1 \Delta t_2) \tag{4.51}$$

First we would like to find $\delta_{u'}$, the perturbation in u', as a function of small perturbations in a_1, a_2, b_1, b_2, Δt_1, and Δt_2. To first order:

$$\begin{aligned}\delta_{u'} \approx\ & (b_2\Delta t_1 - b_1\Delta t_2)\Big[\frac{-b_2}{(a_1b_2 - a_2b_1)^2}\delta_{a_1} + \frac{b_1}{(a_1b_2 - a_2b_1)^2}\delta_{a_2}\Big] \\ & + \Big[\frac{-\Delta t_2}{a_1b_2 - a_2b_1} + \frac{a_2}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2)\Big]\delta_{b_1} \\ & + \Big[\frac{\Delta t_1}{a_1b_2 - a_2b_1} - \frac{a_1}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2)\Big]\delta_{b_2} \\ & + \frac{1}{a_1b_2 - a_2b_1}(b_2\delta_{\Delta t_1} - b_1\delta_{\Delta t_2}) \end{aligned} \tag{4.52}$$

We notice that to first order, $\delta_{u'}$ is zero-mean. Next we find the variance of $\delta_{u'}$ as a function of the variance of the measurement perturbations. We notice that the noise in a_1, a_2, b_1, b_2, Δt_1, and Δt_2, i.e. the corresponding δ's, are all independent of each other (by assuming that the two-pairing consists of 4 distinct cells). Therefore, we see that the variance of $\delta_{u'}$, $\sigma^2_{u'}$ is

$$\begin{aligned} Var(\delta_{u'}) = \sigma^2_{u'} \approx 2\sigma_p^2\Bigg[& (b_2\Delta t_1 - b_1\Delta t_2)^2\Big(\frac{b_1^2 + b_2^2}{(a_1b_2 - a_2b_1)^4}\Big) \\ & + \Big[\frac{-\Delta t_2}{a_1b_2 - a_2b_1} + \frac{a_2}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2)\Big]^2 \\ & + \Big[\frac{\Delta t_1}{a_1b_2 - a_2b_1} - \frac{a_1}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2)\Big]^2\Bigg] \\ & + \sigma^2_{\Delta t}\Big[\frac{b_1^2 + b_2^2}{(a_1b_2 - a_2b_1)^2}\Big] \end{aligned} \tag{4.53}$$

It is easily seen that the first order approximation of $\sigma^2_{w'}$ will be

$$\sigma^2_{w'} = Var(\delta_{w'}) = Var(\delta_{u'})\Big|_{a_1 \leftrightarrow b_1, a_2 \leftrightarrow b_2} \tag{4.54}$$

We choose to reweigh the terms which appear in the sums of 4.114 by 1 over the variance of each corresponding estimate, and then re-normalize the sums. This procedure gives more significance to estimates which have less variance. So 4.114 becomes:

$$\hat{u'} = K_u \frac{1}{T_N} \sum_{k=1}^{T_N} \frac{\hat{u'}_k}{\sigma^2_{u'_k}} \qquad \hat{w'} = K_w \frac{1}{T_N} \sum_{k=1}^{T_N} \frac{\hat{w'}_k}{\sigma^2_{w'_k}} \tag{4.55}$$

where $K_u = \left(\sum_{k=1}^{T_N} \frac{1}{\sigma^2_{u'_k}} \right)^{-1}$ and $K_w = \left(\sum_{k=1}^{T_N} \frac{1}{\sigma^2_{w'_k}} \right)^{-1}$.

4.4.2 Estimating Velocity Vector Using Global Firing Time Information

We now leave the picture of estimating the velocity vector by pairing up cells. Rather, we look at the errors in the firing times of every cell as an ensemble. More specifically, given the velocity vector, and all cell positions, we can calculate how erroneous the firing times are compared to when each cell should have fired according to each of their positions. In other words, we ignore the fact that the cell positions are noisy as well, and minimize the sum of squared timing errors for each cell. If we assume that all cells have comparable noise in their position and firing times, there is no reason to weigh them differently in this minimization.

The ideas presented in this section follow from Prof. Horn's analysis. We refer back to 4.6 and note that the difference in the time when cell i fired and when it should have fired according to its position is $\frac{d_i}{v}$. This situation is depicted in Figure 4-3.

Now, we wish to minimize:

$$\sum_{i=1}^{N} \left(\frac{d_i}{v} \right)^2 = \sum_{i=1}^{N} \left((x_i, y_i) \cdot \frac{(u, w)}{\sqrt{u^2 + w^2}} - (t_i - T) \right)^2 \tag{4.56}$$

by suitable choice of the unknown parameters u, w, and T (the time at which the edge crosses the origin). At first, it seems like the division by $u^2 + w^2$ forces us into a non-linear least squares problem. We can, however, rewrite the sum of squares of errors in the form

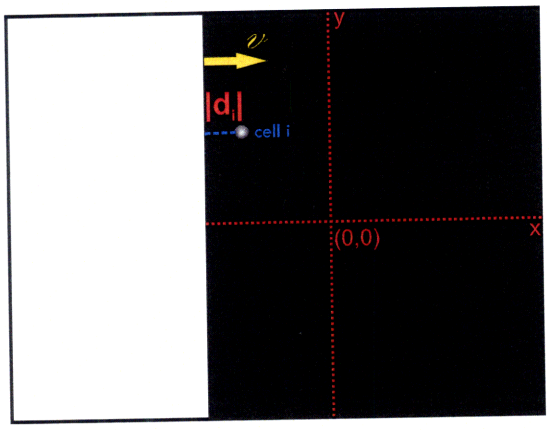

Figure 4-3: This is a picture taken at time t_i (the moment at which cell i fires). It can be seen that the edge is not over cell i's RF center at this moment. The timing error in this picture is then $\frac{d_i}{v}$.

$$\sum_{i=1}^{N} \Big((x_i, y_i) \cdot (u', w') - (t_i - T) \Big)^2 \tag{4.57}$$

where

$$u' = \frac{u}{u^2+w^2} \qquad w' = \frac{w}{u^2+w^2}$$

So we are trying to minimize:

$$\sum_{i=1}^{N} \Big(u'x_i + w'y_i - (t_i - T) \Big)^2 \tag{4.58}$$

by suitable choice of u', w', and T. As in the previous subsection, we can later recover u and w from u' and w' using

$$u = \frac{u'}{u'^2+w'^2} \qquad w = \frac{w'}{u'^2+w'^2}$$

Differentiating the error sum with respect to u', w', and T respectively and setting the results equal to zero leads to:

$$\begin{aligned}
\sum_{i=1}^{N} \Big(u'x_i + w'y_i - (t_i - T)\Big)x_i &= 0 \\
\sum_{i=1}^{N} \Big(u'x_i + w'y_i - (t_i - T)\Big)y_i &= 0 \\
\sum_{i=1}^{N} \Big(u'x_i + w'y_i - (t_i - T)\Big) &= 0
\end{aligned} \tag{4.59}$$

which can be rewritten in the form of three linear equations in three unknown parameters u', w', and T:

$$\begin{aligned}
u' \sum_{i=1}^{N} x_i^2 + w' \sum_{i=1}^{N} x_i y_i + T \sum_{i=1}^{N} x_i &= \sum_{i=1}^{N} t_i x_i \\
u' \sum_{i=1}^{N} x_i y_i + w' \sum_{i=1}^{N} y_i^2 + T \sum_{i=1}^{N} y_i &= \sum_{i=1}^{N} t_i y_i \\
u' \sum_{i=1}^{N} x_i + w' \sum_{i=1}^{N} y_i + T \cdot N &= \sum_{i=1}^{N} t_i
\end{aligned} \tag{4.60}$$

The symmetric 3×3 coefficient matrix depends only on the positions (x_i, y_i) of the cells, while the timing information affects only the right-hand-side vector. Assuming that the coefficient matrix is non-singular[5], we can easily solve for u', w', and T by inverting this matrix and multiplying it by the right-hand-side vector. That is,

[5]The matrix is singular iff the cells lie on a line.

$$\begin{bmatrix} u' \\ w' \\ T \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} y_i^2 & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} t_i x_i \\ \sum_{i=1}^{N} t_i y_i \\ \sum_{i=1}^{N} t_i \end{bmatrix} \tag{4.61}$$

Estimates' Noise Sensitivity

We would like to understand how the noise in the estimates of cell position, and cell firing times affects the estimates of u and w. To do so, we first find the sensitivity in the u' and w' estimates and then proceed from there to find the sensitivity in the u and w estimates. We denote the variance of a cell's measured firing time error δ_{t_k}, $\sigma_{t_k}^2$. It can easily be shown that $\sigma_{\Delta t_k}^2 = 2\sigma_{t_k}^2$.

Once again, to make the sensitivity calculations analytically tractable, we place the N cells on a circumference of radius R. Then, from 4.61, we have that for $\theta_i = \frac{2\pi}{N} i$

$$\begin{aligned} \begin{bmatrix} u' \\ w' \\ T \end{bmatrix} &= \begin{bmatrix} R^2 \sum_{i=1}^{N} \cos^2(\theta_i) & R^2 \sum_{i=1}^{N} \frac{1}{2} \sin(2\theta_i) & R \sum_{i=1}^{N} \cos(\theta_i) \\ R^2 \sum_{i=1}^{N} \frac{1}{2} \sin(2\theta_i) & R^2 \sum_{i=1}^{N} \sin^2(\theta_i) & R \sum_{i=1}^{N} \sin(\theta_i) \\ R \sum_{i=1}^{N} \cos(\theta_i) & R \sum_{i=1}^{N} \sin(\theta_i) & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} t_i x_i \\ \sum_{i=1}^{N} t_i y_i \\ \sum_{i=1}^{N} t_i \end{bmatrix} \\ &= \begin{bmatrix} \frac{R^2 N}{2} & 0 & 0 \\ 0 & \frac{R^2 N}{2} & 0 \\ 0 & 0 & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} t_i x_i \\ \sum_{i=1}^{N} t_i y_i \\ \sum_{i=1}^{N} t_i \end{bmatrix} \end{aligned} \tag{4.62}$$

which gives:

$$u' = \frac{2}{R^2 N}\sum_{i=1}^{N} t_i x_i \qquad w' = \frac{2}{R^2 N}\sum_{i=1}^{N} t_i y_i \qquad T = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{4.63}$$

We first find how the estimate of u' varies as a function of small variations δ_{t_i} and δ_{x_i} from the true values of t_i and x_i, respectively:

$$\delta_{u'} \approx \frac{2}{R^2 N}\Big(\sum_{i=1}^{N} x_i \delta_{t_i} + \sum_{i=1}^{N} t_i \delta_{x_i}\Big) \tag{4.64}$$

As δ_{t_i} and δ_{x_i} are independent by assumption, we have:

$$Var(\delta_{u'}) = \sigma_{u'}^2 \approx \frac{4}{R^4 N^2}\Big(\sum_{i=1}^{N} x_i^2 \sigma_{t_i}^2 + \sum_{i=1}^{N} t_i^2 \sigma_p^2\Big) \tag{4.65}$$

We now plug in $t_i = T + u' x_i + w' y_i$, and $x_i = R\cos(\theta_i)$ into the last equation and get:

$$\begin{aligned}\sigma_{u'}^2 \approx \frac{4}{R^4 N^2}\Big(R^2 \sum_{i=1}^{N} \cos^2(\theta_i)\sigma_{t_i}^2 + \sum_{i=1}^{N}\Big(T^2 + u'^2 R^2 \cos^2(\theta_i) + w'^2 R^2 \sin^2(\theta_i) + \cancelto{0}{2Tu'R\cos(\theta_i)} \\ + \cancelto{0}{2Tw'R\sin(\theta_i)} + \cancelto{0}{2u'w'R^2\frac{1}{2}\sin(2\theta_i)}\Big)\sigma_p^2\Big) \\ = \frac{4}{R^4 N} T^2 \sigma_p^2 + \frac{2}{R^2 N}(\sigma_{t_k}^2 + \frac{1}{v^2}\sigma_p^2)\end{aligned} \tag{4.66}$$

The last equality follows from the fact that $\sum_{i=1}^{N} \cos^2(\theta_i) = \sum_{i=1}^{N} \sin^2(\theta_i) = \frac{N}{2}$, and $\sum_{i=1}^{N} \cos(\theta_i) = \sum_{i=1}^{N} \sin(\theta_i) = \sum_{i=1}^{N} \sin(2\theta_i) = 0$.

In addition, it is easily seen by symmetry that $\sigma_{w'}^2 = \sigma_{u'}^2$.

Before we proceed, for completeness, let us calculate the variance in δ_T, σ_T^2:

$$\delta_T = \frac{1}{N}\sum_{i=1}^{N} \delta_{t_i} \Rightarrow Var(\delta_T) = \sigma_T^2 = \frac{1}{N}\sigma_{t_k}^2 \tag{4.67}$$

We now find the variance in the estimates of u and w as a function of $\sigma_{u'}^2$ and

$\sigma^2_{w'}$. To do so, we express the perturbation in $u = \frac{u'}{u'^2+w'^2}$ as a function of small perturbations in u' and w'. We get, to first order,

$$\begin{aligned}\delta_u &\approx \frac{w'^2 - u'^2}{(u'^2+w'^2)^2}\delta_{u'} + \frac{-2u'w'}{(u'^2+w'^2)^2}\delta_{w'} \\ &= (w^2-u^2)\delta_{u'} - 2uw\delta_{w'} \end{aligned} \tag{4.68}$$

Note that $\delta_w = \delta_u\big|_{u'\leftrightarrow w'}$.

Before we proceed, we need to check for a possible non-zero covariance between $\delta_{u'}$ and $\delta_{w'}$. Since they are both zero-mean, we calculate

$$\begin{aligned}E[\delta_{u'}\delta_{w'}] &= \frac{4}{R^4N^2}E\left[\Big(\sum_{i=1}^N x_i\delta_{t_i} + \sum_{i=1}^N t_i\delta_{x_i}\Big)\Big(\sum_{i=1}^N y_i\delta_{t_i} + \sum_{i=1}^N t_i\delta_{y_i}\Big)\right] \\ &= E\left[\sum_{i=1}^N x_iy_i\delta^2_{t_i}\right] = \sigma^2_{t_k}\sum_{i=1}^N \cancelto{0}{x_iy_i} = 0 \end{aligned} \tag{4.69}$$

where the penultimate equality follows from the fact that the only δ's which have non-zero correlation are δ_{t_i}, δ_{t_j}, for $j = i$.

We are now ready to establish

$$\sigma^2_u \approx (w^2-u^2)^2\sigma^2_{u'} + 4u^2w^2\sigma^2_{w'} \tag{4.70}$$

$$\sigma^2_w \approx (u^2-w^2)^2\sigma^2_{w'} + 4u^2w^2\sigma^2_{u'} \tag{4.71}$$

Variance of Residuals

We are interested in finding the variance of the residuals f_k which enter into the sum we are trying to minimize. If we were to find that the variance of f_k depends on each cell's position and firing time, we could try weighing each f_k in the sum accordingly. However, there is no intuitive reason why the variance should depend on anything else than the curtain speed, the variance in position, and the variance in the firing

times. Therefore, we do not expect that when we minimize this sum any particular cell should be given more significance than any other. Let us verify this. We have that

$$f_k \triangleq u'x_k + w'y_k - (t_k - T) \tag{4.72}$$

Then, for small perturbations δ_{x_k}, δ_{y_k}, and δ_{t_k}, we get an expression for the perturbation in f_k to first order:

$$\delta_{f_k} \approx u'\delta_{x_k} + w'\delta_{y_k} - \delta_{t_k} \tag{4.73}$$

First, we see that δ_{f_k} is zero-mean. We find the variance of δ_{f_k} to be

$$Var(\delta_{f_k}) \approx (u'^2 + w'^2)\sigma_p^2 + \sigma_{t_k}^2 = \left[\left(\frac{u}{u^2+w^2}\right)^2 + \left(\frac{w}{u^2+w^2}\right)^2\right]\sigma_p^2 + \sigma_{t_k}^2 = \frac{1}{v^2}\sigma_p^2 + \sigma_{t_k}^2 \tag{4.74}$$

which does not depend on anything else except the speed of the curtain, the variance in position, and the variance in the firing times; as expected.

4.4.3 Estimating Speed and Direction by Extracting Pairwise Information

In this section, we propose and study two algorithms which make use of pairwise information gathered from the cell ensemble firing times. To commence, we refer back to 4.2 and write

$$v\Delta t_k = d_k \cos(\theta - \theta_k) \tag{4.75}$$

Once again, if the parameters Δt_k, d_k, and θ_k were noise-free, this equation would be solved by the true values of v and θ. However, these parameters come from physical measurements which are subject to noise. In addition, Δt_k, d_k, and θ_k are subject to additional uncertainty because they are each *estimated* to the best of our knowledge

from the data that we gather from experiment.

Given N equations of the form of 4.75, obtained by forming N distinct pairs of cells, we would like to find estimates for v and θ which are very close to their true values. Each of these equations involves a nonlinear function of d_k and θ_k; and therefore, it is not immediately apparent how to set-up the least squares problem. The two following algorithms give a solution to this problem.

CosCos Algorithm

As previously stated, if the measurements of d_k, θ_k, and Δt_k were exact, we could find v and θ exactly using only three cells (looking at two pairs). Since these measurements are noisy, we reformulate 4.75 to estimate v and θ using linear regression. When solving a linear regression, it is assumed that the independent variables (d_k's and θ_k's) are noise-free, but this is not the case. However, we do know that our estimates for them are better than our estimates for Δt_k. Thus, we perform least squares estimates by first noting that $\cos(\theta - \theta_k) = \cos(\theta)\cos(\theta_k) + \sin(\theta)\sin(\theta_k)$ and rewrite 4.75 in the following form:

$$\Delta t_k = \frac{d_k}{v}[\cos(\theta)\cos(\theta_k) + \sin(\theta)\sin(\theta_k)] \tag{4.76}$$

We now rewrite 4.76 in vector form:

$$\Delta t_k = \begin{bmatrix} d_k\cos(\theta_k) & d_k\sin(\theta_k) \end{bmatrix} \cdot \begin{bmatrix} \frac{\cos(\theta)}{v} \\ \frac{\sin(\theta)}{v} \end{bmatrix} \tag{4.77}$$

We notice that this equation is equivalent to a single equation of the matrix system in Equation 4.48. However, in this algorithm, instead of estimating v and θ directly by using every available two-pairing, we propose a solution using a least-squares approach.

Now, our objective is to find the parameters $\alpha = \frac{\cos(\theta)}{v}$ and $\beta = \frac{\sin(\theta)}{v}$ which

minimize the squared error between the observed Δt_k's and the real ones. To carry out the minimization, we have two options: 1) Use all available pairs and form N equations (assuming there are a total of N pairs), 2) Select a subset of pairs which we believe will give us better estimates. If we consider option 2, it is not obvious how to choose these pairs (an active learning approach could be carried through effectively). Assuming that we have chosen N pairs, we organize the Δt_k's corresponding to each pair in a column vector $\mathbf{t}$. We also compose a matrix, $\mathbf{X}$, by making each of its rows a row vector of the form $\begin{bmatrix} d_k \cos(\theta_k) & d_k \sin(\theta_k) \end{bmatrix}$, corresponding to each pair. To show that this linear relationship makes sense, Figure 4-4 depicts a set of points found from the response of cells to downward motion at 714μm/sec. These points have coordinates of the form $(d_k \cos(\theta_k), d_k \sin(\theta_k), \Delta t_k)$.

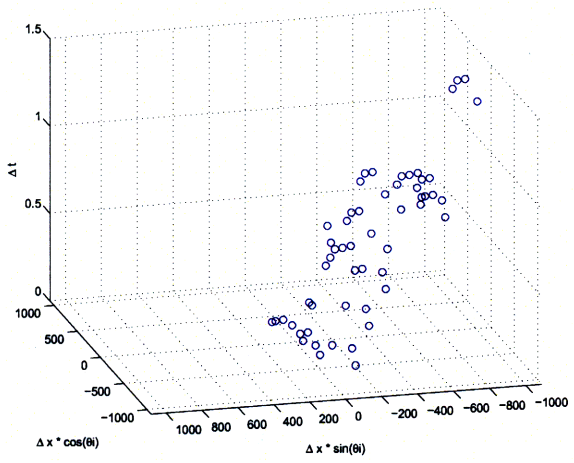

Figure 4-4: Location of points in 3D space. It can be seen that the points approximately lye in a hyperplane, as expected. This would be seen more easily if we were able to rotate the axes.

Our least squares estimates are then given by

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t} \tag{4.78}$$

The estimates of θ and v are then found by enforcing that $\cos^2(\theta) + \sin^2(\theta) = 1$,

which implies that $\alpha^2 + \beta^2 = \frac{1}{v^2}$.[6] Then $v = \sqrt{\frac{1}{\alpha^2+\beta^2}}$, and $\theta = \cos^{-1}(\alpha v)$ or $\theta = \sin^{-1}(\beta v)$. We pick the way to solve for θ based on which of α or β is greater in absolute value. Since the slopes of $\cos^{-1}(x)$ and $\sin^{-1}(x)$ are shallowest at $x = 0$, we use the $\cos^{-1}(x)$ equation if $|\alpha| < |\beta|$ and the $\sin^{-1}(x)$ equation otherwise.

CosCos Sensitivity Analysis

We are now interested in finding how sensitive the estimates for v and θ are when using this algorithm. In particular, we would like to understand how much the estimates vary as we introduce firing time and position errors.

Using the CosCos algorithm we find v and θ as a function of $\alpha = \frac{\cos(\theta)}{v}$ and $\beta = \frac{\sin(\theta)}{v}$ by:

$$v = \sqrt{\frac{1}{\alpha^2+\beta^2}} \qquad \theta = \begin{cases} \cos^{-1}(\alpha v) & \text{if } |\alpha| < |\beta| \\ \sin^{-1}(\beta v) & \text{else} \end{cases}$$

First, we need to understand how the estimates of v and θ vary from the true values as a function of variations in α and β. For small perturbations δ_α and δ_β in the true values of α and β respectively, we have (by differentiating) that

$$\delta_v \approx -\frac{1}{2}(\alpha^2 + \beta^2)^{-3/2}[2\alpha\delta_\alpha + 2\beta\delta_\beta] \tag{4.79}$$

$$\delta_\theta \approx \begin{cases} -\frac{1}{\sqrt{1-(\alpha v)^2}}[v\delta_\alpha + \alpha\delta_v] & \text{if } |\alpha| < |\beta| \\ \frac{1}{\sqrt{1-(\beta v)^2}}[v\delta_\beta + \beta\delta_v] & \text{else} \end{cases} \tag{4.80}$$

Assuming that δ_v and δ_θ are zero-mean (which we show later), we would like to find $Var(\delta_v)$ and $Var(\delta_\theta)$. To do this, we need to express δ_α and δ_β as a function of small perturbations δ_{d_k}, $\delta_{\Delta t_k}$, and δ_{θ_k} in d_k, Δt_k, and θ_k respectively.

We have that

$$d_k \cos(\theta_k)\alpha + d_k \sin(\theta_k)\beta = \Delta t_k \tag{4.81}$$

[6] If the estimates of α and β are good, this is plausible.

We differentiate both sides of this equation with respect to all variables to get an approximate equation relating small changes in each variable to each other. To first order, we get

$$d_k \cos(\theta_k)\delta_\alpha + d_k \sin(\theta_k)\delta_\beta + [\alpha \cos(\theta_k) + \beta \sin(\theta_k)]\delta_{d_k} + [-d_k\alpha \sin(\theta_k) + d_k\beta \cos(\theta_k)]\delta_{\theta_k} = \delta_{\Delta t_k} \tag{4.82}$$

Given N cell pairs, we set up the matrix system:

$$\mathbf{A} \cdot \begin{bmatrix} \delta_\alpha \\ \delta_\beta \end{bmatrix} = \mathbf{C}\underline{\delta\lambda} \tag{4.83}$$

where

$$\mathbf{A} = \begin{bmatrix} d_1 \cos(\theta_1) & d_1 \sin(\theta_1) \\ \vdots & \vdots \\ d_N \cos(\theta_N) & d_N \sin(\theta_N) \end{bmatrix} \quad \text{and} \quad \underline{\delta\lambda} = \begin{bmatrix} \delta_{d_1} \\ \delta_{\theta_1} \\ \delta_{t_1} \\ \vdots \\ \delta_{d_N} \\ \delta_{\theta_N} \\ \delta_{t_N} \end{bmatrix}$$

$\mathbf{C}$ is a $N \times 3N$ matrix composed of N rows. The ith row of $\mathbf{C}$ is composed of $3 \cdot (i-1)$ zeros followed by the row vector $\begin{bmatrix} -\alpha \cos(\theta_i) - \beta \sin(\theta_i), & d_i\alpha \sin(\theta_i) - d_i\beta \cos(\theta_i), & 1 \end{bmatrix}$, then followed by $3 \cdot (N - i)$ zeros.

Based on the equation above, with a least squares picture in mind, we make the following approximation:

$$\begin{bmatrix} \delta_\alpha \\ \delta_\beta \end{bmatrix} \approx (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C}\underline{\delta\lambda} \tag{4.84}$$

We proceed by computing

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \sum_{i=1}^{N} d_i^2 \cos^2(\theta_i) & \sum_{i=1}^{N} d_i^2 \frac{1}{2}\sin(2\theta_i) \\ \sum_{i=1}^{N} d_i^2 \frac{1}{2}\sin(2\theta_i) & \sum_{i=1}^{N} d_i^2 \sin^2(\theta_i) \end{bmatrix} \tag{4.85}$$

Once again, for simplicity, we assume that all cells are distributed uniformly on a circumference of radius R. Each cell is paired with the cell which is a diameter across from it. We have $2N$ cells, and therefore N cell pairs, as presumed above.

Using the trigonometric equalities established in Section 4.2 we simplify $\mathbf{A}^T\mathbf{A}$ to:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 2R^2N & 0 \\ 0 & 2R^2N \end{bmatrix} \Rightarrow (A^TA)^{-1} = \frac{1}{2R^2N}\mathbf{I}_2 \tag{4.86}$$

where $\mathbf{I}_2$ represents the 2×2 identity matrix.

Now we see that:

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \frac{1}{2R^2N}\begin{bmatrix} d_1\cos(\theta_1) & \dots & d_N\cos(\theta_N) \\ d_1\sin(\theta_1) & \dots & d_N\sin(\theta_N) \end{bmatrix} \tag{4.87}$$

and

$$\underbrace{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C}}_{2\text{x}3N} = \frac{1}{2R^2N}\begin{bmatrix} -d_1(\alpha\cos^2(\theta_1) + \beta\frac{1}{2}\sin(2\theta_1)) & d_1^2(\alpha\frac{1}{2}\sin(2\theta_1) - \beta\cos^2(\theta_1)) & d_1\cos(\theta_1) & \dots \\ -d_1(\alpha\frac{1}{2}\sin(2\theta_1) + \beta\sin^2(\theta_1)) & d_1^2(\alpha\sin^2(\theta_1) - \beta\frac{1}{2}\sin(2\theta_1)) & d_1\sin(\theta_1) & \dots \end{bmatrix} \tag{4.88}$$

Finally, we have that

$$\begin{bmatrix} \delta_\alpha \\ \delta_\beta \end{bmatrix} \approx (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C}\underline{\boldsymbol{\delta\lambda}}$$

$$= \frac{1}{2R^2N}\begin{bmatrix} -\sum_{i=1}^{N} d_i(\alpha\cos^2(\theta_i) + \beta\frac{1}{2}\sin(2\theta_i))\delta_{d_i} + \sum_{i=1}^{N} d_i^2(\alpha\frac{1}{2}\sin(2\theta_i) - \beta\cos^2(\theta_i))\delta_{\theta_i} + \sum_{i=1}^{N} d_i\cos(\theta_i)\delta_{\Delta t_i} \\ -\sum_{i=1}^{N} d_i(\alpha\frac{1}{2}\sin(2\theta_i) + \beta\sin^2(\theta_i))\delta_{d_i} + \sum_{i=1}^{N} d_i^2(\alpha\sin^2(\theta_i) - \beta\frac{1}{2}\sin(2\theta_i))\delta_{\theta_i} + \sum_{i=1}^{N} d_i\sin(\theta_i)\delta_{\Delta t_i} \end{bmatrix} \tag{4.89}$$

Plugging in $d_i = 2R, \forall i$ (since we have assumed that the cells are on a circumference) we have:

$$\begin{bmatrix} \delta_\alpha \\ \delta_\beta \end{bmatrix} \approx \frac{1}{2RN} \begin{bmatrix} -2\sum_{i=1}^{N}(\alpha\cos^2(\theta_i) + \beta\frac{1}{2}\sin(2\theta_i))\delta_{d_i} + 4R\sum_{i=1}^{N}(\alpha\frac{1}{2}\sin(2\theta_i) - \beta\cos^2(\theta_i))\delta_{\theta_i} + 2\sum_{i=1}^{N}\cos(\theta_i)\delta_{\Delta t_i} \\ -2\sum_{i=1}^{N}(\alpha\frac{1}{2}\sin(2\theta_i) + \beta\sin^2(\theta_i))\delta_{d_i} + 4R\sum_{i=1}^{N}(\alpha\sin^2(\theta_i) - \beta\frac{1}{2}\sin(2\theta_i))\delta_{\theta_i} + 2\sum_{i=1}^{N}\sin(\theta_i)\delta_{\Delta t_i} \end{bmatrix} \tag{4.90}$$

First, note that $E[\delta_\alpha] = E[\delta_\beta] = 0$ because $E[\delta_{d_k}] = E[\delta_{\theta_k}] = E[\delta_{\Delta t_k}] = 0$. Now, we use the fact that δ_{d_k}, δ_{θ_k}, and $\delta_{\Delta t_k}$ are uncorrelated for all k and find $Var(\delta_\alpha)$ and $Var(\delta_\beta)$ to be

$$\begin{bmatrix} \sigma_\alpha^2 \\ \sigma_\beta^2 \end{bmatrix} \approx \frac{1}{4R^2N^2} \begin{bmatrix} 4\sum_{i=1}^{N}(\alpha^2\cos^4(\theta_i) + \beta^2\frac{1}{4}\sin^2(2\theta_i))\sigma_{d_i}^2 + 16R^2\sum_{i=1}^{N}(\alpha^2\frac{1}{4}\sin^2(2\theta_i) + \beta^2\cos^4(\theta_i))\sigma_{\theta_i}^2 + 4\sum_{i=1}^{N}\cos^2(\theta_i)\sigma_{\Delta t_i}^2 \\ 4\sum_{i=1}^{N}(\alpha^2\frac{1}{4}\sin^2(2\theta_i) + \beta^2 sin^4(\theta_i))\sigma_{d_i}^2 + 16R^2\sum_{i=1}^{N}(\alpha^2 sin^4(\theta_i) + \beta^2\frac{1}{4}\sin^2(2\theta_i))\sigma_{\theta_i}^2 + 4\sum_{i=1}^{N}\sin^2(\theta_i)\sigma_{\Delta t_i}^2 \end{bmatrix} \tag{4.91}$$

Note that in the formula above, inside the sums, the cross terms that appear when squaring the terms which multiply the σ^2's, disappear. This can be seen using simple calculations similar to the ones done using the trigonometric identities presented in Section 4.2.

Once again, we take advantage of the the cell location set-up and the pairing we have enforced. We use the equalities established in Section 4.2 again to conclude:

$$\begin{bmatrix} \sigma_\alpha^2 \\ \sigma_\beta^2 \end{bmatrix} \approx \frac{1}{4R^2N^2} \begin{bmatrix} (4\alpha^2\frac{3N}{8} + \beta^2\frac{N}{2})\sigma_d^2 + (4R^2\alpha^2\frac{N}{2} + 16R^2\beta^2\frac{3N}{8})\sigma_{\theta_k}^2 + 4\frac{N}{2}\sigma_{\Delta t}^2 \\ (\alpha^2\frac{N}{2} + 4\beta^2\frac{3N}{8})\sigma_d^2 + (16R^2\alpha^2\frac{3N}{8} + 4R^2\beta^2\frac{N}{2})\sigma_{\theta_k}^2 + 4\frac{N}{2}\sigma_{\Delta t}^2 \end{bmatrix} \tag{4.92}$$

$$= \frac{1}{4R^2N} \begin{bmatrix} (\frac{3}{2}\alpha^2 + \frac{1}{2}\beta^2)\sigma_d^2 + R^2(2\alpha^2 + 6\beta^2)\sigma_{\theta_k}^2 + 2\sigma_{\Delta t}^2 \\ (\frac{1}{2}\alpha^2 + \frac{3}{2}\beta^2)\sigma_d^2 + R^2(6\alpha^2 + 2\beta^2)\sigma_{\theta_k}^2 + 2\sigma_{\Delta t}^2 \end{bmatrix} \tag{4.93}$$

Once gain, we replace σ_d^2 with $2\sigma_p^2$, and $\sigma_{\theta_k}^2$ with $\frac{\sigma_p^2}{2R^2}$ to obtain:

$$\begin{bmatrix} \sigma_\alpha^2 \\ \sigma_\beta^2 \end{bmatrix} \approx \frac{1}{R^2 N} \begin{bmatrix} (\alpha^2+\beta^2)\sigma_p^2 + \frac{1}{2}\sigma_{\Delta t}^2 \\ (\alpha^2+\beta^2)\sigma_p^2 + \frac{1}{2}\sigma_{\Delta t}^2 \end{bmatrix} \tag{4.94}$$

$$= \frac{1}{R^2 N} \begin{bmatrix} \frac{1}{v^2}\sigma_p^2 + \frac{1}{2}\sigma_{\Delta t}^2 \\ \frac{1}{v^2}\sigma_p^2 + \frac{1}{2}\sigma_{\Delta t}^2 \end{bmatrix} \tag{4.95}$$

where the last equality follows from the fact that $\alpha^2+\beta^2 = \frac{\cos^2(\theta)}{v^2} + \frac{\sin^2(\theta)}{v^2} = \frac{1}{v^2}$.

Before we finalize the solution of σ_v^2 and σ_θ^2, we need to check for a possible non-zero covariance between δ_α and δ_β. We have that $Cov(\delta_\alpha, \delta_\beta) = E[\delta_\alpha \delta_\beta]$ since δ_α and δ_β are zero-mean. So

$$\begin{aligned} Cov(\delta_\alpha, \delta_\beta) &= E[\delta_\alpha \delta_\beta] \\ &= \frac{1}{4R^2N^2} E\Bigg[4\sum_{i=1}^{N} (\alpha\cos^2(\theta_i) + \frac{\beta}{2}\sin(2\theta_i))(\frac{\alpha}{2}\sin(2\theta_i) + \beta\sin^2(\theta_i))\delta_{d_i}^2 \\ &+ 16R^2 \sum_{i=1}^{N} (\frac{\alpha}{2}\sin(2\theta_i) - \beta\cos^2(\theta_i))(\alpha\sin^2(\theta_i) - \frac{\beta}{2}\sin(2\theta_i))\delta_{\theta_i}^2 + 4\sum_{i=1}^{N}\cos(\theta_i)\sin(\theta_i)\delta_{\Delta t_i}^2 \Bigg] \end{aligned} \tag{4.96}$$

The expectation of all other cross terms (where $i \neq j$ in the resulting double sums) go to zero by independence and because the δ's are zero-mean. Therefore, we don't bother to write them.

By using the trigonometric equalities established in Section 4.2 once again, we see that

$$Cov(\delta_\alpha, \delta_\beta) = \frac{1}{R^2N^2}\left[(\alpha\beta\frac{N}{8} + \frac{\alpha\beta}{4}\frac{N}{2})\sigma_d^2 + 4R^2(-\frac{\alpha\beta}{4}\frac{N}{2} - \alpha\beta\frac{N}{8})\sigma_{\theta_k}^2 \right] \tag{4.97}$$

Now, we replace σ_d^2 with $2\sigma_p^2$, and $\sigma_{\theta_k}^2$ with $\frac{\sigma_p^2}{2R^2}$ to see that $\boxed{Cov(\delta_\alpha, \delta_\beta) = 0}$.

For completeness, let us note that since δ_α and δ_β are zero-mean, it follows that δ_v is zero-mean too. Now, based on the above calculation, we easily establish (using

4.79) that:

$$\boxed{\sigma_v^2 \approx (\alpha^2+\beta^2)^{-3}[\alpha^2\sigma_\alpha^2+\beta^2\sigma_\beta^2] = v^4\sigma_\alpha^2 = \frac{v^2}{R^2N}(\sigma_p^2+\frac{v^2}{2}\sigma_{\Delta t}^2)} \tag{4.98}$$

as $\sigma_\alpha^2 = \sigma_\beta^2$ (shown in 4.95).

To find σ_θ^2, the variance in δ_θ we first need to find $Cov(\delta_\alpha, \delta_v) = E[\delta_\alpha \delta_v]$ and $Cov(\delta_\beta, \delta_v) = E[\delta_\beta \delta_v]$. In the case that $|\alpha| < |\beta|$, we only need to find the former. We only find σ_θ^2 for this case because the calculations for the case when $|\alpha| \geq |\beta|$ are very similar.

$$\begin{aligned} Cov(\delta_\alpha, \delta_v) &= E[-(\alpha^2+\beta^2)^{-3/2}(\alpha\delta_\alpha^2 + \beta\delta_\beta\delta_\alpha)] \\ &= -(\alpha^2+\beta^2)^{-3/2}\alpha E[\delta_\alpha^2] = -(\alpha^2+\beta^2)^{-3/2}\alpha\sigma_\alpha^2 \end{aligned} \tag{4.99}$$

where the penultimate equality is due to the recently established fact that δ_α and δ_β are uncorrelated.

Now, if $|\alpha| < |\beta|$, from 4.80 we have that:

$$\begin{aligned} \sigma_\theta^2 &\approx \frac{1}{1-(\alpha v)^2}[v^2\sigma_\alpha^2 + \alpha^2\sigma_v^2 - 2\alpha^2 v(\alpha^2+\beta^2)^{-3/2}\sigma_\alpha^2] \\ &= \frac{1}{1-(\alpha v)^2}[(v^2 - 2\alpha^2 v(\alpha^2+\beta^2)^{-3/2})\sigma_\alpha^2 + \alpha^2\sigma_v^2] \\ &= \boxed{\frac{1}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)} \end{aligned} \tag{4.100}$$

where the last equality follows by plugging in $\alpha = \frac{\cos(\theta)}{v}$ and $\beta = \frac{\sin(\theta)}{v}$ into the previous one.

By symmetry, we expect σ_θ^2 to be equal to the boxed expression in the case that $|\alpha| \geq |\beta|$ as well.

Newton-Raphson Minimization of Residuals

In this subsection, we present and solve a non-linear minimization problem, which provides us with estimates of v and θ. This approach and the problem solution was suggested by Prof. Wyatt. We wish to minimize the sum-of-squares of the residuals f_k:

$$f(v,\theta) \triangleq \sum_{i=1}^{N} f_i^2 \tag{4.101}$$

for f_k as defined in 4.42, and for N the total number of cell pairs. One would hope that by assigning the residuals different weights which depend on the measured parameters, we would get better estimates than by giving them equal weights. However, as noted in Section 4.3, to first order, the variances in the residuals only depend on the speed of the curtain, the variance in the position estimates, and the variance in the firing times. Therefore, we wish to minimize the uniformly weighted sum-of-squares, shown above.

We can find v^* and θ^*, the optimal v and θ respectively, which minimize 4.101 by finding where

$$\underline{f_d} = \begin{bmatrix} f_d^{(1)} \\ f_d^{(2)} \end{bmatrix} \triangleq \begin{bmatrix} \frac{\partial f(v,\theta)}{\partial v} \\ \frac{\partial f(v,\theta)}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.102}$$

We do this by assuming that we have an estimate $(\hat{v},\hat{\theta})$ which is close enough to v^* and θ^*. We use this estimate as an initial value for a Newton-Raphson algorithmic approach.

The Newton-Raphson algorithm leads to the optimal solution iteratively. The updates are given by

$$\begin{bmatrix} \hat{v}_{n+1} \\ \hat{\theta}_{n+1} \end{bmatrix} = - \begin{bmatrix} \frac{\partial f_d^{(1)}}{\partial v} & \frac{\partial f_d^{(1)}}{\partial \theta} \\ \frac{\partial f_d^{(2)}}{\partial v} & \frac{\partial f_d^{(2)}}{\partial \theta} \end{bmatrix}^{-1}_{\hat{v}_n,\hat{\theta}_n} \underline{f_d}(n) + \begin{bmatrix} \hat{v}_n \\ \hat{\theta}_n \end{bmatrix} \tag{4.103}$$

Newton-Raphson Sensitivity Analysis

We choose the estimates of v and θ to be the arguments which minimize $f(v,\theta)$, defined in 4.101. The location of this minimum (in the $v-\theta$ plane) depends on the values of $3N$ measured parameters (i.e., the vector $(d_1,\theta_1,\Delta t_1,\ldots,d_N,\theta_N,\Delta t_N)$). In other words, the minimum (v_{min}, θ_{min}) is a function of this vector of measured parameters. Precisely, the estimates of v and θ are the values which solve 4.102. By the implicit function theorem, we know that if the Jacobian Matrix $[J\underline{f_d}]$ is invertible, then a solution exists in a neighborhood of the true values of v and θ for small perturbations of the vector of measured parameters. In addition, we can find how v and θ vary as a function of variations in the measured parameter vector in the following manner:

$$\delta_v \approx \sum_{i=1}^{N} \frac{\partial v}{\partial d_i}\delta_{d_i} + \sum_{i=1}^{N} \frac{\partial v}{\partial \theta_i}\delta_{\theta_i} + \sum_{i=1}^{N} \frac{\partial v}{\partial \Delta t_i}\delta_{\Delta t_i} \tag{4.104}$$

$$\delta_\theta \approx \sum_{i=1}^{N} \frac{\partial \theta}{\partial d_i}\delta_{d_i} + \sum_{i=1}^{N} \frac{\partial \theta}{\partial \theta_i}\delta_{\theta_i} + \sum_{i=1}^{N} \frac{\partial \theta}{\partial \Delta t_i}\delta_{\Delta t_i} \tag{4.105}$$

where

$$\begin{bmatrix} \frac{\partial v}{\partial d_1} & \frac{\partial v}{\partial \theta_1} & \frac{\partial v}{\partial \Delta t_1} & \cdots \\ \frac{\partial \theta}{\partial d_1} & \frac{\partial \theta}{\partial \theta_1} & \frac{\partial \theta}{\partial \Delta t_1} & \cdots \end{bmatrix} = -[J\underline{f_d}]^{-1} \begin{bmatrix} \frac{\partial f_d^{(1)}}{\partial d_1} & \frac{\partial f_d^{(1)}}{\partial \theta_1} & \frac{\partial f_d^{(1)}}{\partial \Delta t_1} & \cdots \\ \frac{\partial f_d^{(2)}}{\partial d_1} & \frac{\partial f_d^{(2)}}{\partial \theta_1} & \frac{\partial f_d^{(2)}}{\partial \Delta t_1} & \cdots \end{bmatrix} \tag{4.106}$$

and the matrices on the right hand side are evaluated at the true values of v and θ and the true values of the parameters.

By making the simplification of evenly distributing the cells on a circumference of radius R, we see that the variance of δ_v and δ_θ (found by using the equations above) are exactly equal to what was obtained when finding the variance in v and θ (in Section 4.2), that is,

$$\sigma_v^2 \approx \frac{v^2}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2) \tag{4.107}$$

$$\sigma_\theta^2 \approx \frac{1}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2) \tag{4.108}$$

Therefore, we notice that the speed and direction estimate sensitivities for the CosCos and Newton-Raphson algorithms are equal, to first order, for cells equally distributed on a circumference and paired as we described.

4.5 Estimating Thin Bar Motion Parameters

We assume that for each DS cell we have a function $h_k : \theta \rightarrow \mathbb{R}^+$ which approximates the number of spikes that DS cell k fires for motion of a thin bar (moving exactly over it's RF) in the direction θ. For each h_k we define a corresponding residual g_k:

$$g_k(\theta) \triangleq h_k(\theta) - S_k \tag{4.109}$$

where S_k is the number of spikes that cell k actually fired when a particular bar was moved over its receptive field.

Now, once again, we wish to minimize the sum-of-squares of the residuals. In other words, we wish to minimize

$$g(\theta) \triangleq \sum_{i=1}^{N} g_i^2(\theta) \tag{4.110}$$

where N is the number of DS cells that we use to make the estimates of v and θ.

Given only DS cells, we can use their ON-OFF property alone (ignoring their directional properties) to make the estimates exactly as described in Section 4.4. We also have the option of making an estimate of θ merely by using the directionality property of the DS cells and minimizing 4.110. More interestingly, we can make the estimates by merging the information from non-DS cells with the non-directional and directional information from DS cells. We wish to do this by minimizing a weighted sum-of-squares of residuals of the form

$$q(v,\theta) \triangleq K_g \sum_{i=1}^{N} g_i^2(\theta) + \sum_{i=1}^{M} f_i^2(v,\theta) \tag{4.111}$$

where f_k is defined as in Section 4.4.3, N is the number of DS cells that reacted to the particular bar, and M is the number of cell pairs formed when observing the ON/OFF response to the same bar.

We weigh the residuals differently mainly because of a difference in the units of f_k and g_k. To do so, we must provide a manner of selecting K_g, the weight assigned to the g_k residuals.

4.5.1 Weighing the Residuals of DS and non-DS cells

Our goal is to find a value for K_g such that $K_g \cdot Var(g_k)$ is comparable to $Var(f_k)$. By doing so, we give each term in the minimization coming from the DS directional properties the same significance as the terms coming from non-directional ON/OFF responses of DS and non-DS cells. If, for example, we would like to give DS cells more significance to further refine direction estimates, then K_g would have to be larger than the value we find in this subsection. It should also be noted that if there are many terms in the minimization due to non-DS information (i.e. there are many non-DS cells which fire) then it is also preferable to increase K_g.

We invoke 4.44 which tells us that $Var(f_k) \approx 2\sigma_p^2 + v^2\sigma_{\Delta t}^2$. Judging by the typical amount of noise in the time and positions estimates, it is reasonable to set $\sigma_p = 100\mu$m, and $\sigma_{\Delta t} = 0.1$sec. This gives $Var(f_k) \approx 20000(\mu\text{m})^2 + v^2 0.01\text{sec}^2$. Let's treat v as a random variable which takes a value in the range $300 - 3000\mu$m/sec uniformly[7]. Then $E[Var(f_k)] \approx 20000(\mu\text{m})^2 + 0.01E[v^2]$. We find:

[7]This is a reasonable assumption. We do not expect the cells to respond very well to speeds lower than 300μm/sec, and we do not expect to be able to estimate speeds and directions accurately for speeds greater than $3000\mu m/sec$.

$$E[v^2] = Var(v) + (E[v])^2$$
$$= \frac{2700^2}{12} + (1650)^2 = 33 \cdot 10^5 (\mu\text{m/sec})^2 \quad (4.112)$$

We conclude that $E[Var(f_k)] \approx 53 \cdot 10^3$.

Based on experimental data it is reasonable to assume that $Var(g_k) \approx 53$. This suggests that $K_g = 10^3$.

4.6 Algorithms and Sensitivities Summary

4.6.1 Sensitivities of v and θ as a Function of Noisy Measured Parameters

We base the following sensitivity calculations on Equation 4.2.

Assumptions: 2N cells are equally spaced on circumference. Each cell is paired with the cell which is a diameter across from it; R is radius of circumference.

$$\sigma_v^2 \approx \frac{v^2}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

$$\sigma_\theta^2 \approx \frac{1}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

4.6.2 Variance of residuals $f_k \triangleq d_k \cos(\theta_k - \theta) - \Delta t_k v$:

$$\sigma_{f_k}^2 \approx 2\sigma_p^2 + v^2\sigma_{\Delta t}^2$$

4.6.3 Estimating Velocity Vector Directly in Rectangular Coordinates (Adam's Method)

From each two-pairing of the cells (T_N such two-pairings) we get:

$$\begin{bmatrix} u' \\ w' \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}^{-1} \begin{bmatrix} \Delta t_1 \\ \Delta t_2 \end{bmatrix}$$

where a_i and b_i are $d_i\cos(\theta_i)$ and $d_i\sin(\theta_i)$, respectively.

We choose to minimize

$$\sum_{k=1}^{T_N} \left\| \begin{bmatrix} \hat{u}'_k \\ \hat{w}'_k \end{bmatrix} - \begin{bmatrix} \hat{u}' \\ \hat{w}' \end{bmatrix} \right\|^2 = \sum_{k=1}^{T_N} (\hat{u}'_k - \hat{u}')^2 + \sum_{k=1}^{T_N} (\hat{w}'_k - \hat{w}')^2 \tag{4.113}$$

where $\hat{u}'_k$ and $\hat{w}'_k$ are the solution of the estimates of u' and w' from each of the T_N two-pairings, and $\hat{u}'$ and $\hat{w}'$ are the overall estimates we obtain by minimizing the above sum. Our estimates our then:

$$\hat{u}' = \frac{1}{T_N} \sum_{k=1}^{T_N} \hat{u}'_k \qquad \hat{w}' = \frac{1}{T_N} \sum_{k=1}^{T_N} \hat{w}'_k \tag{4.114}$$

Then we express the u and w estimates for this two-pairing, $\hat{u}$ and $\hat{w}$, as:

$$\hat{u} = \frac{\hat{u}'}{\hat{u}'^2 + \hat{w}'^2}$$

$$\hat{w} = \frac{\hat{w}'}{\hat{u}'^2 + \hat{w}'^2}$$

Sensitivities

$$Var(\delta_{u'}) = \sigma_{u'}^2 \approx 2\sigma_p^2 \Bigg[(b_2\Delta t_1 - b_1\Delta t_2)^2 \Big(\frac{b_1^2 + b_2^2}{(a_1b_2 - a_2b_1)^4} \Big) + \Big[\frac{-\Delta t_2}{a_1b_2 - a_2b_1} + \frac{a_2}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2) \Big]^2 + \Big[\frac{\Delta t_1}{a_1b_2 - a_2b_1} - \frac{a_1}{(a_1b_2 - a_2b_1)^2}(b_2\Delta t_1 - b_1\Delta t_2) \Big]^2 \Bigg] + \sigma_{\Delta t}^2 \Big[\frac{b_1^2 + b_2^2}{(a_1b_2 - a_2b_1)^2} \Big]$$

$$\sigma_{w'}^2 = Var(\delta_{w'}) = Var(\delta_{u'})\Big|_{a_1 \leftrightarrow b_1, a_2 \leftrightarrow b_2}$$

4.6.4 Adam's Method Revisited — Weighted Average of Two-Pairing Estimates

$$\hat{u'} = K_u \frac{1}{T_N} \sum_{k=1}^{T_N} \frac{\hat{u'}_k}{\sigma_{u'_k}^2} \qquad \hat{w'} = K_w \frac{1}{T_N} \sum_{k=1}^{T_N} \frac{\hat{w'}_k}{\sigma_{w'_k}^2}$$

where $K_u = \left(\sum_{k=1}^{T_N} \frac{1}{\sigma_{u'_k}^2} \right)^{-1}$ and $K_w = \left(\sum_{k=1}^{T_N} \frac{1}{\sigma_{w'_k}^2} \right)^{-1}$.

4.6.5 Estimating Velocity Vector Using Global Firing Time Information (Berthold's Method)

In this method cells *are not* paired up. Assuming we have N cells:

$$\begin{bmatrix} u' \\ w' \\ T \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} y_i^2 & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} t_i x_i \\ \sum_{i=1}^{N} t_i y_i \\ \sum_{i=1}^{N} t_i \end{bmatrix}$$

We transform back to u and w as in Adam's Method.

Sensitivities

Assuming N cells (not cell pairs) are equally spaced on a circumference of radius R:

$$\sigma_u^2 \approx (w^2 - u^2)^2 \sigma_{u'}^2 + 4u^2 w^2 \sigma_{w'}^2$$
$$\sigma_w^2 \approx (u^2 - w^2)^2 \sigma_{w'}^2 + 4u^2 w^2 \sigma_{u'}^2$$

Variance of Residuals $f_k \triangleq u'x_k + w'y_k - (t_k - T)$**:**

As expected, the variance of each residual does not depend on anything else except the speed of the curtain, the variance in position estimates, and the variance in the firing time estimates.

$$Var(\delta_{f_k}) \approx= \frac{1}{v^2}\sigma_p^2 + \sigma_{t_k}^2$$

4.6.6 CosCos Algorithm

In this method cells *are* paired up. N refers to the number of pairs available. We have equations of the form:

$$\Delta t_k = \frac{d_k}{v}[\cos(\theta)\cos(\theta_k) + \sin(\theta)\sin(\theta_k)]$$

Letting $\alpha = \frac{\cos(\theta)}{v}$ and $\beta = \frac{\sin(\theta)}{v}$.

Assuming that we have chosen N pairs, we organize the Δt_k's corresponding to each pair in a column vector $\mathbf{t}$. We also compose a matrix, $\mathbf{X}$, by making each of its rows a row vector of the form $\begin{bmatrix} d_k\cos(\theta_k) & d_k\sin(\theta_k) \end{bmatrix}$, corresponding to each pair.

Our least squares estimates are given by:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$$

Notice that $\alpha = u'$ and $\beta = w'$, and each equation formed by pairing cells is equivalent to each of the two equations formed by a two-pairing in Adam's method. We transform to v and θ estimates from α and β as we did in Adam's method.

Sensitivities

$$\sigma_v^2 \approx \frac{v^2}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

$$\sigma_\theta^2 \approx \frac{1}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

4.6.7 Newton-Raphson Algorithm (John's Method)

In this method cells *are* paired up. N refers to the number of cell pairs available. We minimize

$$f(v,\theta) \triangleq \sum_{i=1}^{N} f_i^2$$

where $f_k \triangleq d_k \cos(\theta_k - \theta) - \Delta t_k v$. The solution of v and θ which minimize $f(v,\theta)$ is solved by Newton-Raphson minimization.

Sensitivities

$$\sigma_v^2 \approx \frac{v^2}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

$$\sigma_\theta^2 \approx \frac{1}{2R^2N}(2\sigma_p^2 + v^2\sigma_{\Delta t}^2)$$

Chapter 5

A Least Squares Method for Minimizing a Weighted Sum of Variances

5.0 Acknowledgements and Introduction

Before beginning this chapter, I would like to acknowledge that the work presented in this chapter was done jointly with Professor John Wyatt. We are grateful for Professor Megretski's important help and Professor Strang's guidance and input. Specifically, Professor Megretski outlined the derivation of the optimal weight matrix presented in section 5.3. We also thank Shamim Nemati for his help.

In this chapter, we present a novel way of weighing least squares so as to minimize a weighted sum of the variances of a set of estimates $\hat{x}$ which we are interested in. Although the derivations are done for a general case, it may be helpful for the reader to think of the parameters which we are interested in estimating as the speed and angle of motion of a moving edge. At first glance, this chapter may seem like a digression from the rest of this thesis; however, the goal of this work is to find a way which we can reweigh least squares problems in which we are interested in the accuracy of some estimates more than others. For example, we may be more interested in the fidelity of the angle estimate than that of the speed estimate.

Section 5.1 reviews the optimal weighted least squares solution for the case where we are interested in minimizing the variance of a weighted sum of elements in $\hat{x}$; this is the Best Linear Unbiased Estimator (BLUE) as presented in Professor Strang's textbook *An Introduction to Applied Mathematics*. Section 5.2 proves that the BLUE is optimal for weighing least squares so as to minimize a weighted sum of the variances of a set of estimates $\hat{x}$. Section 5.3 presents a novel method for finding a family of optimal weight matrices; this family of optimal weight matrices leads to the BLUE. Section 5.4 presents

some examples on finding this optimal weight matrix. Section 5.5 is the appendix to this chapter where all the proofs are presented.

5.1 The Best Linear Unbiased Estimator (BLUE)

5.1.1 The Setup:

We consider the equation:

$$Ax = b \tag{5.1.1}$$

$A \in \mathbb{R}^{mxf}$ where we consider A to be fixed and b to be variable. Suppose A has linearly independent columns ($\ker\{A\} = \{0\}$), $x \in \mathbb{R}^{fx1}$, $b \in \mathbb{R}^{mx1}$, $m \geq f \geq 1$ ("m" stands for many, "f" stands for few). We consider only "solutions" $\hat{x}$ with the following 2 properties:

(i) $\hat{x}$ is a linear operator on b, namely: $\hat{x} = Lb$ for some $L \in \mathbb{R}^{fxm}$.

(ii) For any b_p in the range of A, $Lb_p = x_p$ is the exact solution to $Ax_p = b_p$.

A stochastic interpretation is then possible. Suppose that $b = b_p + e$ where $e \in \mathbb{R}^m$ is a zero mean random vector of errors. Then, condition (i) implies that:

$$\hat{x} = Lb = L(b_p + e) \tag{5.1.2}$$

which implies that, since L is linear, $\hat{x}$ is an unbiased estimator of x_p, i.e.

$\mathrm{E}\{\hat{x}\} = Lb_p = x_p$.

The following lemmas lead to a constraint which the class of linear operators L must satisfy in order for condition (ii) to hold and the estimator $\hat{x}$ to be unbiased; their proofs are given in the appendix of this chapter.

Lemma 5.1: Condition (ii) holds if and only if $LA=I$.

Lemma 5.2: The linear estimator $\hat{x} = Lb = L(b_p + e)$ is an unbiased estimator of x_p for every $b_p \in \mathbb{R}^{mx1}$ in the range of A if and only if $LA=I$.

Lemma 5.3: For any $L \in \mathbb{R}^{fxm}$ such that $LA=I$, let $\hat{x} = L(b_p + e)$. Then, the covariance matrix $\Lambda_{\hat{x}}$ of $\hat{x}$ is given by: $\Lambda_{\hat{x}} = L\Lambda_e L^T$.

One class of linear operators L which fulfill conditions (i) and (ii) consists of the weighted linear least squares solutions:

$$L_w = (A^T W A)^{-1} A^T W \tag{5.1.3}$$

for every positive definite symmetric matrix $W \in \mathbb{R}^{mxm}$. The least squares solution $\hat{x} = L_w b$ minimizes:

$$(b - Ax)^T W (b - Ax) \tag{5.1.4}$$

5.1.2 The Best Linear Unbiased Estimator (BLUE)

Let $\Lambda_e \in \mathbb{R}^{mxm}$ be the covariance matrix of the error vector e; we assume Λ_e is positive definite. In his book *Introduction to Applied Mathematics,* Professor Strang shows that the covariance matrix of the error in $\hat{x}$ ($\Lambda_{\hat{x}} = \mathrm{E}\{(\hat{x} - x_p)(\hat{x} - x_p)^T\}$) is minimized in a certain sense when L has the form of equation (5.1.3) with $W = \Lambda_e^{-1}$.

Lemma 5.4: The matrix $L_o = [(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}] \in \mathbb{R}^{fxm}$, fulfills the following criteria:

1. $L_o A=I$ ($\hat{x} = L_o b$ is unbiased).
2. Let $L \in \mathbb{R}^{fxm}$ be any matrix that satisfies $LA=I$. Let $\hat{x} = L_o(b_p + e)$ and $y = L(b_p + e)$. Then, $(\Lambda_y - \Lambda_{\hat{x}})$ is positive semidefinite for any choice of $b_p \in Range(A)$. Furthermore, if $L \neq L_o$, then $(\Lambda_y - \Lambda_{\hat{x}})$ is nonzero.

In other words, $\Lambda_{\hat{x}}(W) \geq \Lambda_{\hat{x}}(\Lambda_e^{-1})$, in the partial ordering sense of matrices. The proof of Lemma 5.4 is given in the appendix of this chapter. We call $\hat{x} = L_o b$ the best linear unbiased estimator of x.

Corollary 5.1: Let $a \in \mathbb{R}^f$; let $L \in \mathbb{R}^{fxm}$ be any matrix that satisfies $LA=I$. Let $\hat{x} = L_o(b_p + e)$ and $y = L(b_p + e)$. Then, $a^T y$ and $a^T \hat{x}$ are weighted sums of the elements in y and $\hat{x}$ respectively and it is easily shown that $a^T \Lambda_y a = \text{var}(a^T y)$ and $a^T \Lambda_{\hat{x}} a = \text{var}(a^T \hat{x})$. By Lemma 5.4, $(\Lambda_y - \Lambda_{\hat{x}})$ is positive semidefinite $\Rightarrow a^T \Lambda_y a \geq a^T \Lambda_{\hat{x}} a \quad \forall a \in \mathbb{R}^m$ which means that the variance of the weighted sum of the elements in y is at least as large as the variance of the weighted sum of the elements in $\hat{x}$ when the same weights a are used for both. Furthermore, since $(\Lambda_y - \Lambda_{\hat{x}}) \neq 0$ for any $L \neq L_o$ with $LA=I$, it follows that for any such L and any $b_p \in Range(A)$, there exists a weight vector $a \in \mathbb{R}^f$ such that $\text{var}(a^T y) > \text{var}(a^T \hat{x})$.

5.2 Our Problem and How It Relates to the BLUE

We now switch our attention from the variance of the weighted sum of the elements in $\hat{x}$ to the *weighted sum of the variances* of the elements in $\hat{x}$. More specifically, we wish to find an unbiased linear estimator L that minimizes a weighted sum Φ of the variances of the elements in $\hat{x}$:

$$\Phi = \sum_{k=1}^{f} p_{kk} \sigma_{\hat{x}_k}^2 \tag{5.2.1}$$

where the weights p_{kk} are positive but otherwise arbitrary. Let $P \in \mathbb{R}^{fxf}$ be a diagonal positive definite matrix. Equation (5.2.1) can be rewritten as $tr\{P\Lambda_{\hat{x}}\} = tr\{PL\Lambda_e L^T\}$, which holds as long as $LA=I$ (by Lemma 5.4). So our objective is stated as follows:

$$\begin{aligned} &\min_{L \in \mathbb{R}^{fxm}} tr\{PL\Lambda_e L^T\} \\ &st\ LA = I \end{aligned} \tag{5.2.2}$$

5.2.1 The Orthogonality Principle:

Before continuing, the abstract form of the orthogonality principle is presented as it will be used later on; it has been adapted from the *Functional Analysis* (6.972) Class Notes of Spring 2007.

Theorem 5.1: Let $\sigma : V \to R$ be a quadratic form on a real vector space V. Let $v_1 \in V$ *and* $\mathcal{U} \subset V$ be an element and a linear subspace of V. Assume that the restriction of σ to $\mathcal{U}$ is positive semidefinite. Then, for every vector $v^* \in v_1 + \mathcal{U}$, the following conditions are equivalent:

1. $\sigma(v) \geq \sigma(v^*)\ \forall\ v \in v_1 + \mathcal{U}$
2. $(v^*, u)_\sigma = 0\ \forall\ u \in \mathcal{U}$

As shown in figure 5-1 below, the projection theorem can be interpreted in terms of the "distance" and the "scalar product"

$$dist(v,u) = \sigma(v-u)^{1/2}, \quad (v,u)_\sigma = \frac{\sigma(v+u) - \sigma(v-u)}{4} \tag{5.2.3}$$

in V, induced by the positive semidefinite quadratic form σ. The theorem claims that a vector $v^* \in v_1 + \mathcal{U}$ has minimal "distance" to the origin $0 \in V$ if and only if it is "orthogonal" to every element in $\mathcal{U}$.

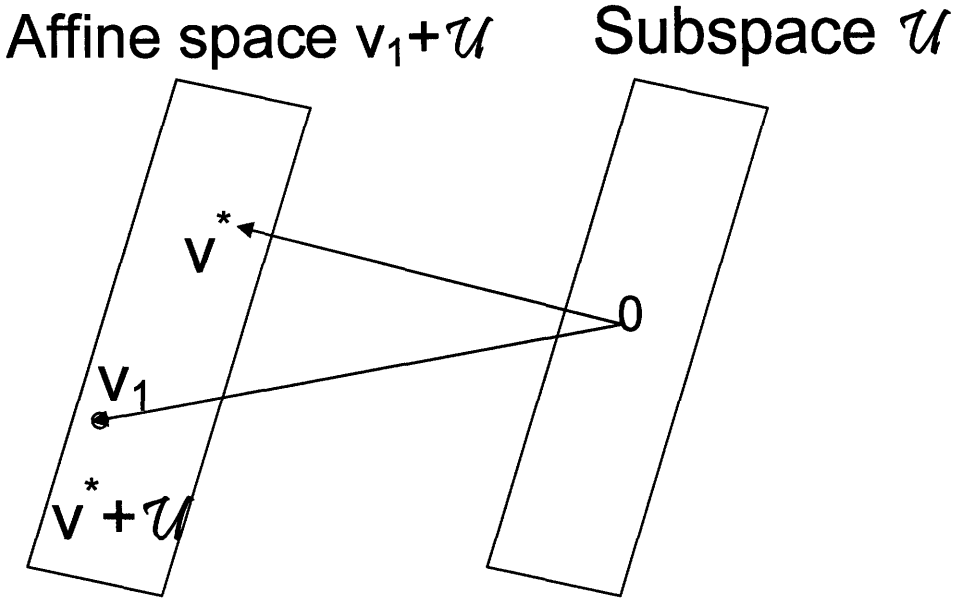

Figure 5-1: Graphical Representation of Theorem 5-1.

5.2.2: Finding the Conditions for an Optimal Solution to Our Problem

Theorem 5.2: Let $P \in R^{fxf}$ be a positive definite symmetric matrix. The *only* optimal solution to:

$$\begin{aligned} &\min_{L \in \mathbb{R}^{fxm}} tr\{PL\Lambda_e L^T\} \\ &st\ LA = I \end{aligned} \tag{5.2.4}$$

is $L_o = (A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}$, and the minimum value of equation (5.2.4) is $tr\{P(A^T \Lambda_e^{-1} A)^{-1}\}$.

Proof:

We will first prove that $L_o = (A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}$ is an optimal solution to equation (5.2.4); we then will prove that it is unique.

Let us define an inner product that maps $\mathbb{R}^{fxm} x \mathbb{R}^{fxm} \rightarrow \mathbb{R}$ in the following way:

$$\langle L, M \rangle = tr\{L\Lambda_e M^T\} \tag{5.2.5}$$

where $L \in \mathbb{R}^{fxm}$ and $M \in \mathbb{R}^{fxm}$. Using equation (5.2.5), we can rewrite our minimization problem as:

$$\begin{aligned} &\min_{L\in\mathbb{R}^{fxm}} tr\{PL\Lambda_e L^T\} = \min_{L\in\mathbb{R}^{fxn}} tr\{P^{1/2}L\Lambda_e L^T P^{1/2}\} = \min_{L\in\mathbb{R}^{fxm}} \left\langle P^{1/2}L, P^{1/2}L \right\rangle \\ &st\ LA = I \qquad\qquad st\ LA = I \qquad\qquad st\ LA = I \end{aligned} \tag{5.2.6}$$

where $P^{1/2}$ is the unique symmetric positive definite square root of P. Let $Q = P^{1/2}L$; Then, the constraint $LA=I$ becomes $QA = P^{1/2}$ and the problem can be posed as follows:

$$\begin{aligned} &\min_{Q\in\mathbb{R}^{fxm}} tr\{Q\Lambda_e Q^T\} \\ &st\ QA = P^{1/2} \end{aligned} \tag{5.2.7}$$

Let us define the following subspaces:

$$\begin{aligned} &\mathfrak{M} \triangleq \{M \in \mathbb{R}^{fxm} \mid MA = 0\} \\ &S \triangleq \{Q \in \mathbb{R}^{fxm} \mid QA = P^{1/2}\} \end{aligned} \tag{5.2.8}$$

The subspaces $\mathfrak{M}$ and S are graphically illustrated in figure 5-2. $Q^* \in S$ is an optimal solution if and only if:

$$\langle Q^*, M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M} \tag{5.2.9}$$

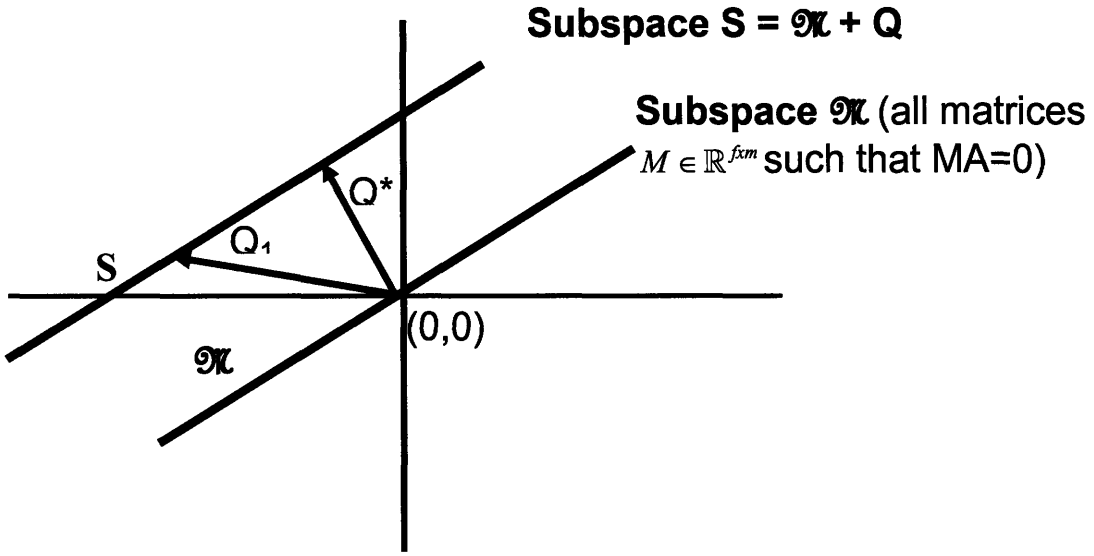

Figure 5-2: Graphical Representation of the Optimal **Q***. Let $Q_1 \in \mathbb{R}^{fxm}$ be any matrix[1] such that $Q_1 A = P^{1/2}$. The set S of all matrices $Q \in \mathbb{R}^{fxm}$ such that $QA = P^{1/2}$ can be written as $S = \mathfrak{M} + Q_1$.

We have thus established our optimality criteria on Q. By lemma 5.4, we know that $L_o A = I$ and thus $Q^* A = P^{1/2}$. We test to see if the L_o obtained by finding the BLUE of x_p in section 5.1 satisfies equation (5.2.9). We have $Q^* = P^{1/2} L_o = P^{1/2}(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}$ and thus:

$$\begin{aligned} \langle Q^*, M \rangle &= tr\{P^{1/2}(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1} \Lambda_e M^T\} = \\ & tr\{P^{1/2}(A^T \Lambda_e^{-1} A)^{-1} (MA)^T\} = tr\{0\} = 0 \end{aligned} \tag{5.2.10}$$

and thus, equation (5.2.9) is satisfied by Q^*.

We have shown that $Q^* = P^{1/2} L_o$ is an optimal solution to the problem posed in equation (5.2.7) and thus, L_o is an optimal solution to the problem posed in equation (5.2.4). We now show that Q^* is the *unique* optimal solution to the problem posed in equation (5.2.7) and thus L_o is the *unique* optimal solution to the problem posed in equation (5.2.4).

---

[1] There exists at least one such Q_1 since $Q^* = P^{1/2} L_o = P^{1/2}(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}$ satisfies $Q^* A = P^{1/2}$

We know that $Q^* \in S$; thus, all feasible solutions Q to (5.2.7) can be written as $Q = Q^* + \mathfrak{M}$. Let $Q_1 = Q^* + M_1$ be any optimal solution to (5.2.7); we will show that $M_1 = 0$ and thus, the only optimal solution to (5.2.7) is Q^*. By the orthogonality principle, since Q_1 is optimal:

$$\begin{aligned} \langle Q_1, M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M} &\Leftrightarrow \langle (Q^* + M_1), M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M} \\ &\Leftrightarrow tr\{Q^* \Lambda_e M^T\} + tr\{M_1 \Lambda_e M^T\} = 0 \ \forall \mathrm{M} \in \mathfrak{M} \\ &\Leftrightarrow \langle Q^*, M \rangle + \langle M_1, M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M} \end{aligned} \tag{5.2.11}$$

However, from (5.2.10), we have that $\langle Q^*, M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M}$ and thus, it must be the case that:

$$\langle M_1, M \rangle = 0 \ \forall \mathrm{M} \in \mathfrak{M} \tag{5.2.12}$$

Equation (5.2.12) implies that:

$$\langle M_1, M_1 \rangle = 0 \Leftrightarrow tr\{M_1 \Lambda_e M_1^T\} = 0 \tag{5.2.13}$$

Since $M_1 \Lambda_e M_1^T$ is positive semidefinite and has a trace of zero, it must be the case that $M_1 \Lambda_e M_1^T = 0$. Since Λ_e is symmetric positive definite ($\ker\{\Lambda_e\} = \ker\{\Lambda_e^T\} = \{0\}$), we can conclude the following:

$$\begin{aligned} M_1 \Lambda_e M_1^T = 0 \Rightarrow x^T M_1 \Lambda_e M_1^T x = 0 \ \forall x \in \mathbb{R}^f \Rightarrow M_1^T x = 0 \ \forall x \in \mathbb{R}^f \\ \Rightarrow M_1^T = 0 \Rightarrow M_1 = 0 \end{aligned} \tag{5.2.14}$$

Since $M_1 = 0$, all optimal solutions $Q_1 = Q^*$; thus, Q* is the unique optimal solution for (5.2.7) meaning that L_o is the unique optimal solution for (5.2.4). Plugging in $L_o = (A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}$ into equation (5.2.4) gives a cost of:

$$tr\{P(A^T \Lambda_e^{-1} A)^{-1}\} \tag{5.2.15}$$

■

We have shown that the linear operator L_o minimizes both the *variance of the weighted sum* of the elements in $\hat{x}$ (Corollary 5.1) and the *weighted sum of the variances* of the elements in $\hat{x}$ (Theorem 5.2). Furthermore, it is unique in both cases. It is important to realize the strength of theorem 5.2: it made no assumptions on the form of the optimal linear operator L except that it fulfills the condition $LA=I$. In other words, it did not assume that L was of the form $(A^TWA)^{-1}A^TW$ for some weight matrix W: however, out of all possible L which satisfy $LA=I$, the optimal L turns out to be of the form $(A^TWA)^{-1}A^TW$ with $W=\Lambda_e^{-1}$. Section 5.3 will show that there exists a class of optimal weight matrices W_o which lead to a least squares solution which minimizes the weighted sum of the variances in $\hat{x}$. Through examples, it is shown that all such W_o lead to the same optimal linear operator L_o.

5.3 Deriving the Optimal Weight Matrix W_o

We consider the equation:

$$Ax = b_p + e \equiv b \tag{5.3.1}$$

where $A \in \mathbb{R}^{mxf}$, $x \in \mathbb{R}^{fx1}$, and $b \in \mathbb{R}^{mx1}$, and e is zero mean random vector of errors. The weighted linear least squares solution is: $\hat{x} = L_w b = L_w(b_p + e)$ where L_w is of the form:

$$L_w = (A^TWA)^{-1}A^TW \tag{5.3.2}$$

As mentioned in the previous sections, we are interested in the *weighted sum of the variances* of the elements in $\hat{x}$. More specifically, we wish to find an unbiased linear estimator L that minimizes a weighted sum of the variances of the elements in $\hat{x}$:

$$\varphi(W) = \sum_{k=1}^{f} p_{kk}\sigma_{\hat{x}_k}^2 \tag{5.3.3}$$

where the weights p_{kk} are positive but otherwise arbitrary. Let $P \in \mathbb{R}^{fxf}$ be a diagonal positive definite matrix. Equation (5.3.3) can be rewritten as $tr\{P\Lambda_{\hat{x}}\} = tr\{PL_w\Lambda_e L_w^T\}$, which holds for all L_w of the form of equation (5.3.2) as well as any matrix $D \in \mathbb{R}^{fxm}$ which satisfies $DA = I$. So our objective is stated as follows:

$$\begin{aligned} &\min_{L \in \mathbb{R}^{fxm}} tr\{PL_w\Lambda_e L_w^T\} \\ &st\ L_w A = I \end{aligned} \tag{5.3.4}$$

Our goal is to find the family of matrices W which minimize equation (5.3.4). Since A has linearly independent columns by assumption, $\Re(A)$ is an f-dimensional subspace of $\mathbb{R}^m$. We pick a basis $\{v_1, v_2, \bullet\bullet\bullet, v_f\}$ for $\Re(A)$ and extend it to a basis $\{v_1, v_2, \bullet\bullet\bullet, v_f, v_{f+1}, \bullet\bullet\bullet, v_m\}$ for $\mathbb{R}^m$. Let $\mathbb{R}^m$ have any basis. In these bases, A has the matrix representation:

$$A = \begin{matrix} f\{ \\ m-f\{ \end{matrix} \underbrace{\begin{pmatrix} M \\ --- \\ 0 \end{pmatrix}}_{f}, \tag{5.3.5}$$

where $M \in \mathbb{R}^{fxf}$ is nonsingular. The block partition of W is with respect to the new basis for $\mathbb{R}^m$ is:

$$W = \begin{matrix} f\ \{ \\ m-f\ \{ \end{matrix} \underbrace{\begin{bmatrix} X & | & Y \\ - & - & - \\ Y^T & | & Z \end{bmatrix}}_{f\quad m-f}, \tag{5.3.6}$$

where $X \in \mathbb{R}^{fxf}$ and $Z \in \mathbb{R}^{(m-f)x(m-f)}$ are positive-definite symmetric (since W is). Recall that any real, symmetric, positive definite matrix has a real, symmetric, positive definite square root. We use this to factor Λ_e into:

$$\Lambda_e = F_{\Lambda_e} F_{\Lambda_e}^T;\ \Lambda_e, F_{\Lambda_e} \in \mathbb{R}^{mxm} \tag{5.3.7}$$

Also, let $L \triangleq P^{1/2} L_w F_{\Lambda_e}$: then,

$$tr\{PL_w \Lambda_e L_w^{\ T}\} = tr\{LL^T\} \tag{5.3.8}$$

Let's decompose F_{Λ_e} into upper and lower blocks:

$$F_{\Lambda_e} = \begin{matrix} f\{ \\ m-f\{ \end{matrix} \underbrace{\begin{bmatrix} G \\ --- \\ H \end{bmatrix}}_{m} \tag{5.3.9}$$

In order to express L_w and L in terms of the matrices *X, Y, Z, G,* and *H*, note that:

$$\begin{gathered} (A^T WA)^{-1} = (M^T XM)^{-1} = M^{-1} X^{-1} M^{T^{-1}}, \\ A^T W F_{\Lambda_e} = M^T (XG + YH) \end{gathered} \tag{5.3.10}$$

$$L = P^{1/2} L_W F_{\Lambda_e} = P^{1/2} (A^T WA)^{-1} A^T W F_{\Lambda_e} = P^{1/2} M^{-1} X^{-1} M^{T^{-1}} M^T (XG + YH) = P^{1/2} M^{-1} (G + KH) \tag{5.3.11}$$

where $K = X^{-1}Y$, $K \in \mathbb{R}^{(f)x(m-f)}$. Note that *K* is of the same dimension as *Y*. Moreover, since X^{-1} is positive definite and $Y \in \mathbb{R}^{(f)x(m-f)}$ can range over all *fx(m-f)* matrices, $K \in \mathbb{R}^{(f)x(m-f)}$ can range over all *fx(m-f)* matrices. Also, note that for each $K \in \mathbb{R}^{(f)x(m-f)}$, there correspond many pairs $(X^{-1} \in \mathbb{R}^{fxf}, Y \in \mathbb{R}^{(f)x(m-f)})$ such that $K = X^{-1}Y$. Thus, finding an optimal K_o does not give a unique solution for X_o and Y_o and thus does not give a unique solution for the optimal weight matrix W_o.

Now, our objective is the following:

$$\begin{gathered} \min \varphi(W) = \min_{L \in \mathbb{R}^{fxm}} tr\{LL^T\} \\ st\ L_w A = I \qquad st\ L_w A = I \end{gathered} \tag{5.3.12}$$

Let us define an inner product $\mathbb{R}^{fxm} x \mathbb{R}^{fxm} \rightarrow \mathbb{R}$ in the following way:

$$\langle C, D \rangle = tr\{CD^T\} \tag{5.3.13}$$

for all $C \in \mathbb{R}^{fxm}$ and $D \in \mathbb{R}^{fxm}$. Thus:

$$\varphi(W) = \| L \|^2 = \langle L, L \rangle = \langle P^{1/2} M^{-1} (G + KH), P^{1/2} M^{-1} (G + KH) \rangle \tag{5.3.14}$$

The expression given in equation (5.3.11) above is the sum of a fixed matrix and a linear subspace of matrices. Thus, from the orthogonality principle, for any (candidate optimal) choice of $K \in \mathbb{R}^{(f)x(m-f)}$

$$\begin{aligned} &\left\| P^{1/2} M^{-1} (G + K_o H) \right\| \leq \left\| P^{1/2} M^{-1} (G + KH) \right\|, \ \forall K \in \mathbb{R}^{(f)x(m-f)} \Leftrightarrow \\ &\left\langle P^{1/2} M^{-1} (G + K_o H), \ \ P^{1/2} M^{-1} KH \right\rangle = 0, \ \forall K \in \mathbb{R}^{(f)x(m-f)} \end{aligned} \tag{5.3.15}$$

See figure 5-3 for an illustration of the orthogonality principle for equation (5.3.15).

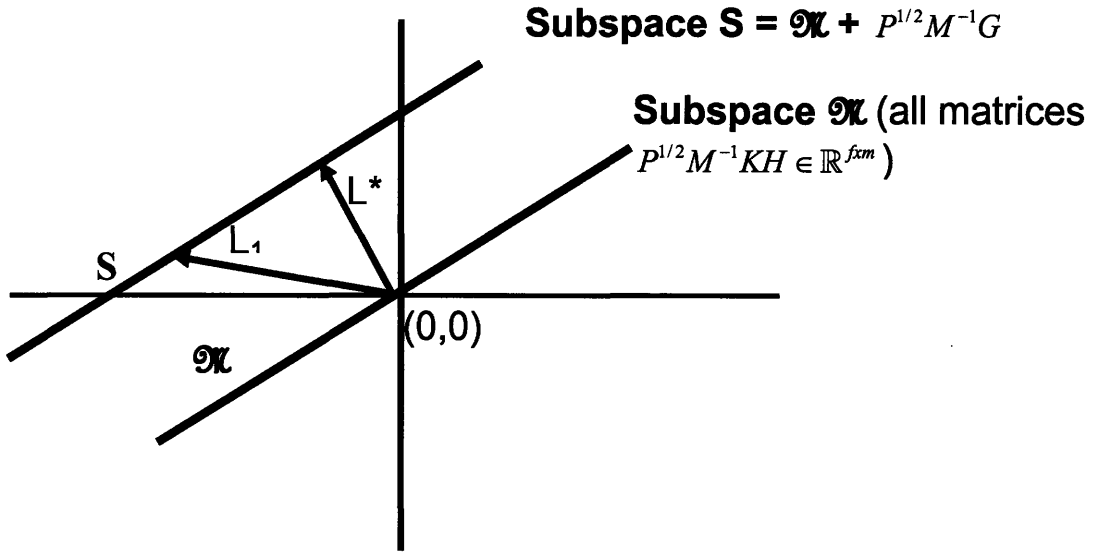

Figure 5-3: Let $\mathfrak{M}$ be the subspace of all matrices $P^{1/2} M^{-1} KH \in \mathbb{R}^{fxm}$ where $K \in \mathbb{R}^{fx(m-f)}$ can vary over all *fx(m-f)* matrices. Let S be the subspace defined by adding the matrix $P^{1/2} M^{-1} G \in \mathbb{R}^{fxm}$ to all matrices in $\mathfrak{M}$. The matrix $L^* = P^{1/2} M^{-1} (G + K_o H)$ which lies in S has the smallest norm compared to all other $L_1 = P^{1/2} M^{-1} (G + KH)$ where $K \neq K_o$ which lie in S.

Thus, by the orthogonality principle, the condition in equation (5.3.15) becomes:

$$\begin{aligned} &tr\{P^{1/2}M^{-1}(G+K_oH)(P^{1/2}M^{-1}KH)^T\}=0 \ \ \forall K\in\mathbb{R}^{(f)x(m-f)} \Leftrightarrow \\ &tr\{(P^{1/2}M^{-1}G+P^{1/2}M^{-1}K_oH)H^TK^T(M^{-1})^TP^{1/2}\}=0 \ \ \forall K\in\mathbb{R}^{(f)x(m-f)} \Leftrightarrow \\ &tr\{[\underbrace{(M^{-1})^TPM^{-1}GH^T}_{\mathbb{R}^{(f)x(m-f)}}+\underbrace{(M^{-1})^TPM^{-1}K_oHH^T}_{\mathbb{R}^{(f)x(m-f)}}]\ \underbrace{K^T}_{\mathbb{R}^{(m-f)x(f)}}\}=0 \ \ \forall K\in\mathbb{R}^{(f)x(m-f)} \end{aligned} \quad (5.3.16)$$

Since the term in brackets in equation (5.3.16) lies in $\mathbb{R}^{(f)x(m-f)}$ and (5.3.16) holds for all $K\in\mathbb{R}^{(f)x(m-f)}$, it must be true that:

$$\underbrace{(M^{-1})^TPM^{-1}GH^T}_{\Sigma}+\underbrace{(M^{-1})^TPM^{-1}}_{\Omega}K_o\underbrace{HH^T}_{\Pi}=0 \quad (5.3.17)$$

where:

$$\Omega\triangleq(M^{-1})^TPM^{-1},\ \ \Sigma\triangleq\Omega GH^T,\ \ \Pi\triangleq HH^T \quad (5.3.18)$$

Now, recall that P is symmetric positive definite and M has full rank (rank(M)=f) because its columns are a basis of $\mathbb{R}^f$; thus, $\ker\{\Omega\}=0$ and Ω is invertible. Also, F_{Λ_e} is symmetric positive definite which means that it has full row rank; thus, $H\in\mathbb{R}^{(m-f)xm}$ has full row rank which means that $\Pi=HH^T\in\mathbb{R}^{(m-f)x(m-f)}$ is invertible. Therefore, we can solve for K_o:

$$K_o=-\Omega^{-1}\Sigma\Pi^{-1}\Leftrightarrow X_o^{-1}Y_o=-\underbrace{\{(MP^{-1}M^T)\}}_{\Omega^{-1}}\underbrace{\{(M^{-1})^TPM^{-1}GH^T\}}_{\Sigma}\underbrace{\{HH^T\}^{-1}}_{\Pi^{-1}} \quad (5.3.19)$$

Equation (5.3.19) gives a rule for picking the optimal X and Y entries of the weight matrix W. Equation (5.3.19) also says that we are free to choose a positive definite Z so long as it keeps W positive definite. Thus, we may take $Z\in\mathbb{R}^{(m-f)x(m-f)}$ to be any $(m-f)\times(m-f)$ positive definite matrix that does not violate the positive-definiteness

requirement of W. One such Z is obtained by the Nyström approximation technique and is given by:

$$\tilde{Z} = Y^T X^{-1} Y \tag{5.3.20}$$

Thus, the class of weight matrices W_o which minimize equation (5.3.4) are of the form:

$$W_o = \begin{matrix} f \\ m-f \end{matrix} \begin{matrix} \{ \\ \{ \end{matrix} \underbrace{\begin{bmatrix} X & | & Y \\ - & - & - \\ Y^T & | & Y^T X^{-1} Y + \Delta \end{bmatrix}}_{f \qquad\qquad m-f}, \tag{5.3.21}$$

where $\Delta \in \mathbb{R}^{(m-f)x(m-f)}$ is chosen such that W_o is positive definite. If $\Delta = 0$, W_o becomes singular.

5.4 Examples of Optimal Weight Matrix Construction

In order to clarify the theory presented in sections 5.3, a series of examples will be presented. The goal is to increase the reader's intuition about how the optimal weight matrix for a weighted least squares estimator is determined. Moreover, these examples empirically show that $W = \Lambda_e^{-1}$ is in fact a special case of W_o of equation (5.3.21).

5.4.1: The 2x1 Case with Correlated Errors

5.4.1.1 Verification that BLUE Gives the Least Positive Definite $\Lambda_{\hat{x}}$

Consider the following setup:

$$\underbrace{\begin{bmatrix} a \\ 0 \end{bmatrix}}_{A \in \mathbb{R}^{2x1}} x_p = \underbrace{\begin{bmatrix} b_{1p} \\ b_{2p} \end{bmatrix}}_{b_p \in \mathbb{R}^{2x1}} + \underbrace{\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}}_{e \in \mathbb{R}^{2x1}} \tag{5.4.1}$$

where x_p is the exact solution to $Ax_p = b_p$. The error vector e is a zero mean random vector of noise added to b_p; Thus, Λ_e (the covariance matrix of the error vector e) is:

$$\Lambda_e = \begin{bmatrix} \sigma_{e_1}^2 & \sigma_{e_1 e_2} \\ \sigma_{e_1 e_2} & \sigma_{e_2}^2 \end{bmatrix} \tag{5.4.2}$$

where $\sigma_{e_1 e_2}$ is the covariance between e_1 and e_2. We seek an unbiased estimator $\hat{x}$ of x_p which satisfies the two conditions presented in section 5.1; such an $\hat{x}$ is of the form $\hat{x} = L_d(b_p + e)$ where:

$$L_d = \begin{bmatrix} \frac{1}{a} & d \end{bmatrix} \forall \mathrm{d} \in \mathbb{R} \tag{5.4.3}$$

Thus, we have the following relationship:

$$\hat{x} = L_d(b_p + e) = \begin{bmatrix} \frac{1}{a} & d \end{bmatrix} \begin{bmatrix} b_{1p} + e_1 \\ e_2 \end{bmatrix} = \frac{b_{1p} + e_1}{a} + de_2 \tag{5.4.4}$$

We know that $x_p = \frac{b_{1p}}{a}$; thus, $\hat{x} - x_p = \frac{e_1}{a} + de_2$ and the covariance matrix of the error in $\hat{x}$ takes the form:

$$\Lambda_{\hat{x}} = \mathrm{E}\{(\hat{x} - x_p)(\hat{x} - x_p)^T\} = \frac{\sigma_{e_1}^2}{a^2} + d^2\sigma_{e_2}^2 + 2\frac{d}{a}\sigma_{e_1}\sigma_{e_2} \tag{5.4.5}$$

Our first goal is to verify that Lemma 5.4 of section 5.1.2 holds for this example. Consider the BLUE $L_o = [(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}] \in \mathbb{R}^{1x2}$. We call $\hat{x}^*$ the estimate $L_o(b_p + e)$ of x_p. Substituting for the matrices in this example gives:

$$L_o = \begin{bmatrix} \frac{1}{a} & -\frac{\sigma_{e_1 e_2}}{a\sigma_{e_2}^2} \end{bmatrix} \tag{5.4.6}$$

This gives an $\hat{x}^*$ of:

$$\hat{x}^* = L_o(b_p + e) = \frac{b_{1p} + e_1}{a} - \frac{e_2\sigma_{e_1e_2}}{a\sigma_{e_2}^2} \tag{5.4.7}$$

Thus:

$$\hat{x}^* - x_p = \frac{e_1}{a} - \frac{e_2\sigma_{e_1e_2}}{a\sigma_{e_2}^2} \Rightarrow \Lambda_{\hat{x}^*} = \frac{\sigma_{e_1}^2}{a^2} - \frac{\sigma_{e_1e_2}^2}{a^2\sigma_{e_2}^2} \tag{5.4.8}$$

Equation (5.4.5) gives the covariance matrix of the error $\hat{x} - x_p$ and equation (5.4.8) the covariance matrix of the error $\hat{x}^* - x_p$. Subtracting (5.4.8) from (5.4.5) leads to:

$$\Lambda_{\hat{x}} - \Lambda_{\hat{x}^*} = d^2\sigma_{e_2}^2 + 2\frac{d}{a}\sigma_{e_1e_2} + \frac{\sigma_{e_1e_2}^2}{a^2\sigma_{e_2}^2} > 0 \ \forall d \neq -\frac{\sigma_{e_1e_2}}{a\sigma_{e_2}^2} \tag{5.4.9}$$

which means that $\Lambda_{\hat{x}} \geq \Lambda_{\hat{x}^*}$ in the partial ordering sense of matrices; it is easy to establish that $d = -\frac{\sigma_{e_1e_2}}{a\sigma_{e_2}^2}$ is a minimum of equation (5.4.9); for this value of d, $\hat{x}^* = \hat{x}$ and $\Lambda_{\hat{x}} - \Lambda_{\hat{x}^*} = 0$. Thus, we conclude that the BLUE $L_o = \left[\frac{1}{a} \quad -\frac{\sigma_{e_1e_2}}{a\sigma_{e_2}^2}\right]$ satisfies Lemma 5.4 of section 5.1.2.

5.4.1.2 Optimal Weight Matrix Construction

Now, consider the class of estimators $L_w = (A^TWA)^{-1}A^TW$ presented in section 5.1.1. Our goal is to find the class of weight matrices W_o of the form of equation (5.3.21) which satisfy the constraint in equation (5.3.17). We then test to see if the BLUE

$L_o = \begin{bmatrix} \frac{1}{a} & -\frac{\sigma_{e_1e_2}}{a\sigma_{e_2}^2} \end{bmatrix}$ is a part of that class. We must first define the matrices $M, P, G,$ and H for this example:

$$\underbrace{M}_{1x1} = a, \quad \underbrace{P}_{1x1} = p_1 > 0, \quad \underbrace{F_{\Lambda_e}}_{2x2} = \begin{bmatrix} \sigma_{e_1} & 0 \\ \frac{\sigma_{e_1e_2}}{\sigma_{e_1}} & \sqrt{\sigma_{e_2}^2 - \frac{\sigma_{e_1e_2}^2}{\sigma_{e_1}^2}} \end{bmatrix} \Rightarrow$$

$$\underbrace{G}_{1x2} = \begin{bmatrix} \sigma_{e_1} & 0 \end{bmatrix} \text{ and } \underbrace{H}_{1x2} = \begin{bmatrix} \frac{\sigma_{e_1e_2}}{\sigma_{e_1}} & \sqrt{\sigma_{e_2}^2 - \frac{\sigma_{e_1e_2}^2}{\sigma_{e_1}^2}} \end{bmatrix} \tag{5.4.10}$$

Thus, we have the following $\Omega, \Sigma,$ and Π:

$$\begin{aligned} \Omega &= (M^{-1})^T P M^{-1} = \frac{p_1}{a^2} \\ \Sigma &= \Omega G H^T = \frac{\sigma_{e_1e_2} p_1}{a^2} \\ \Pi &= \sigma_{e_2}^2 \end{aligned} \tag{5.4.11}$$

Therefore:

$$K_o = X_0^{-1} Y_o = -\Omega^{-1}\Sigma\Pi^{-1} = -\frac{\sigma_{e_1e_2}}{\sigma_{e_2}^2} \tag{5.4.12}$$

Equation (5.4.12) gives the optimal K_o; note that there are many pairs (X_o, Y_o) with $X_o > 0$ that satisfy (5.4.12). Thus, all pairs (X_o, Y_o) are of the following form:

$$X_o = c^{-1}\sigma_{e_2}^2, \; Y_o = -c^{-1}\sigma_{e_1e_2}, \; c>0 \tag{5.4.13}$$

We then choose a scalar Z_o to ensure that W_o is positive definite; the Nystrom approximation given in equation (5.3.20) leads to $Z_o = \frac{\sigma_{e_1e_2}^2}{c\sigma_{e_2}^2} + \Delta, \; \forall \Delta > 0$ Thus:

$$W_o = \begin{bmatrix} c^{-1}\sigma_{e_2}^2 & -c^{-1}\sigma_{e_1e_2} \\ -c^{-1}\sigma_{e_1e_2} & \frac{\sigma_{e_1e_2}^2}{c\sigma_{e_2}^2} + \Delta \end{bmatrix} \tag{5.4.14}$$

Therefore:

$$L_w = (A^T W_o A)^{-1} A^T W_o = \begin{bmatrix} \frac{1}{a} & -\frac{\sigma_{e_1e_2}}{a\sigma_{e_2}^2} \end{bmatrix} \tag{5.4.15}$$

Notice that L_w in equation (5.4.15) is exactly the BLUE L_o of equation (5.4.6). Thus, for any choice of $c>0$, our optimal weight matrix W_o leads to the BLUE L_o.

5.4.2: The 3x1 Case with Correlated Errors

5.4.2.1 Verification that BLUE Gives the Least Positive Definite $\Lambda_{\hat{x}}$

Consider the following setup:

$$\underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \\ 0 & 0 \end{bmatrix}}_{A\in\mathbb{R}^{3x2}} x_p = \underbrace{\begin{bmatrix} b_{1p} \\ b_{2p} \\ b_{3p} \end{bmatrix}}_{b_p\in\mathbb{R}^{3x1}} + \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ e_2 \end{bmatrix}}_{e\in\mathbb{R}^{3x1}} \tag{5.4.16}$$

where x_p the exact solution to $Ax_p = b_p$. The error vector e is a zero mean random vector of noise added to b_p; moreover, let e_2 and e_3 be correlated and let e_1 be uncorrelated with both e_2 and e_3. Thus, Λ_e (the covariance matrix of the error vector e) is:

$$\Lambda_e = \begin{bmatrix} \sigma_{e_1}^2 & 0 & 0 \\ 0 & \sigma_{e_2}^2 & \sigma_{e_2e_3} \\ 0 & \sigma_{e_2e_3} & \sigma_{e_3}^2 \end{bmatrix} \tag{5.4.17}$$

We seek an unbiased estimator $\hat{x}$ of x_p which satisfies the two conditions presented in section 5.1.1; such an $\hat{x}$ is of the form $\hat{x} = L_d(b_p + e)$ where:

$$L_d = \begin{bmatrix} \frac{1}{a_1} & 0 & d_1 \\ 0 & \frac{1}{a_2} & d_2 \end{bmatrix} \forall d_1, d_2 \in \mathbb{R} \tag{5.4.18}$$

Note that $\hat{x}$ is unbiased for $b_{3p} = 0$. Thus, we have the following relationship:

$$\hat{x} = L_d(b_p + e) = \begin{bmatrix} \frac{1}{a_1} & 0 & d_1 \\ 0 & \frac{1}{a_2} & d_2 \end{bmatrix} \begin{bmatrix} b_{1p} + e_1 \\ b_{2p} + e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \frac{b_{1p} + e_1}{a_1} + d_1 e_3 \\ \frac{b_{2p} + e_2}{a_2} + d_2 e_3 \end{bmatrix} \tag{5.4.19}$$

We know that $x_p = \begin{bmatrix} \frac{b_{1p}}{a_1} \\ \frac{b_{2p}}{a_2} \end{bmatrix}$; thus, $\hat{x} - x_p = \begin{bmatrix} \frac{e_1}{a_1} + d_1 e_3 \\ \frac{e_2}{a_2} + d_2 e_3 \end{bmatrix}$ and the covariance matrix of the error in $\hat{x}$ takes the form:

$$\Lambda_{\hat{x}} = \mathrm{E}\{(\hat{x} - x_p)(\hat{x} - x_p)^T\} = \begin{bmatrix} \frac{\sigma_{e_1}^2}{a_1^2} + d_1^2 \sigma_{e_3}^2 & d_1 d_2 \sigma_{e_3}^2 + \frac{d_1}{a_2} \sigma_{e_2 e_3} \\ d_1 d_2 \sigma_{e_3}^2 + \frac{d_1}{a_2} \sigma_{e_2 e_3} & \frac{\sigma_{e_2}^2}{a_2^2} + d_2^2 \sigma_{e_3}^2 + \frac{2 d_2 \sigma_{e_2 e_3}}{a_2} \end{bmatrix} \tag{5.4.20}$$

Our first goal is to verify that Lemma 5.4 of section 5.1.2 holds for this example. Consider the BLUE $L_o = [(A^T \Lambda_e^{-1} A)^{-1} A^T \Lambda_e^{-1}] \in \mathbb{R}^{2x3}$. We call $\hat{x}^*$ the estimate $L_o(b_p + e)$ of x_p. Substituting for the matrices in this example gives:

$$L_o = \begin{bmatrix} \frac{1}{a_1} & 0 & 0 \\ 0 & \frac{1}{a_2} & -\frac{\sigma_{e_2 e_3}}{a_2 \sigma_{e_3}^2} \end{bmatrix} \tag{5.4.21}$$

This gives an $\hat{x}^*$ of:

$$\begin{bmatrix} \hat{x}_1^* \\ \hat{x}_2^* \end{bmatrix} = L_o(b_p + e) = \begin{bmatrix} \dfrac{b_{1p} + e_1}{a_1} \\ \dfrac{b_{2p} + e_2}{a_2} - \dfrac{\sigma_{e_2e_3}}{a_2\sigma_{e_3}^2} e_3 \end{bmatrix} \tag{5.4.22}$$

Therefore:

$$\hat{x}^* - x_p = \begin{bmatrix} \dfrac{e_1}{a_1} \\ \dfrac{e_2}{a_2} - \dfrac{\sigma_{e_2e_3}}{a_2\sigma_{e_3}^2} e_3 \end{bmatrix} \Rightarrow \Lambda_{\hat{x}^*} = \begin{bmatrix} \dfrac{\sigma_{e_1}^2}{a_1^2} & 0 \\ 0 & \dfrac{\sigma_{e_2}^2}{a_2^2} - \dfrac{\sigma_{e_2e_3}^2}{a_2^2\sigma_{e_3}^2} \end{bmatrix} \tag{5.4.23}$$

Equation (5.4.20) gives the covariance matrix of the error $\hat{x} - x_p$ and equation (5.4.23) the covariance matrix of the error $\hat{x}^* - x_p$. Subtracting (5.4.23) from (5.4.20) leads to:

$$\Lambda_{\hat{x}} - \Lambda_{\hat{x}^*} = \begin{bmatrix} d_1^2\sigma_{e_3}^2 & d_1d_2\sigma_{e_3}^2 + \dfrac{d_1}{a_2}\sigma_{e_2e_3} \\ d_1d_2\sigma_{e_3}^2 + \dfrac{d_1}{a_2}\sigma_{e_2e_3} & d_2^2\sigma_{e_3}^2 + \dfrac{2d_2\sigma_{e_2e_3}}{a_2} + \dfrac{\sigma_{e_2e_3}^2}{a_2^2\sigma_{e_3}^2} \end{bmatrix} \tag{5.4.24}$$

The matrix of equation (5.4.24) has the following eigenvalues:

$$\lambda_1 = d_1^2\sigma_{e_3}^2, \ \lambda_2 = 0 \tag{5.4.25}$$

Since $\lambda_1, \lambda_2 \geq 0$, $\Lambda_{\hat{x}} - \Lambda_{\hat{x}^*}$ is symmetric positive semidefinite and we conclude that the BLUE $L_o = \begin{bmatrix} \dfrac{1}{a_1} & 0 & 0 \\ 0 & \dfrac{1}{a_2} & -\dfrac{\sigma_{e_2e_3}}{a_2\sigma_{e_3}^2} \end{bmatrix}$ satisfies Lemma 5.4 of section 5.1.2.

5.4.2.2 Optimal Weight Matrix Construction

Now, consider the class of estimators $L_w = (A^T W A)^{-1} A^T W$ presented in section 5.1. Our goal is to find the class of weight matrices W_o of the form of equation (5.3.21) which satisfy the constraint in equation (5.3.17). We then test to see if the BLUE

$L_o = \begin{bmatrix} \frac{1}{a_1} & 0 & 0 \\ 0 & \frac{1}{a_2} & -\frac{\sigma_{e_2e_3}}{a_2\sigma_{e_3}^2} \end{bmatrix}$ is a part of that class. We must first define the matrices $M, P, G,$ and H for this example:

$$\underbrace{M}_{2x2} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}, \quad \underbrace{P}_{2x2} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} > 0$$

$$\underbrace{F_{\Lambda_e}}_{3x3} = \begin{bmatrix} \sigma_{e_1} & 0 & 0 \\ 0 & \sigma_{e_2} & 0 \\ 0 & \frac{\sigma_{e_2e_3}}{\sigma_{e_2}} & \sqrt{\sigma_{e_3}^2 - \frac{\sigma_{e_2e_3}}{\sigma_{e_2}^2}} \end{bmatrix} \Rightarrow \tag{5.4.26}$$

$$\underbrace{G}_{2x3} = \begin{bmatrix} \sigma_{e_1} & 0 & 0 \\ 0 & \sigma_{e_2} & 0 \end{bmatrix} \text{ and } \underbrace{H}_{1x3} = \begin{bmatrix} 0 & \frac{\sigma_{e_2e_3}}{\sigma_{e_2}} & \sqrt{\sigma_{e_3}^2 - \frac{\sigma_{e_2e_3}}{\sigma_{e_2}^2}} \end{bmatrix}$$

Thus, we have the following $\Omega, \Sigma,$ and Π :

$$\Omega = (M^{-1})^T P M^{-1} = \begin{bmatrix} \frac{p_1}{a_1^2} & 0 \\ 0 & \frac{p_2}{a_2^2} \end{bmatrix}$$

$$\Sigma = \Omega G H^T = \begin{bmatrix} 0 \\ \frac{\sigma_{e_2 e_3} p_2}{a_2^2} \end{bmatrix} \tag{5.4.27}$$

$$\Pi = \sigma_{e_3}^2$$

Therefore:

$$K_o = -\Omega^{-1} \Sigma \Pi^{-1} = -\begin{bmatrix} 0 \\ \frac{\sigma_{e_2 e_3}}{\sigma_{e_3}^2} \end{bmatrix} \tag{5.4.28}$$

Equation (5.4.28) gives the optimal K_o; note that there are many pairs (X_o, Y_o) where X_o is positive definite that satisfy (5.4.28). Let $C \in \mathbb{R}^{2x2}$ be any positive definite symmetric matrix with entries:

$$C = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} \tag{5.4.29}$$

then, all pairs (X_o, Y_o) are of the following form:

$$X_o = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix}, \; Y_o = -\begin{bmatrix} \frac{c_2 \sigma_{e_2 e_3}}{\sigma_{e_3}^2} \\ \frac{c_3 \sigma_{e_2 e_3}}{\sigma_{e_3}^2} \end{bmatrix} \tag{5.4.30}$$

We then choose a scalar Z_o to ensure that W_o is positive definite; the Nystrom approximation given in equation (5.3.20) leads to $Z_o = \frac{c_3 \sigma_{e_2 e_3}^2}{(\sigma_{e_3}^2)^2} + \Delta, \; \forall \Delta > 0$.

Thus:

$$W_o = \begin{bmatrix} c_1 & c_2 & -\frac{c_2\sigma_{e_2e_3}}{\sigma^2_{e_3}} \\ c_2 & c_3 & -\frac{c_3\sigma_{e_2e_3}}{\sigma^2_{e_3}} \\ -\frac{c_2\sigma_{e_2e_3}}{\sigma^2_{e_3}} & -\frac{c_3\sigma_{e_2e_3}}{\sigma^2_{e_3}} & \frac{c_3\sigma^2_{e_2e_3}}{(\sigma^2_{e_3})^2}+\Delta \end{bmatrix} \tag{5.4.31}$$

Therefore:

$$L_w = (A^T W_o A)^{-1} A^T W_o = L_o = \begin{bmatrix} \frac{1}{a_1} & 0 & 0 \\ 0 & \frac{1}{a_2} & -\frac{\sigma_{e_2e_3}}{a_2\sigma^2_{e_3}} \end{bmatrix} \tag{5.4.32}$$

Notice that L_w in equation (5.4.32) is exactly the BLUE L_o of equation (5.4.21). Thus, for any choice of positive definite matrix $C \in \mathbb{R}^{2x2}$, our optimal weight matrix W_o leads to the BLUE L_o.

5.5 Appendix

This section contains the proofs for lemmas 5.1, 5.2, 5.3, and 5.4.

Proof of Lemma 5.1: If condition (ii) of section 5.1.1 holds,
then: $LAx_p = Lb_p = x_p \Rightarrow LA = I$. If $LA=I$, then $Lb_p = LAx_p = x_p \Rightarrow$
condition (ii) of section 5.1.1 holds.

Proof of Lemma 5.2:
Our goal is to find an estimator of x_p that is linear and unbiased. For our estimator $\hat{x}$ to be unbiased, we require that:

$$\mathrm{E}\{\hat{x} - x_p\} = 0 \tag{5.5.1}$$

which leads to:

$$\mathrm{E}\{Lb_p + Le - x_p\} = \mathrm{E}\{LAx_p + Le - x_p\} = (LA - I)x_p = 0 \tag{5.5.2}$$

and thus we require that:

$$LA = I \tag{5.5.3}$$

Proof of Lemma 5.3: The covariance matrix of $\hat{x}$ can be rewritten as:

$$\begin{aligned}\Lambda_{\hat{x}} = \mathrm{E}\{(\hat{x} - x_p)(\hat{x} - x_p)^T\} = \mathrm{E}\{(Lb_p + Le - x_p)(Lb_p + Le - x_p)^T\} = \\ \mathrm{E}\{[(LA - I)x_p + Le][(LA - I)x_p + Le]^T\} = L\Lambda_e L^T\end{aligned} \tag{5.5.4}$$

where the last equality follows from the constraint $LA=I$.

Proof of Lemma 5.4:

1) Since the columns of $A \in \mathbb{R}^{mxf}$ are independent and Λ_e^{-1} is positive definite, $A^T\Lambda_e^{-1}A$ is invertible and thus, $L_o A = (A^T\Lambda_e^{-1}A)^{-1}A^T\Lambda_e^{-1}A = I$.

2) We are trying to show that any L≠ L_o leads to the matrix $(\Lambda_y - \Lambda_{\hat{x}})$ being positive semidefinite. Let's write any L as follows:

$$L = L_o + (L - L_o) \tag{5.5.5}$$

Substituting this into $L\Lambda_e L^T$ gives:

$$L\Lambda_e L^T = L_o\Lambda_e L_o^T + (L - L_o)\Lambda_e L_o^T + L_o\Lambda_e(L - L_o)^T + (L - L_o)\Lambda_e(L - L_o)^T \tag{5.5.6}$$

Now, $(L-L_o)\Lambda_e L_o^T = (L-L_o)\Lambda_e\Lambda_e^{-1}A(A^T\Lambda_e^{-1}A)^{-1} = 0$ because $\Lambda_e\Lambda_e^{-1} = I$ and $(L-L_o)A = I-I = 0$ for all L that satisfy the constraint $LA=I$. Moreover, $L_o\Lambda_e(L-L_o)^T = 0$ as it is just the transpose of $(L-L_o)\Lambda_e L_o^T$. Thus, we are left with:

$$L\Lambda_e L^T = L_o\Lambda_e L_o^T + (L-L_o)\Lambda_e(L-L_o)^T \tag{5.5.7}$$

which is minimized when $L=L_o$. Moreover, notice that if $L \neq L_o$, $L\Lambda_e L^T \neq L_o\Lambda_e L_o^T$ and thus, $(\Lambda_y - \Lambda_{\hat{x}})$ is nonzero. We have thus shown that L_o is the unique minimizer of $\Lambda_{\hat{x}}$. Plugging in $L=L_o$ in equation (5.5.7) gives us:

$$L\Lambda_e L^T = L_o\Lambda_e L_o^T = (A^T\Lambda_e^{-1}A)^{-1} \tag{5.5.8}$$

and thus, $\Lambda_{\hat{x}} = (A^T\Lambda_e^{-1}A)^{-1}$.

Chapter 6

A Likelihood Model for the Receptive Fields of ON/OFF Cells

6.0 Acknowledgements and Introduction

Before beginning this chapter, I would like to acknowledge the help that my colleagues gave me in developing the likelihood models described. Professor John L. Wyatt came up with the idea of trying to estimate speed and angle via a likelihood approach and he gave us the idea of creating a cell receptive field model which resembles a 2-D Gaussian; moreover, he was always there to help with any problems that we had when trying to figure out the specifics of the model and implement it. Jessica Wu and Shamim Nemati collaborated with me in making the model work and implementing it. Shamim played an important role in the numerical optimization of our model and Jessica was responsible for obtaining the initial estimates of our model parameters. Moreover, Shamim and Jessica coded a big chunk of the model implementation. Both Jessica and Shamim also played an important role in dealing with the experimental data. As this effort was extremely collaborative, it is hard to specifically say who did what exactly; however, their work was extremely important for the development of this model.

The previous chapters have presented algorithms which estimate the speed and angle of moving edges of light (curtains and bars). The model used in those algorithms approximated each ON or OFF cell as a "sensor" which "fired" when an edge passed over its center. In essence, these cells formed a sensor network in which each cell gave information about when the edge passed over its center. In its simplicity, this model did not account for the fact that ON and OFF cells have non-negligible receptive fields which, when stimulated, produce firing patterns. Moreover, the model used in the

previous chapters assumed that there is no time lag between the stimulation of a cell and its response to the stimulation. Furthermore, the model did not take into account firing pattern differences between the cells used in the estimation algorithms. As described in chapter 6 of Eisenman 2007 [1], the only information used in determining a cell's firing time in response to a moving edge was the mean of its response spike train. With this method, the time dynamics of a response spike train were lost.

In this chapter, we develop a likelihood based receptive field model for ON and OFF cells. This statistical model for the receptive field is defined by representing the spatial dependence of the cell firing on the location of the edge as an inhomogeneous Poisson process. Here, the rate parameter of the inhomogeneous Poisson process is modeled as a function of the edge's position with respect to the center of the cell. Section 6.1 describes the model details; section 6.2 describes how we obtain the model parameters for each cell from real data via a maximum likelihood approach (this will be termed the "forward problem"); section 6.3 describes how we estimate the speed and angle of a moving edge via a maximum likelihood approach from a group of cell responses (this will be termed the "inverse problem"). Figure 6-1 depicts a flowchart of the steps taken in both the "forward" and "inverse" problems.

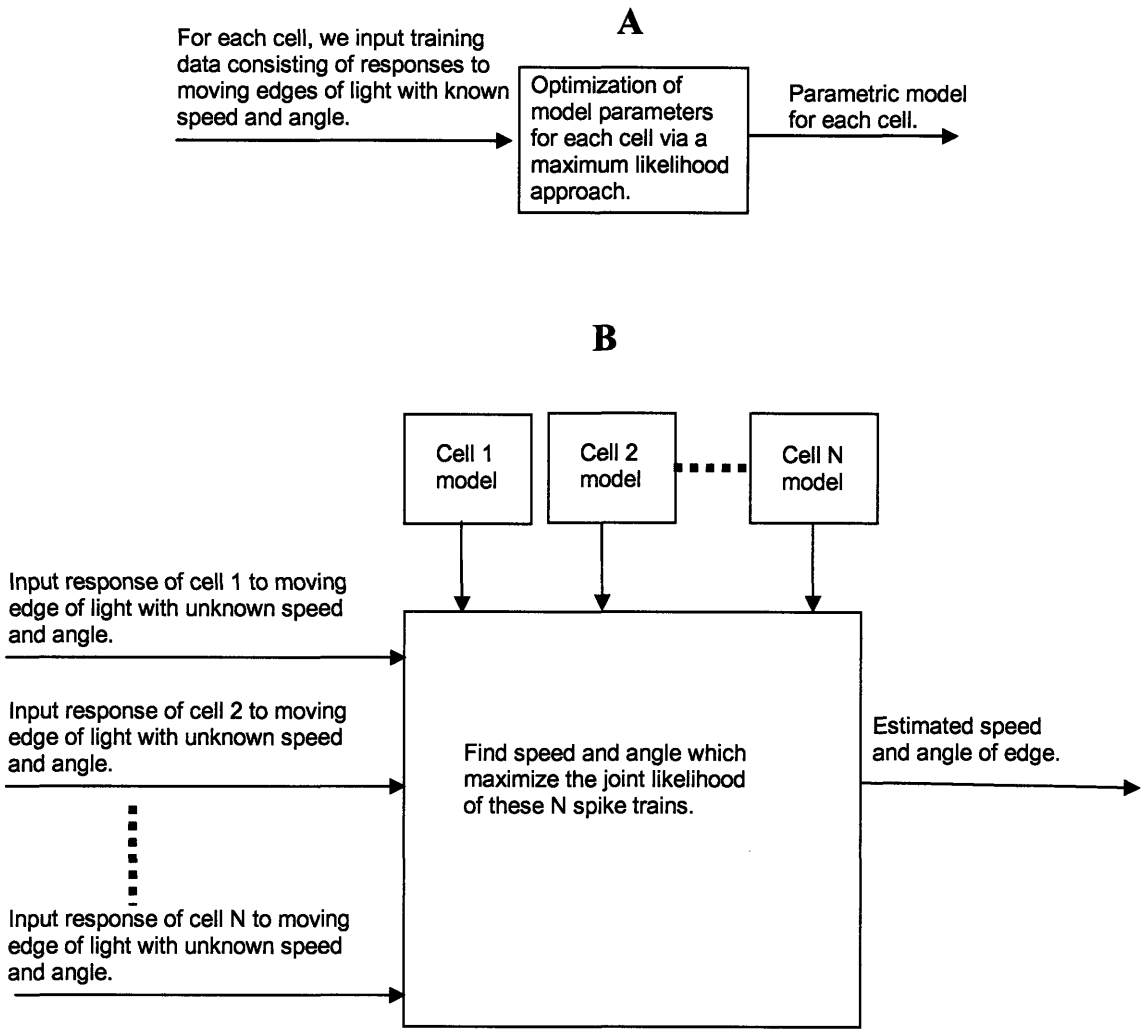

Figure 6-1: Flow chart representation of the sequence of steps in the "forward" problem (panel A) and in the "inverse" problem (panel B). In the "forward" problem, we train each cell's firing rate model by using its responses to edges with known speeds and angles; this is done with a maximum likelihood approach. In the "inverse" problem, we maximize the joint likelihood of all cell responses to an edge whose speed and angle we are trying to estimate; the (V^*, θ^*) pair which maximizes this joint likelihood is our estimate of the edge's speed and angle.

6.1 Likelihood Model Description

6.1.1 Inhomogeneous Poisson Firing Rate Model

We model the receptive field of an ON (OFF) cell in response to a moving bright (dark) 1-D edge as a 2-dimensional Gaussian-shaped sensitivity function. We then assume that the spike trains generated by an ON or OFF cell can be modeled as an inhomogeneous Poisson process whose rate $\lambda(t)$ depends on the integral of the sensitivity function over the 1-D edge in the plane; we also assume that the cell has a constant background firing rate. To be concrete, let us assume we have a cell located at the coordinates $m = (m_x, m_y)$ with a 2-D Gaussian receptive field of standard deviation σ_x in the x-direction and standard deviation σ_y in the y-direction. Moreover, let $W = \begin{bmatrix} \frac{1}{\sigma_x^2} & 0 \\ 0 & \frac{1}{\sigma_y^2} \end{bmatrix}$ be the inverse of the covariance matrix of this 2-D Gaussian sensitivity function; note that for simplicity, we assume that the covariance is zero. Then,

$$\lambda(t) \triangleq c + \int_{l(t)} \frac{Ke^{-\frac{(\mathbf{z}-\mathbf{m})^T W(\mathbf{z}-\mathbf{m})}{2}}}{2\pi\sigma_x\sigma_y} dl \qquad (6.1.1)$$

where c is the constant background firing rate of the cell and K is a measure of how vigorously the cell fires and is a property of the cell that is not stimulus dependent. Thus, $\lambda(t)$ is the integral of the 2-D sensitivity function over the 1-D edge. Note that $l(t)$ is the equation of the edge boundary as a function of time and that $\mathbf{z} = (z_1, z_2)$ contains the coordinates of any point in the $z_1 - z_2$ plane. Also note that the units of $\lambda(t)$ are spikes/second.

This model is specific for moving edge stimuli; for any moving 1-D edge stimulus, we can integrate the Gaussian sensitivity function along that edge in order to

determine the firing rate. A graphical representation of this setup is portrayed in figure 6-2. Let us see in detail how equation (6.1.1) is used. We are interested in characterizing a cell's response to a moving edge of light; note that these stimuli are essentially a continuum of moving points arranged in a straight line. We assume that the spike train generated by a cell in response to a moving edge of light over its receptive field is an inhomogeneous Poisson process with rate $\lambda(t)$; this $\lambda(t)$ can be extracted by integrating the 2-D Gaussian receptive field model of (6.1.1) over the edge.

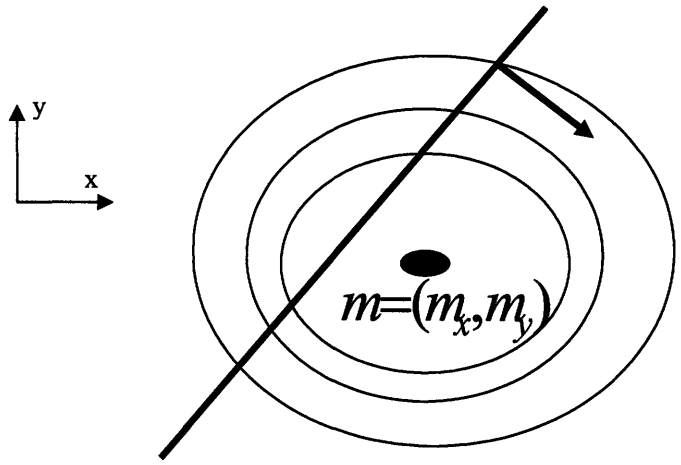

Figure 6-2: This figure graphically depicts a cell's 2-D Gaussian receptive field over which a 1-D edge is moving. The point $m = (m_x, m_y)$ is the center of the receptive field of the cell. Along each contour, the height is constant.

We first derive the result for an edge with a velocity vector normal to the edge having an angle of $\theta=0°$ (note that this is the same convention as that used in the previous chapters) and then generalize it to an edge moving at any angle θ. For the case of an edge moving at $\theta=0°$, we integrate along the y-direction as shown in figure 6-3. Thus,

$$\lambda(t) \triangleq c + \int_{-\infty}^{\infty} \frac{Ke^{-\frac{(\mathbf{z}-\mathbf{m})^T W(\mathbf{z}-\mathbf{m})}{2}}}{2\pi\sigma_x\sigma_y} dz = c + \frac{Ke^{-\frac{(x(t)-m_x)^2}{2\sigma_x^2}}}{\sqrt{2\pi}\sigma_x} \qquad (6.1.2)$$

where $x(t)$ is the x-coordinate of the edge location at time t.

For the case of an edge moving at any angle θ, we rotate the *coordinate system* by θ as shown in figure 6-4; in this new u-v coordinate system, the edge is moving horizontally and we thus integrate out the variable v.

Before continuing, a review of coordinate rotation is presented. Let as assume that we want to rotate our current coordinate system x-y by a counterclockwise angle θ (we will call this rotated coordinate system u-v). Note that the coordinates of any fixed vector z in the rotated coordinate system (i.e. the u-v coordinate system) are now given by a rotation matrix which is the matrix transpose of the fixed-axis rotation matrix and, as can be seen in figure 6-4, is equivalent to rotating the *vector* z by an angle of $-\theta$ (this means a clockwise rotation of θ) relative to a fixed set of axes. This is done by applying the rotation matrix R:

$$R \triangleq \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{6.1.3}$$

There are two ways to view this R: 1) it rotates the coordinate system by θ. 2) it rotates every vector of the old coordinate system (i.e. the x-y coordinate system) by $-\theta$. Note that $R^{-1} = R^T$ and $\det(R) = 1$. Thus, the relationship between the new coordinates (u,v) and the old coordinates (x,y) is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = R \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.1.4}$$

We thus substitute the following into equation (6.1.2):

$$\mathbf{z} - \mathbf{m} = R^{-1} \begin{bmatrix} u(t) - m_u \\ v(t) - m_v \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} u(t) - m_u \\ v(t) - m_v \end{bmatrix} \tag{6.1.5}$$

where the point $\bar{m} = (m_u, m_v)$ is the representation of the center of the cell's receptive field in the u-v coordinate system and $\mathbf{w}(t) = [u(t) - m_u \ , \ v(t) - m_v]$ is the representation of the location of the edge in the u-v coordinate system at time t. Note that the relationship between the cell center $\bar{m} = (m_u, m_v)$ in the u-v coordinate system and the cell center $m = (m_x, m_y)$ in the x-y coordinate system is the following:

$$\begin{bmatrix} m_u \\ m_v \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \end{bmatrix} \tag{6.1.6}$$

Thus, in the u-v coordinate system, the argument of the integral in equation (6.1.2) can be rewritten as follows:

$$\frac{Ke^{-\frac{(\mathbf{z}(t)-\mathbf{m})^T W(\mathbf{z}(t)-\mathbf{m})}{2}}}{2\pi\sigma_x\sigma_y} = \frac{Ke^{-\frac{[R^{-1}(\mathbf{w}(t)-\bar{\mathbf{m}})]^T W[R^{-1}(\mathbf{w}(t)-\bar{\mathbf{m}})]}{2}}}{2\pi\sigma_x\sigma_y} \tag{6.1.7}$$

and we can now integrate equation (6.1.7) over the variable v in the u-v coordinate system.

Thus, the inhomogeneous Poisson rate for a given angle θ now becomes:

$$\lambda(t;\theta) = c + \int_{-\infty}^{\infty} \frac{Ke^{-\frac{[R^{-1}(\mathbf{w}(t)-\bar{\mathbf{m}})]^T W[R^{-1}(\mathbf{w}(t)-\bar{\mathbf{m}})]}{2}}}{2\pi\sigma_x\sigma_y} dv = c + \frac{Ke^{-\frac{(u(t)-m_u)^2}{2(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}}}{\sqrt{2\pi(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}} \tag{6.1.8}$$

where $u(t)-m_u$ is the distance between the center of the cell and the horizontally moving (in the u-v coordinates) edge. Proving that equation (6.1.8) holds is best done by a geometric argument. After we have rotated the coordinates, we are now in the u-v coordinate system; the layout is portrayed in figure 6-5. We are interested in integrating out v: thus, we are integrating the 2-D Gaussian over v in the u-v coordinate system. The only difference between this case and the $\theta=0°$ case of equation (6.1.2) is that the "standard deviation" of the 2-D Gaussian along the v-axis is different. In fact, the "standard deviation" along the v-axis is exactly $\sqrt{(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}$. Thus, by replacing the σ_x^2 in equation (6.1.2) with $\sqrt{(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}$, and keeping in mind that the x-coordinate in (6.1.2) is now the u-coordinate, we get the result of equation (6.1.8).

Let V be the speed of the moving edge. If we define t_o as the time when the curtain passes the origin, the distance $|u(t)-m_u|$ is equal to $|(t-t_o)V-m_u|$. Thus, equation (6.1.8) can be rewritten as:

$$\lambda(t;V,\theta) = c + \frac{Ke^{\frac{\{V(t-t_o)-(\cos(\theta)m_x+\sin(\theta)m_y)\}^2}{2(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}}}{\sqrt{2\pi(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}} \tag{6.1.9}$$

Equation (6.1.9) gives us the instantaneous inhomogeneous Poisson firing rate of the cell's spike train in response to an edge of constant velocity moving at an angle θ across its receptive field. Let's define the start time of the edge's motion as 0 and the stop time of its motion as T (i.e. the total time of movement of the edge from one end of the screen to the other is T). Then, for a given angle of movement θ and speed V, we have the instantaneous inhomogeneous Poisson firing rate $\lambda(t;V,\theta)$ at all times $t \in [0,T]$.

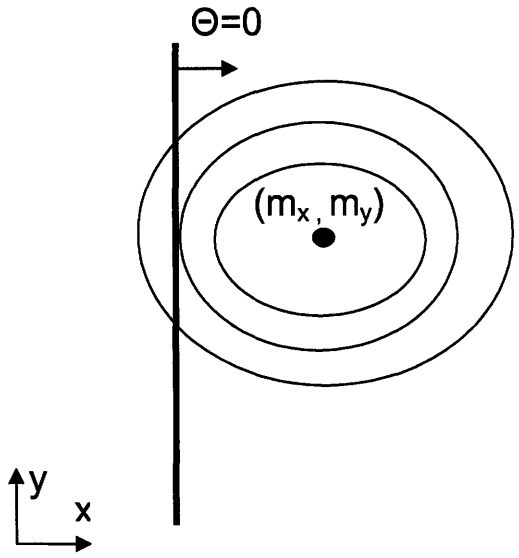

Figure 6-3: This figure depicts an edge moving horizontally. In order to get the firing rate $\lambda(t)$ in response to this edge, we must integrate the 2-D Gaussian along the y-direction.

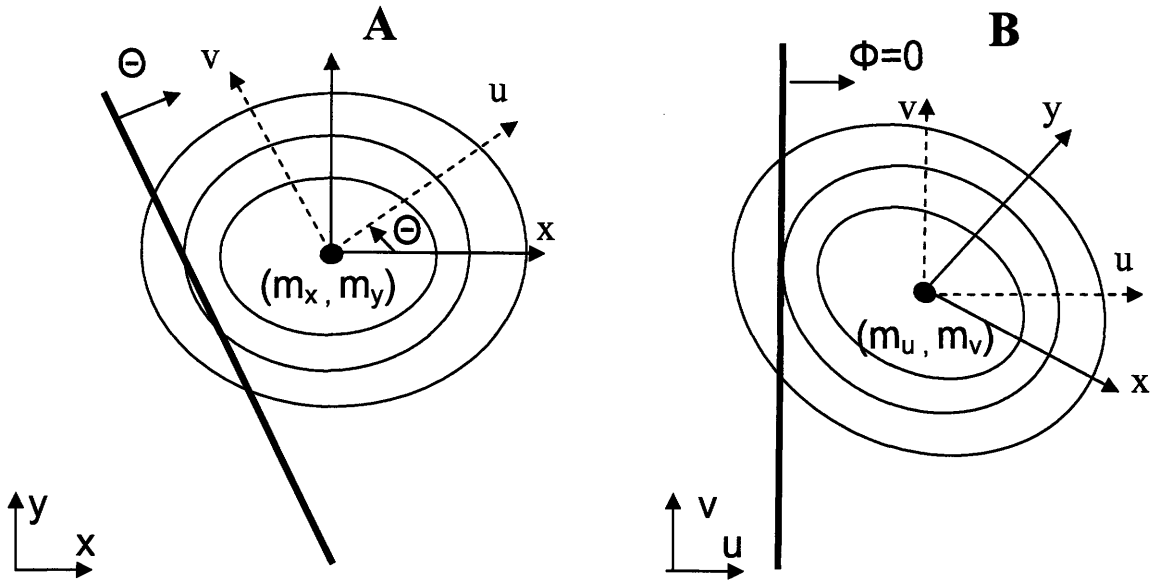

Figure 6.4: When plotted in u-v coordinates, the picture in panel A takes on the appearance in panel B. The angle of motion in the u-v coordinate system is Φ. Note that rotating the x-y coordinate system by θ and rotating a vector in the x-y coordinate system by $-\theta$ are equivalent.

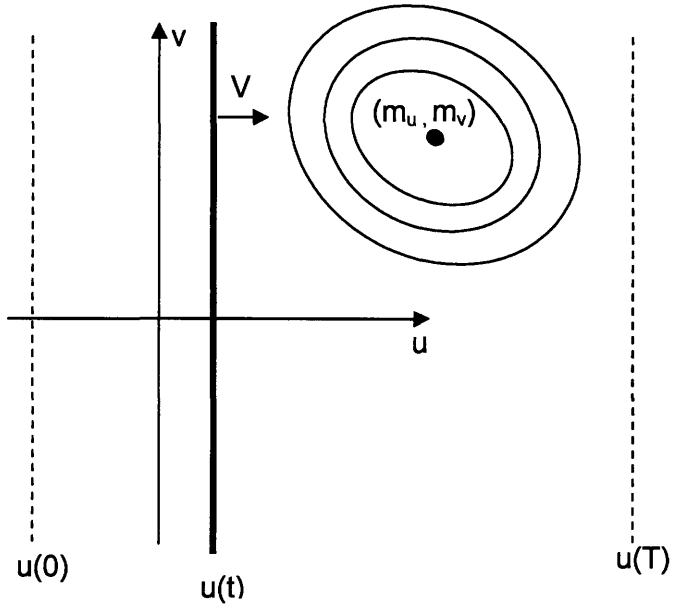

Figure 6-5: In the u-v coordinate system, the edge appears to move with a constant velocity V horizontally. Its location at each time t is given by *u(t)*. If we define t_o as the time which the curtain passes through the origin, the distance $|u(t) - m_u|$ is equal to $|(t - t_o)V - m_u|$. The start time of the edge motion is defined as *t=0* (i.e. when it enters the screen from the left) and the end time of the edge motion is at *t=T* (i.e. when it exits the screen from the right).

6.1.2 The Likelihood of a Spike Train

In this section, we present the method for obtaining the likelihood of a spike train which is a realization of an inhomogeneous Poisson process. The likelihood of a spike train is essentially the probability of observing a spike train given the parameters of the model which generated it. Viewed from a different perspective, the likelihood of a spike train is equivalent to the probability of observing a sequence of inter-spike intervals (ISI's) given the parameters of the model which generated them. The procedure for calculating the likelihood of a spike train is best illustrated through an example. Consider the spike train shown in figure 6-6. The time interval of interest is of length T and there are two spikes at times t_1 and t_2. The firing rate for the Poisson process is $\lambda(t)$. One method of deriving the likelihood of the spike train is given below; an alternative is given in the appendix to this chapter.

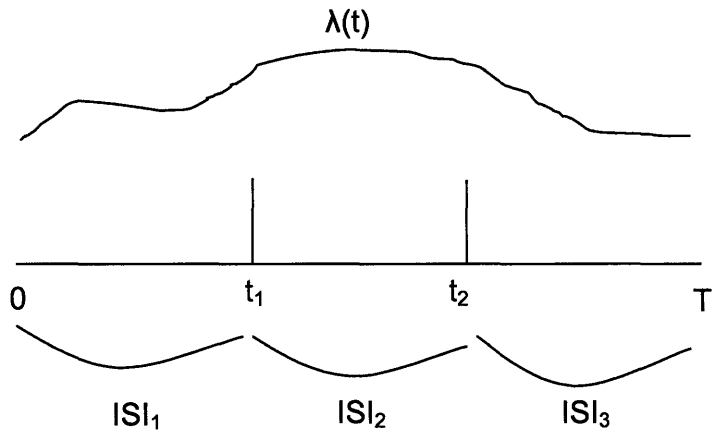

Figure 6-6: A hypothetical spike train is depicted with two spikes occurring at times t_1 and t_2. Each ISI is an inter-spike interval. The inhomogeneous Poisson firing rate which generates the spike train is $\lambda(t)$.

The likelihood of this spike train is defined as the joint probability density (i.e. it is a pdf) of the following 3 events: a) inter-spike interval 1 is of length t_1, b) inter-spike interval 2 is of length $t_2 - t_1$, and c) no spikes occur between t_2 and T. From Bayes rule, we have the following (note that there is a slight abuse of notation):

$$f\{ISI_1 = t_1 \cap ISI_2 = t_2 - t_1 \cap ISI_3 > T - t_2\} = P\{ISI_3 > T - t_2 \mid ISI_1 = t_1 \cap ISI_2 = t_2 - t_1\} f\{ISI_2 = t_2 - t_1 \mid ISI_1 = t_1\} f\{ISI_1 = t_1\} \quad (6.1.10)$$

where $f\{.\}$ denotes a probability density function and $P\{.\}$ denotes a probability. Let us evaluate $f\{ISI_1 = t_1\}$. Our objective is to find the cumulative density function $F_{ISI_1}\{t_1\}$; then, $f\{ISI_1 = t_1\} = \frac{dF_{ISI_1}\{t_1\}}{dt_1}$. We know that:

$$F_{ISI_1}\{t_1\} \triangleq P\{t \le t_1\} = 1 - P\{t > t_1\} = 1 - P\{0 \text{ spikes in the interval } [0,t_1]\} \quad (6.1.11)$$

and since the process which generate these spikes is an inhomogeneous Poisson process:

$$P\{0 \text{ spikes in the interval } [0,t_1]\} = e^{-\int_0^{t_1} \lambda(t)dt} \quad (6.1.12)$$

Thus,

$$f\{ISI_1 = t_1\} = \frac{dF_{ISI_1}\{t_1\}}{dt_1} = \frac{d(1 - P\{0 \text{ spikes in the interval } [0,t_1]\})}{dt_1} = \frac{d(1 - e^{-\int_0^{t_1} \lambda(t)dt})}{dt_1} = \lambda(t_1) e^{-\int_0^{t_1} \lambda(t)dt} \quad (6.1.13)$$

By a similar argument:

$$f\{ISI_2 = t_2 - t_1 \mid ISI_1 = t_1\} = \lambda(t_2) e^{-\int_{t_1}^{t_2} \lambda(t)dt} \quad (6.1.14)$$

Moreover, since the spike train is generated by an inhomogeneous Poisson process:

$$P\{ISI_3 > T - t_2 \mid ISI_1 = t_1 \cap ISI_2 = t_2 - t_1\} = e^{-\int_{t_2}^{T} \lambda(t)dt} \quad (6.1.15)$$

Thus, substituting the results of equations (6.1.13), (6.1.14), and (6.1.15) into equation (6.1.10) yields:

$$f\{ISI_1 = t_1 \cap ISI_2 = t_2 - t_1 \cap ISI_3 \ge T - t_2\} = \lambda(t_1)\lambda(t_2) e^{-\int_0^{T} \lambda(t)dt} \quad (6.1.16)$$

This result can easily be generalized for a spike train of length T consisting of N spikes which occur at times $t_1, t_2, ...t_N$ where $t_1 < t_2 < ..... < t_N < T$. The likelihood of such a spike train is:

$$Likelihood = \lambda(t_1)\lambda(t_2)...\lambda(t_N)e^{-\int_0^T \lambda(t)dt} \tag{6.1.17}$$

Note that if the rate $\lambda(t)$ were constant, equation (6.1.17) would become:

$$Likelihood = \lambda^N e^{-\lambda T} \tag{6.1.18}$$

which is the well known result for homogeneous Poisson processes.

6.1.3 Estimating the Lag in the Time Response of Each Cell

It is well known that there is a time lag between the time at which an edge passes over a cell's receptive field and the time at which the cell responds. This is a characteristic of the cell we wish to model in order for our likelihood model to be an accurate representation of reality. We assume that lag is a characteristic of the cell and is not stimulus dependent. Thus, the lag of a cell will be independent of the direction and speed of a moving edge. Consider the situation below in figure 6-7; let us assume that the cell fires only when the edge is passing over its receptive field. Let $\mathbf{t} = [t_1, t_2, ..., t_N]$ be the vector of spike times and assume $t_1 < t_2 <, ..., < t_N$. Moreover, at $x=0$, we define $t=0$. Since the speed V is constant, there is a 1-1 correspondence between time and space. Thus, $x=Vt$ and we can represent the cell's receptive field length in space or time coordinates. In both panels of figure 6-7, an edge is moving towards the right; the spikes are plotted at the points in time where the edge was located at the moment that the cell fired. If the cell has no lag in its response (panel A), each spike time occurs when the edge is over the cell's receptive field; no spikes occur after the edge has exited the receptive field. If the cell has a time lag Δ in its response (panel B), the spike train is

shifted to the right by Δ; in other words, each spike is delayed and thus, some spike times occur after the edge has exited the receptive field of the cell.

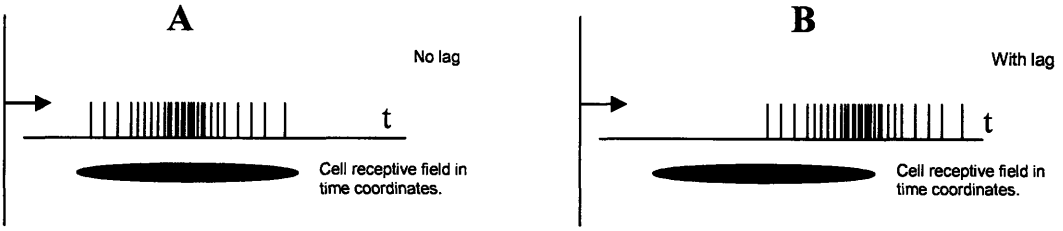

Figure 6-7: In panel A, each spike time occurs when the edge is over the cell's receptive field; no spikes occur after the edge has exited the receptive field. In panel B, the spike train is shifted to the right by Δ; in other words, each spike is delayed and thus, some spike times occur after the edge has exited the receptive field of the cell.

Since V is constant, we can view each spike train as a function of space; in particular, each spike event occurs at a certain location along the axis of movement of the edge. This point is illustrated in figure 6-8; the edge is moving towards the left (θ=180°) with a constant speed V. Panel A of figure 6-8 depicts the spike train as a function of time. Panel B of figure 6-8 depicts the spike train as a function of space; notice that the x-axis has been scaled by the speed V. If the spike train is a sequence of times **t**, the spike train as a function of space is a sequence of locations $\mathbf{x} \triangleq V(T-\mathbf{t})=[V(T-t_1), V(T-t_2),....., V(T-t_N)]$, where $V(T-t_1) > V(T-t_2) > .... > V(T-t_N)$ since $t_1 < t_2 <,...,< t_N < T$ (i.e. the spike times have been "flipped" and "scaled"). Thus, the rightmost spike from the spike train in panel A corresponds to the left most spike of the spike train in panel B (i.e. t_N corresponds to $V(T-t_N)$); analogously, the left most spike of the spike train in panel A corresponds to the right most spike of the spike train in panel B (i.e. t_1 corresponds to $V(T-t_1)$). The lag in space of the spike train in panel B of figure 6-8 is $V\Delta$.

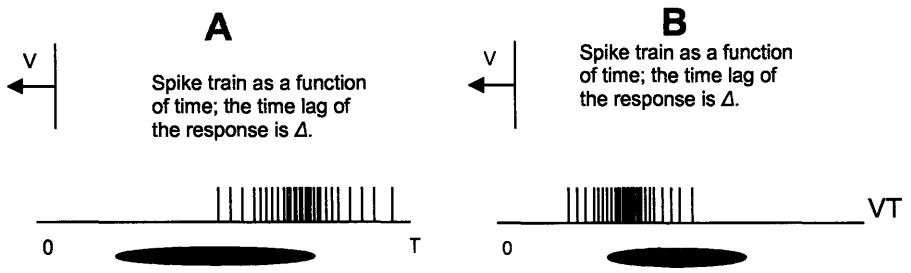

Figure 6-8: The edge is moving with a velocity V towards the left (θ=180°). In panel A, the spike train is plotted as a function of time. In panel B, the spike train is plotted as a function of space; note that the x-axis on the right has been scaled by V. Each spike from the spike train in panel B corresponds to a location in space.

As our experimental data consists of edges moving in opposite directions (see chapter 3 for details), we are able to estimate the lag of a cell. Since we assume that the cell's lag is independent of the edge direction, we estimate the lag along each axis of motion and then average the resulting lags to obtain the characteristic lag of the cell.

For a given axis of motion, our goal is to determine the cell's lag in that axis of motion. We choose as our lag estimate the delay δ that minimizes the (sample) variance of the *union* of the spike trains in response to edges moving in opposite directions. The cell's lag in that axis of motion is then $\delta/2$. An illustration is given in figure 6-9 for the case where the axis of motion is the x-axis. Let $\mathbf{X}_1$ be the spike train corresponding to the right moving edge and let N_1 be the number of spikes in $\mathbf{X}_1$. Let $\mathbf{X}_2$ be the spike train corresponding to the left moving edge and let N_2 be the number of spikes in $\mathbf{X}_2$. Also, let $\mathbf{X}(\delta) = \{\mathbf{X}_1 \cup (\mathbf{X}_2 - \delta)\}$; note that this is the union of all elements in $\mathbf{X}_1$ with all elements in $\mathbf{X}_2$ shifted by δ. As stated above, our goal is to determine the delay δ that minimizes the variance of $\mathbf{X}(\delta)$. Since $\mathrm{var}\left(\mathbf{X}(\delta)\right) = E(\mathbf{X}(\delta)^2) - \left(E(\mathbf{X}(\delta))\right)^2$. Thus, we must find an expression for var($\mathbf{X}(\delta)$) and take its derivative with respect to δ. We have that:

$$E\left[\mathbf{X}^2\right] = \frac{\sum_{n=1}^{N_1} \mathbf{X}_1^2[n] + \sum_{n=1}^{N_2} \left(\mathbf{X}_2[n] - \delta\right)^2}{N_1 + N_2} = \frac{\sum_{n=1}^{N_1} \mathbf{X}_1^2[n] + \sum_{n=1}^{N_2} \mathbf{X}_2^2[n] - 2\delta \sum_{n=1}^{N_2} \mathbf{X}_2[n] + N_2\delta^2}{N_1 + N_2} \qquad (6.1.19)$$

and:

$$\left(E[\mathbf{X}]\right)^2 = \frac{\left(\left(\sum_{n=1}^{N_1}\mathbf{X}_1[n]\right)^2 + \left(\sum_{n=1}^{N_2}\mathbf{X}_2[n]\right)^2 + N_2^2\delta^2 + 2\sum_{n=1}^{N_1}\mathbf{X}_1[n]\sum_{n=1}^{N_2}\mathbf{X}_2[n] - 2N_2\delta\left(\sum_{n=1}^{N_1}\mathbf{X}_1[n] + \sum_{n=1}^{N_2}\mathbf{X}_2[n]\right)\right)}{\left(N_1+N_2\right)^2} \tag{6.1.20}$$

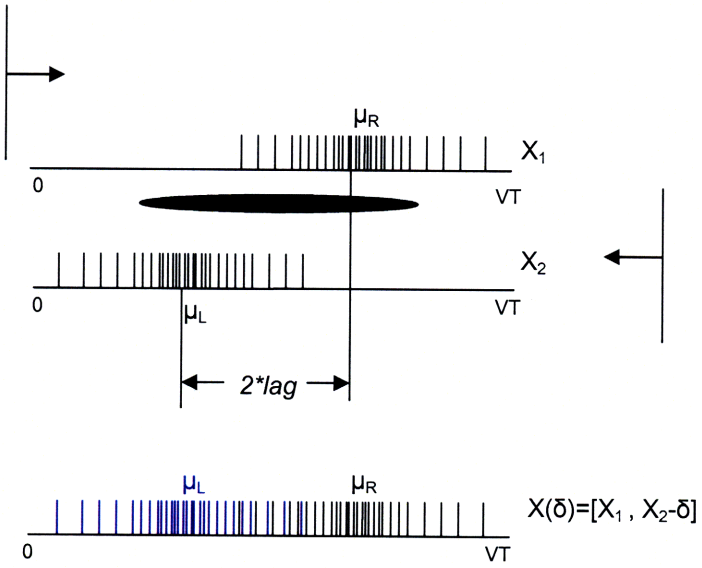

Figure 6-9: The top spike train ($\mathbf{X}_1$) is the response in space of the cell to an edge moving towards the right; μ_R is its mean. The bottom spike train ($\mathbf{X}_2$) is the response in space of the cell to an edge moving towards the left; μ_L is its mean. The union of the two spike trains $\mathbf{X}(\delta)$ contains all of the spikes from spike train $\mathbf{X}_1$ and all of the spike times from spike train $\mathbf{X}_2$ shifted by δ. Our goal is to find the δ that minimizes the variance of $\mathbf{X}(\delta)$. This δ is equal to twice the lag of the cell for this axis of motion.

Thus,

$$\frac{d\,\mathrm{var}\left(\mathbf{X}(\delta)\right)}{d\Delta} = \frac{-2\sum_{n=1}^{N_2}\mathbf{X}_2[n] + 2N_2\delta}{N_1+N_2} - \frac{2N_2^2\delta - 2N_2\left(\sum_{n=1}^{N_1}\mathbf{X}_1[n] + \sum_{n=1}^{N_2}\mathbf{X}_2[n]\right)}{\left(N_1+N_2\right)^2} = 0 \Leftrightarrow$$
$$\delta = \frac{N_1\sum_{n=1}^{N_2}\mathbf{X}_2[n] - N_2\sum_{n=1}^{N_1}\mathbf{X}_1[n]}{N_1N_2} = E[\mathbf{X}_2] - E[\mathbf{X}_1] \tag{6.1.21}$$

Thus, the delay δ that minimizes the (sample) variance of $\mathbf{X}(\delta)$ is equal to $\mu_L - \mu_R$.

Intuitively, this result means that the set X has the minimum variance if we shift the sets $\mathbf{X}_1$ and $\mathbf{X}_2$ so that their means are equal. Since we have assumed that the lag of a cell is

the same for any direction of the stimulus, it must be the case that the lag *in space* of the cell along the x-axis is $\frac{\delta}{2} = \frac{\mu_L - \mu_R}{2}$ and the lag *in time* of the cell is $\Delta = \frac{\delta}{2V} = \frac{\mu_L - \mu_R}{2V}$. Note that this means that all times that we *observe* have this lag; this means that if there were no lag, each spike would have occurred $\left|\frac{\mu_L - \mu_R}{2V}\right|$ seconds earlier. By repeating the above procedure for all axes of motion (for the data presented in chapter 8, we have 4 axes of motion), we get many estimates of the cell's lag; we then average these values to obtain the cell's characteristic lag in response to moving edges.

If we incorporate lag into equation (6.1.9), we get the following expression for $\lambda(t;V,\theta)$:

$$\lambda(t;V,\theta) = c + \frac{Ke^{\frac{\{V(t-|\Delta|-t_o)-(\cos(\theta)m_x+\sin(\theta)m_y)\}^2}{2(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}}}{\sqrt{2\pi(\sigma_x^2\cos^2(\theta)+\sigma_y^2\sin^2(\theta))}} \qquad (6.1.22)$$

6.2 Likelihood Model Optimization

This section presents how we obtain our model parameters for a cell from the responses of that cell to edges moving in 8 directions at a speed of V=714um/sec (this was the data recorded on the experiment date 04/06/2007 as described in section 3.3.3); these directions of motion were $0^0, \pm 45^0, \pm 90^0, \pm 135^0$, and 180^0. We train each cell by presenting it with 2 trials of each possible (V,θ) pair (8 pairs total). Section 6.2.1 contains the derivation of the likelihood function we are trying to optimize; this function is highly nonlinear and thus we proceed with numerical optimization. Section 6.2.2 describes how we obtain initial estimates of the parameters we are trying to optimize over and section 6.2.3 describes the numerical optimization algorithm we use.

6.2.1 Derivation of Likelihood Function

The training data for a cell consists of 16 spike trains. Our goal is to choose parameters for the cell model that maximize the likelihood of observing these sixteen spike trains. Let $L_1, L_2, ..., L_{16}$ denote the likelihood functions of these 16 spike trains. We assume that the likelihood of each spike train is independent from any other spike train. Then, the overall likelihood L of observing these 16 spike trains is:

$$L = \prod_{i=1}^{16} L_i \tag{6.2.1}$$

We are interested in maximizing L with respect to the cell model parameters presented in section 6.1; we denote these by the vector $\mathbf{x}$ where $\mathbf{x} = (\mu_x, \mu_y, \sigma_x, \sigma_y, c, K)$. Maximizing the likelihood function is equivalent to maximizing the natural logarithm of the likelihood function. Thus, our goal is to maximize the "log-likelihood" function with respect to the vector of model parameters $\boldsymbol{x}$:

$$\max_x \{\ln(L)\} = \max_x \left\{ \sum_{i=1}^{16} \ln(L_i) \right\} \tag{6.2.2}$$

Before continuing, the notation that will be used for the rest of this chapter is presented. The total number of spikes in spike train i is denoted by N_i. The k^{th} spike time of spike train i is denoted by $t_{i,k}$.

From equation (6.1.17), the likelihood L_i of spike train i is:

$$\ln(L_i) = \ln\left\{\lambda(t_{i,1};V,\theta_i)\lambda(t_{i,2};V,\theta_i)...\lambda(t_{i,N_i};V,\theta_i)e^{-\int_0^T \lambda(t;V,\theta_i)dt}\right\} \tag{6.2.3}$$
$$= \sum_{k=1}^{N_i} \ln\left\{\lambda(t_{i,k};V,\theta_i)\right\} - \int_0^T \lambda(t;V,\theta_i)dt$$

As portrayed in figure 3-3 of the methods section, the cells we recorded from lay within a square MEA of side 1.4mm. The projection area had a height of 2.718mm and a width of 2.038mm; moreover, the projection area and the MEA shared the same center. Thus, at $t=0$ and $t=T$, the distance of the edge to any cell center was at least 314um; as the typical diameter of RGC's is 300-400um, it is safe to assume that at the times $t=0$ and $t=T$, the edge is not stimulating any part of the receptive field of any cell from which we recorded from. Thus, we can rewrite the second term of equation (6.2.3) as follows:

$$\int_0^T \lambda(t;V,\theta_i)dt = \int_{t=0}^{t=T} c + \frac{Ke^{-\frac{\{V(t-|\Delta|-t_o)-(\cos(\theta_i)\mu_x+\sin(\theta_i)\mu_y)\}^2}{2(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}}{\sqrt{2\pi(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}dt =$$

$$cT + \int_{g=0}^{g=\frac{T}{V}} \frac{Ke^{-\frac{\{g-|\Delta|-t_o-(\cos(\theta_i)\mu_x+\sin(\theta_i)\mu_y)\}^2}{2(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}}{V\sqrt{2\pi(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}dg \approx \tag{6.2.4}$$

$$cT + \int_{-\infty}^{\infty} \frac{Ke^{-\frac{\{g-|\Delta|-t_o-(\cos(\theta_i)\mu_x+\sin(\theta_i)\mu_y)\}^2}{2(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}}{V\sqrt{2\pi(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}dg = cT + \frac{K}{V}$$

Thus, we can rewrite (6.2.3) as:

$$\ln(L_i) = \sum_{k=1}^{N_i} \ln\left\{\lambda(t_{i,k};V,\theta_i)\right\} - \left(cT + \frac{K}{V}\right) \tag{6.2.5}$$

Therefore, our objective stated in equation (6.2.2) can now be posed as:

$$\max_{x}\{\ln(L)\} = \max_{x}\left\{\sum_{i=1}^{16}\left\{\left(\sum_{k=1}^{N_i}\ln\left\{\lambda(t_{i,k};V,\theta_i)\right\}\right)-\left(cT+\frac{K}{V}\right)\right\}\right\} \tag{6.2.6}$$

where:

$$\ln\left\{\lambda(t_{i,k};V,\theta_i)\right\} = \ln\left\{c+\frac{Ke^{\frac{\{V(t_{i,k}-|\Delta|-t_o)-(\cos(\theta_i)\mu_x+\sin(\theta_i)\mu_y)\}^2}{2(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}}{\sqrt{2\pi(\sigma_x^2\cos^2(\theta_i)+\sigma_y^2\sin^2(\theta_i))}}\right\} \tag{6.2.7}$$

Clearly, maximizing the log-likelihood over $\mathbf{x}=(\mu_x,\mu_y,\sigma_x,\sigma_y,c,K)$ is a highly nonlinear problem; thus, we must resort to a numerical optimization procedure. The method we use is called *Quasi-Newton* optimization and is presented in section 6.2.3. This optimization method requires initial estimates of $\mathbf{x}=(\mu_x,\mu_y,\sigma_x,\sigma_y,c,K)$ which are "close" to their true values.

6.2.2 Obtaining Initial Estimates of a Cell's Likelihood Model Parameters

We seek to obtain initial estimates of the model parameters $\mathbf{x}=(\mu_x,\mu_y,\sigma_x,\sigma_y,c,K)$ for a cell. The reason for this is that we wish to perform numerical optimization in order to maximize the joint likelihood and thus, we need good initial estimates of **x.** Recall that μ_x and μ_y are the coordinates of the cell center, and σ_x and σ_y are the standard deviations of the cell's receptive field along the x-axis and y-axis respectively; moreover, c is the cell's spontaneous firing rate (we assume it to be constant) and K is a measure how vigorously the cell fires in response to a moving edge; a more detailed description of what K is and how it can be estimated is given later in this section.

Figure 6-10 depicts the PSTH as a function of space of a cell's response to an OFF edge (i.e. a light to dark transition) moving at 0°, 180°, -90°, and 90°. For each row (i.e. edge direction) in this figure, the firing pattern of the cell is shown as a function of spatial location of the edge at the moment each spike occurred. For each row, the entire field is bright at the far left and dark at the far right. For each of the four responses, we define as the *main response* every point within three standard deviations of the mean; the

background signal is defined as every point farther than three standard deviations from the mean. The *main response* is the cell's response to the moving stimulus; the *background signal* reflects the cell's spontaneous firing activity and does not depend on the presence of a stimulus.

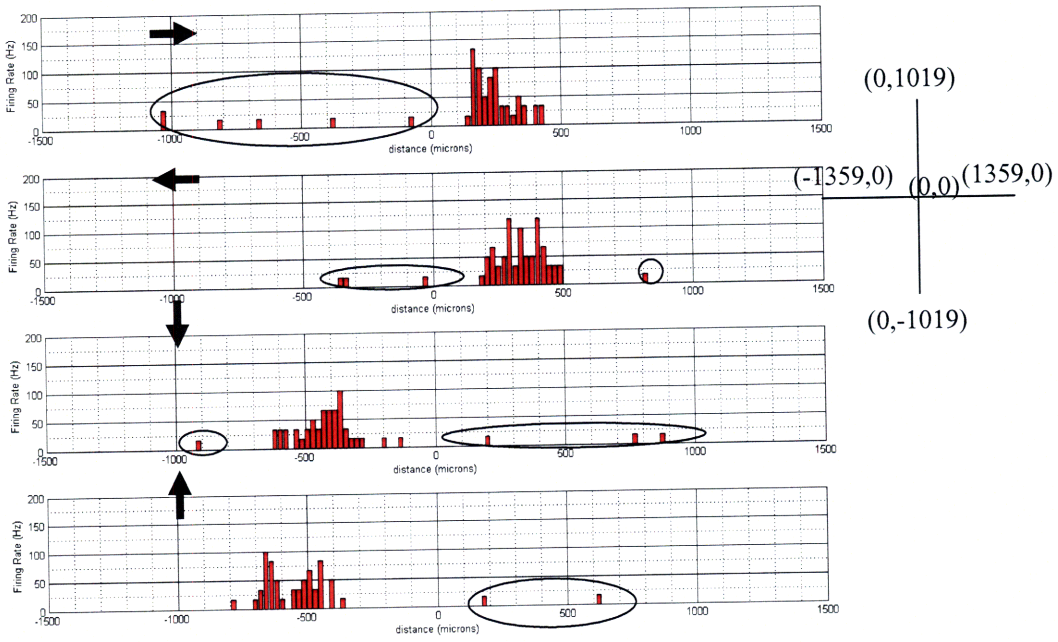

Figure 6-10: This figure depicts a cell's response (PSTH) as a function of space to edges moving rightward (top row), leftward (second row), downward (third row), and upward (fourth row). The axis on the top right corner shows the coordinate system we are using; the center of the projection area is defined as the point (0,0). The projection area has a height of 2038um and a width of 2718um. The spikes that have been circled are the "outliers" used to get an initial estimate for *c*.

The background firing rate *c* (units are spikes/seconds) is determined by counting the spikes of the *background signal* and dividing by the duration of the *background signal.* Note that this is done for each direction and the results are averaged to obtain the final *c.* The For example, in figure 6-10, for each row, the number of the "outlier" spikes (circled spikes) is divided by the length of the *background signal*; the resulting value (which has units of spikes/meter) is subsequently multiplied by V in order to get *c.* We average the estimates of *c* that we get from each row (direction) to obtain our final estimate of *c.*

The procedure of obtaining initial estimates for μ_x and μ_y is made clear by observing figure 6-10; the first two rows correspond to the responses of the cell to an

edge moving at 0° and 180° respectively. From each row, we omit the outlier spikes: we define as outliers any spikes which lie outside of ±3 standard deviations from the mean. As these plots have accounted for the lag in the cell's response, the initial estimate of μ_x is the average of the means of the first two responses after outliers have been removed. The last two rows of figure 6-10 correspond to the responses of the cell to an edge moving at -90° and 90° respectively; the initial estimate of μ_y is the average of the means of these two responses after outliers have been removed.

The procedure of obtaining initial estimates for σ_x and σ_y is made clear by observing figure 6-10. As the first two rows correspond to the responses of the cell to an edge moving at 0° and 180° respectively, the initial estimate of σ_x is the average of the standard deviations of the first two responses after outliers have been removed. As the last two rows of figure 6-10 correspond to the responses of the cell to an edge moving at -90° and 90° respectively, the initial estimate of σ_y is the average of the means of these two responses after outliers have been removed.

According to our likelihood model of the cell, all spiking activity for the duration of the moving edge stimulus is due to two factors: the spontaneous firing rate of the cell and the cell's response to the moving edge. Thus, for a given stimulus, the total number of spikes over time T (the duration of the stimulus) is:

$$\text{Total spikes for a given direction } \theta_i = \int_0^T \Lambda(t;V,\theta_i)dt = cT + \frac{K}{V} \qquad (6.2.8)$$

Since we have already estimated c, we are in a position to estimate K for each direction of movement of the edge. For each direction, we count the total number of spikes in the cell's response and solve for K. We then average these estimates of K over all directions to get the final estimate of K. Having obtained the initial estimates for $\mathbf{x} = (\mu_x, \mu_y, \sigma_x, \sigma_y, c, K)$, we are now able to start the numerical optimization procedure.

6.2.3 Numerical Optimization of Cell Likelihood Model

In order to numerically optimize the likelihood model parameters for a cell, we use *Quasi-Newton* optimization. Recall that the function which we are trying to maximize is given by equation (6.2.6); maximizing $\ln(L)$ is equivalent to minimizing $-\ln(L)$; in this section, we refer to $-\ln(L)$ as the joint likelihood function. We assume that our initial estimates of the likelihood model parameters are "close" to their true values. We also assume that the joint likelihood function which we are trying to minimize can be approximated by a quadratic near the region of our initial parameter estimates. Our goal is the following:

$$\min_x f(x) = \min_x \frac{1}{2}\mathbf{x}^T H\mathbf{x} + a^T\mathbf{x} + b \tag{6.2.9}$$

where the Hessian matrix H is a positive definite symmetric matrix, a is a constant vector, and b is a constant. The optimal solution $\mathbf{x}^*$ for this problem occurs when the gradient of $\mathbf{x}$ goes to zero:

$$\nabla f(\mathbf{x}^*) = H\mathbf{x}^* + a = 0 \Leftrightarrow \mathbf{x}^* = -H^{-1}a \tag{6.2.10}$$

Newton numerical optimization methods calculate H directly and proceed in a direction of descent to locate the minimum after a number of iterations. However, calculating H and inverting it numerically involves a large amount of computation. The *Quasi-Newton* method avoids this by using the BFGS (Broyden 1970 [3]) rank two update formula for the inverse Hessian matrix (the derivation of the rank two inverse formula is given in the appendix of this chapter). The update formula is:

$$B_{k+1} = B_k + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{q}_k} - \frac{B_k\mathbf{q}_k\mathbf{q}_k^T B_k^T}{\mathbf{q}_k^T B_k^T \mathbf{q}_k} \tag{6.2.11}$$

where $B_k \triangleq H_k^{-1}$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{q}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$. At each major iteration k, the algorithm performs a line search along the direction $\mathbf{d}_k = -H_k^{-1}\nabla f(x_k)$; the line search method searches along the line containing the current point, $\mathbf{x}_k$ parallel to the search direction $\mathbf{d}_k$; that is, the method finds the next iterate $\mathbf{x}_{k+1}$ of the form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a\mathbf{d}_k \tag{6.2.12}$$

where $\mathbf{x}_k$ denotes the current iterate and a is a scalar step length parameter. It should be noted that the Hessian matrix H is always maintained to be positive definite so that the direction of search $\mathbf{d}$ is always in a descent direction. This means that for some arbitrarily small step a in the direction $\mathbf{d}$, the objective function decreases in magnitude. We achieve positive definiteness of H by ensuring that H is initialized to be positive definite and thereafter $\mathbf{q}_k^T\mathbf{s}_k$ is always positive. The term $\mathbf{q}_k^T\mathbf{s}_k$ is a product of the line search step length parameter a_k and a combination of the search direction $\mathbf{d}_k$ with gradient evaluations at times k and $k+1$:

$$\mathbf{q}_k^T\mathbf{s}_k = a_k(\nabla f(\mathbf{x}_{k+1})^T - \nabla f(\mathbf{x}_k)^T)\mathbf{d}_k \tag{6.2.13}$$

One can always achieve the condition that $\mathbf{q}_k^T\mathbf{s}_k$ is positive by performing a sufficiently accurate line search. This is because the search direction $\mathbf{d}_k$ is a descent direction and thus, a_k and $-\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$ are always positive. Thus, the possible negative term $\nabla f(x_{k+1})^T\mathbf{d}_k$ can be made as small in magnitude as required by increasing the accuracy of the line search.

Figure 6-11 depicts the results of the numerical optimization on each of the cells which we used. Each cell's receptive field is portrayed by a 2-D Gaussian sensitivity function of the same form as the one shown in figure 6-2; the square delineates the spatial extent of the MEA which we recorded from. It should be noted that the likelihood model parameters c and K are not plotted as they do not depend on space. It is interesting to observe that the shape of the cells' receptive fields are approximately circular in most cases; thus, the inclusion of a covariance term in the 2-D Gaussian sensitivity model is not expected to significantly affect the results presented in the next chapters.

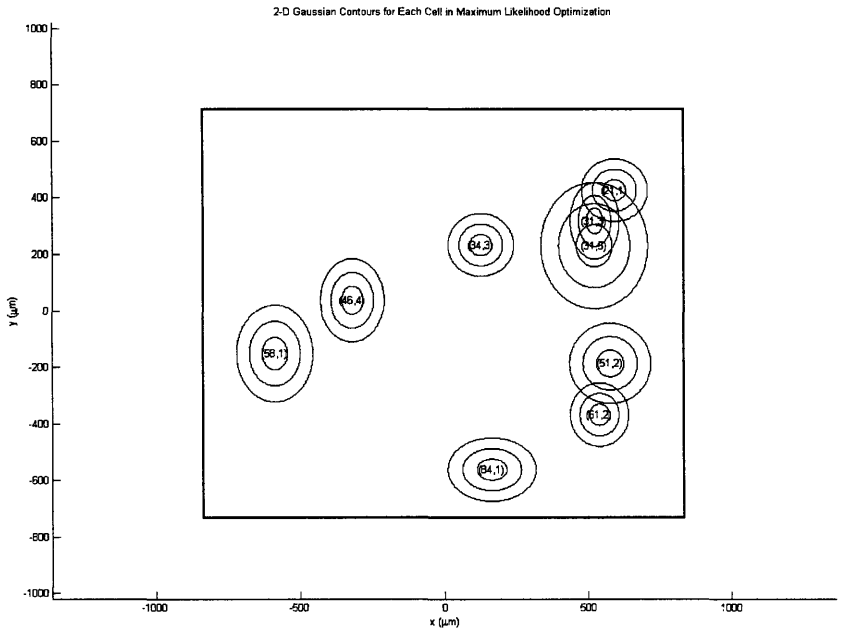

Figure 6-11: The receptive fields of the cells we modeled are depicted in this figure as 2-D Gaussian sensitivities; these receptive fields are of the same form as the one portrayed in figure 6-2. For each cell, three contours are depicted; the height of the sensitivity function along a contour is constant. The square delineates the spatial extent of the MEA from which we were recording.

6.3 The Inverse Problem

This section describes the method from which we estimate the speed and angle of a moving edge given the responses of the cells which we have modeled. We do this by finding the speed and angle which maximize the joint likelihood of these responses. Section 6.3.1 describes this procedure in detail.

6.3.1 Estimating V and θ for Moving Edges via a Maximum Likelihood Approach

Consider the following setup: we have K cells which are stimulated by a moving edge of unknown direction and speed; the duration of this stimulus is T. Each cell's response is a

spike train of length T. The total number of spikes in spike train i is denoted by N_i. The k^{th} spike time of spike train i is denoted by $t_{i,k}$. Our goal is to maximize the likelihood of observing these K spike trains. Let $L_1, L_2, ..., L_K$ denote the likelihood functions of these K spike trains. We assume that the likelihood of each spike train is independent from the likelihood of any other spike train. Then, the overall likelihood L of observing these K spike trains is:

$$L = \prod_{i=1}^{K} L_i \tag{6.3.1}$$

We are interested in maximizing L with respect to V and θ. The (V,θ) pair which maximizes L is our estimate of the speed and angle of the moving edge. Maximizing the likelihood function is equivalent to maximizing the natural logarithm of the likelihood function. Thus, our goal is to maximize the "log-likelihood" function with respect to the pair (V,θ):

$$\max_{V,\theta}\{\ln(L)\} = \max_{V,\theta}\left\{\sum_{i=1}^{K}\ln(L_i)\right\} \tag{6.3.2}$$

From equation (6.2.3), the likelihood L_i of spike train i is:

$$\begin{aligned}\ln(L_i) &= \ln\left\{\lambda(t_{i,1};x_i)\lambda(t_{i,2};x_i)...\lambda(t_{i,N_i};x_i)e^{-\int_0^T \lambda(t;x_i)dt}\right\} \\ &= \sum_{k=1}^{N_i}\ln\left\{\lambda(t_{i,k};x_i)\right\} - \int_0^T \lambda(t;x_i)dt\end{aligned} \tag{6.3.3}$$

where $\mathbf{x}_i = (\mu_{x,i}, \mu_{y,i}, \sigma_{x,i}, \sigma_{y,i}, c_i, K_i)$ are the model parameters for cell i. With the help of the result of (6.2.4), we can rewrite (6.3.3) as follows:

$$\ln(L_i) = \sum_{k=1}^{N_i}\ln\left\{\lambda(t_{i,k};x_i)\right\} - \left(c_i T + \frac{Ki}{V}\right) \tag{6.3.4}$$

Therefore, our objective stated in equation (6.3.2) can now be posed as:

$$\max_{V,\theta}\{\ln(L)\} = \max_{V,\theta}\left\{\sum_{i=1}^{K}\left\{\left(\sum_{k=1}^{N_i}\ln\left\{\lambda(t_{i,k};x_i)\right\}\right)-\left(c_i T+\frac{K_i}{V}\right)\right\}\right\} \tag{6.3.5}$$

where:

$$\ln\left\{\lambda(t_{i,k};x_i)\right\} = \ln\left\{c_i+\frac{K_i e^{-\frac{\{V(t_{i,k}-|\Delta_i|-t_o)-(\cos(\theta)\mu_{x,i}+\sin(\theta)\mu_{y,i})\}^2}{2(\sigma_{x,i}^2\cos^2(\theta)+\sigma_{y,i}^2\sin^2(\theta))}}}{\sqrt{2\pi(\sigma_{x,i}^2\cos^2(\theta)+\sigma_{y,i}^2\sin^2(\theta))}}\right\} \tag{6.3.6}$$

As noted above, our objective is to maximize the log-likelihood function of equation (6.3.5) with respect to the pair (V,θ); since we have "trained" the cells by the method described in section 6.2, we already know the parameters $\mathbf{x}_i = (\mu_{x,i},\mu_{y,i},\sigma_{x,i},\sigma_{y,i},c_i,K_i)$ for each cell. Clearly, maximizing the log-likelihood over (V,θ) is a highly nonlinear problem. As we are optimizing over only two variables, we resort to an exhaustive search over (V,θ). We search for θ over the range 0° to 359° and we search for V over the range 1um/sec to 2000um/sec, which is the physiologically normal range for speeds that we observe in nature; rarely is it the case that we observe a speed of over 2000um/sec in nature. The (V,θ) pair which maximizes (6.3.5) is our estimate of the speed and angle of the moving edge.

6.4 Appendix

The derivation of the rank two update of the inverse of the Hessian matrix is presented in this section; an alternative derivation for the likelihood of a spike train is also presented in this section.

Rank two update of Hessian Inverse:

The Newton condition which holds at each iteration k of the algorithm is:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = B^{k+1}(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)) \tag{6.4.1}$$

where $B_{k+1} \triangleq H_{k+1}^{-1}$ is the inverse Hessian matrix. For notational simplicity, let us use the following notation: $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{q}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$. Then, equation (6.4.1) becomes:

$$\mathbf{s}_k = B^{k+1}\mathbf{q}_k \tag{6.4.2}$$

For some vectors **u** and **w**, the rank two update of the inverse Hessian matrix is:

$$B_{k+1} = B_k + \alpha uu^T + \beta ww^T \tag{6.4.3}$$

where α and β are constants. Our objective is to find the **u**, **w**, α and β which satisfy the Newton condition of equation (6.4.2):

$$\mathbf{s}_k = B_{k+1}\mathbf{q}_k = (B_k + \alpha \mathbf{u}\mathbf{u}^T + \beta \mathbf{w}\mathbf{w}^T)\mathbf{q}_k \tag{6.4.4}$$

Let us set $\mathbf{u} = \mathbf{s}_k$ and $\mathbf{w} = B_k\mathbf{q}_k$; moreover, let us set the constants α and β in the following manner:

$$\alpha \mathbf{u}^T\mathbf{q}_k = 1 \text{ and } \beta \mathbf{w}^T\mathbf{q}_k = -1 \Leftrightarrow \alpha = \frac{1}{\mathbf{u}^T\mathbf{q}_k} \text{ and } \beta = -\frac{1}{\mathbf{w}^T\mathbf{q}_k} \tag{6.4.5}$$

Then, the update formula in (6.4.3) can be rewritten as:

$$B_{k+1} = B_k + \alpha \mathbf{u}\mathbf{u}^T + \beta \mathbf{w}\mathbf{w}^T = B_k + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{q}_k} - \frac{B_k \mathbf{q}_k \mathbf{q}_k^T B_k^T}{\mathbf{q}_k^T B_k^T \mathbf{q}_k} \tag{6.4.6}$$

which is exactly the same formula as that in equation (6.2.11).

Alternative derivation for the likelihood of a spike train:

Consider again the spike train of figure 6-6. We subdivide the time axis into K bins of equal size; these bins are sufficiently small and thus we can either have 0 or 1 spikes in a bin. The bin width is $\delta = \frac{T}{K}$. The probability of observing the spike train of figure 6-6 is the probability of observing k_1 empty bins before t_1 and k_2 empty bins after the bin corresponding to t_1 and before the bin corresponding to t_2 and k_3 empty bins after the bin corresponding to t_2. See figure 6-12 for an illustration:

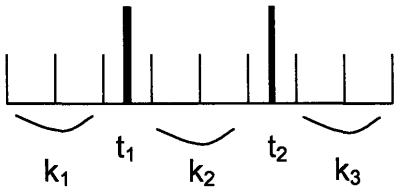

Figure 6-12: This figure illustrates the binned spike train of figure 6-6. There are k_1 empty bins before t_1; there are k_2 empty bins after the bin corresponding to t_1 and before the bin corresponding to t_2; there are k_3 empty bins after the bin corresponding to t_2.

The probability of the spike train is thus:

$$e^{-\lambda\{0\}\delta} e^{-\lambda\{\delta\}\delta} \ldots e^{-\lambda\{k_1\delta\}\delta} \lambda\{t_1\} e^{-\lambda\{(k_1+1)\delta\}\delta} e^{-\lambda\{(k_1+2)\delta\}\delta} \ldots \lambda\{t_2\} e^{-\lambda\{(k_1+1+k_2)\delta\}\delta} \ldots e^{-\lambda\{(k_1+1+k_2+1+k_3)\delta\}\delta} = \lambda\{t_1\}\lambda\{t_2\} e^{-\sum_{i=0}^{k_1+1+k_2+1+k_3} \lambda\{i\delta\}\delta} \tag{6.4.7}$$

If we take the limit of (6.4.7) as $\delta \to 0$, we obtain:

$$\lambda(t_1)\lambda(t_2)e^{-\int_0^T \lambda(t)dt} \tag{6.4.8}$$

This is the same expression as the one in equation (6.1.16).

Chapter 7

Experimental Results

7.0 Acknowledgements and Introduction

Before beginning this chapter, I would like to acknowledge the importance of Jessica's and Shamim's role in processing the data and implementing and running the code necessary to produce the results which are presented in this chapter; it was an immense amount of work and I am grateful that we were able to collaborate so well.

In this chapter, we present the results of estimating the speed and direction of moving edges by using the method presented in chapter 6; this method will be called the "likelihood algorithm". Moreover, we compare these results to those obtained by using the "global firing time information" algorithm presented in chapter 4 of this thesis.

We will present results based on data acquired on the experimental day of 04/06/07. On that day we ran 3 trials of edges moving at a single speed of 714um/sec in 8 distinct directions; these directions of motion were $0^0, \pm 45^0, \pm 90^0, \pm 135^0$, and 180^0. We trained each cell by presenting it with 2 trials of each possible (V,θ) pair (8 pairs total). The details of this experimental day are presented in section 3.3.3 of the methods section. The results presented in this chapter use a total of nine OFF cells; we selected these cells because they fired consistently across all three trials of the stimuli presentations. The receptive fields of the cells which are used are depicted in figure 6-11.

Section 7.1 presents the relationship between our (V,θ) estimates and the number of cells used in the estimation procedure; this relationship is presented for both the likelihood and global firing time information algorithms. Section 7.2 presents the relationship between our (V,θ) estimates and the spacing between the cells used in the estimation procedure; this relationship is presented for both the likelihood and global firing time information algorithms. To fully understand the dependence of the likelihood model on the number of cells and the spacing between the cells used, we also create a

"simulated" retina in which we can control the number and the positioning of cells; the results of these simulations are presented in chapter 8. Section 7.3 presents the fidelity of our data: we make sure that the results do not change significantly depending on which trials we train on. Moreover, we show that all speed and angle errors are zero mean and establish that there is not an experimental imprecision.

7.1 Relationship Between (V,θ) Estimates and Number of Cells Used

As the likelihood algorithm involves optimization over a highly nonlinear and non-quadratic likelihood function, it is infeasible to obtain closed form sensitivity equations of the (V,θ) estimates as functions of the model parameters $x_i = (\mu_{x,i}, \mu_{y,i}, \sigma_{x,i}, \sigma_{y,i}, c_i, K_i)$ of each cell. However, we expect that as more cells are used in estimating (V,θ) with the likelihood algorithm, the errors in the estimates should decrease; this is because the amount of "information" about the stimulus is increased as we use more cells. Figure 7-1 depicts the median of the absolute values of the errors in speed (panel A) and direction (panel B) estimates as a function of the number of cells used. The dots and red line represent experimental data. Each dot corresponds to a single value of N (the number of cells used) and a single direction of edge motion. Each dot represents the median of the absolute values of the errors in the estimate, averaged over $\binom{9}{N}$ trials (all possible N-cell combinations which can be picked from the set of nine cells). For example, suppose we are looking at the results for $N=5$ in panel A. Each of the eight dots in that "column" represents the median of the absolute values of the errors in speed for a certain direction averaged over all possible choices of 5-cell combinations out of the 9 available cells (there are 126 ways to choose 5-cell combinations out of 9 cells). The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as N is varied. The blue line represents the results from simulation: that is, for each of the nine cells which we have modeled, we artificially generate a spike train in response to each stimulus. As we have the inhomogeneous Poisson rate $\lambda(t; x_i)$ for each

cell, we are able to generate a spike train given a speed and angle; such a spike train is a realization of an inhomogeneous Poisson process with rate $\lambda(t; x_i)$. The green line represents results from simulation where we have increased the c and K parameters of each cell to $10c$ and $10K$ respectively (intuitively, the cell will fire 10 times more vigorously than before). From these modified cell models, we artificially generate a spike train in response to each stimulus; the procedure is analogous to that used when obtaining the results for the blue line.

Before continuing, the sources of the differences between the red line and the other lines are discussed. In both the blue and green line simulations, we input the model parameters; thus, there are no errors in determining any of the cell model parameters $x_i = (\mu_{x,i}, \mu_{y,i}, \sigma_{x,i}, \sigma_{y,i}, c_i, K_i)$ because these parameters are specified; there is no data fitting of the type as presented in section 6.2 involved. Thus, in the blue line simulations, we generate spike trains according to an inhomogeneous Poisson process from cell models which are perfectly "known"; we then solve the inverse problem as presented in section 6.3. Thus, the differences between the red line and the blue line are due to the fact that the real spike trains are not generated according to an inhomogeneous Poisson process whereas the simulated spike trains are. Therefore, this difference is an upper bound on the error resulting from our approximation of the cell generated spike trains as instances of an inhomogeneous Poisson process; given that the gaps between the red and blue lines are relatively small (on average, about 20um/sec in the speed estimates and about 1° in the direction estimates), our model of inhomogeneous Poisson firing is a good approximation to reality. Since the blue line simulation is essentially estimating the model which we input, we would expect the speed and angle errors to be very close to 0. However, this is not the case because we get a sparse sampling of the Poisson rate $\lambda(t; x_i)$; specifically, we generate spike trains according an inhomogeneous Poisson process where we know each cell's $\lambda(t; x_i)$ perfectly; however, in the inverse problem, we evaluate the likelihood of each spike train as presented in chapter 6. Clearly, this step would be more accurate if we were able to observe more Poisson generated spikes; in other words, we expect to get a better estimate of a cell's $\lambda(t; x_i)$ if it fires more vigorously. In order to test this hypothesis, we ran simulations in which we increased the

c and K parameters of each cell to $10c$ and $10K$. As is evident from figure 7-1, both the speed and angle errors were pretty close to 0. In theory, we expect that if we continue increasing c and K, we would get errors of 0. The difference between the blue and green lines is that we have a less sparse sampling of each cell's $\lambda(t; x_i)$ in the green line simulation than in the blue line simulation.

As is evident from figure 7-1, both the speed and angle estimates improve as the number of cells used increases; this is true for both the real spike train responses and the simulated spike train responses. For the case where we are using three cells, the median of the absolute values of the errors in speed over all stimuli is around 35um/sec (5% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 3°. However, for the case where we are using all nine cells, the median of the absolute values of the errors in speed over all stimuli is around 20um/sec (3% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 2°. These estimates are very close to the truth. Even for a low number of cells, our estimates are fairly close to the truth.

It is interesting to compare the results of the likelihood algorithm with those of the global firing time information algorithm; note that delay is accounted for in both the global firing time information and likelihood algorithms The simulation results in chapter 5 of Eisenman 2007 [1] suggest that the (V,θ) estimates of the global firing time information algorithm are sensitive to the number of cells being used in the estimation. Figure 7-2 depicts the median of the absolute values of the errors in speed (panel A) and direction (panel B) estimates as a function of the number of cells used for the global firing time information algorithm. The dots and red line represent experimental data. Each dot corresponds to a single value of N (the number of cells used) and a single direction of edge motion. Each dot represents the median of the absolute values of the errors in the estimate, averaged over $\binom{9}{N}$ trials (all possible N-cell combinations which can be picked from the set of nine cells). For example, suppose we are looking at the results for $N=5$ in panel A. Each of the eight dots in that "column" represents the median of the absolute values of the errors in speed for a certain direction averaged over all possible choices of 5-cell combinations out of the 9 available cells (there are 126 ways to

choose 5-cell combinations out of 9 cells). The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as N is varied. The blue line represents the results from simulation: that is, for each of the nine cells which we have modeled, we artificially generate a spike train in response to each stimulus. As we have the inhomogeneous Poisson rate $\lambda(t; x_i)$ for each cell, we are able to generate a spike train given a speed and angle; such a spike train is a realization of an inhomogeneous Poisson process with rate $\lambda(t; x_i)$. The green line represents results from simulation where we have increased the c and K parameters of each cell to $10c$ and $10K$ respectively (intuitively, the cell will fire 10 times more vigorously than before). From these modified cell models, we artificially generate a spike train in response to each stimulus; the procedure is analogous to that used when obtaining the results for the blue line. It is clear from both figure 7-1 and 7-2 that more "sensitive" sensors (i.e. cells which fir more vigorously) give better estimates.

As is evident from figure 7-2, both the speed and angle estimates improve as the number of cells used increases; this is true for both the real spike train responses and the simulated spike train responses. For the case where we are using three cells, the median of the absolute values of the errors in speed over all stimuli is around 50um/sec (7% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 4.5°. These estimates are worse than those from the likelihood algorithm. For the case where we are using all nine cells, the median of the absolute values of the errors in speed over all stimuli is around 30um/sec (4% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 3°. These estimates are slightly poorer than those obtained via the likelihood algorithm. By comparing figures 7-1 and 7-2, it is clear that the likelihood algorithm estimates are more robust to a low number of cells than those of the global firing time information algorithm. For the latter algorithm, there is a dramatic improvement as the number of cells being used increases. It should be noted that the same sets of cells are used for creating the corresponding data points in the likelihood and global firing time information algorithm plots. For example, for the data point $N=5$, the same $\binom{9}{5}$ 5-cell combinations are used in both the likelihood and global firing time information algorithms This is the case in all data results plots of this chapter.

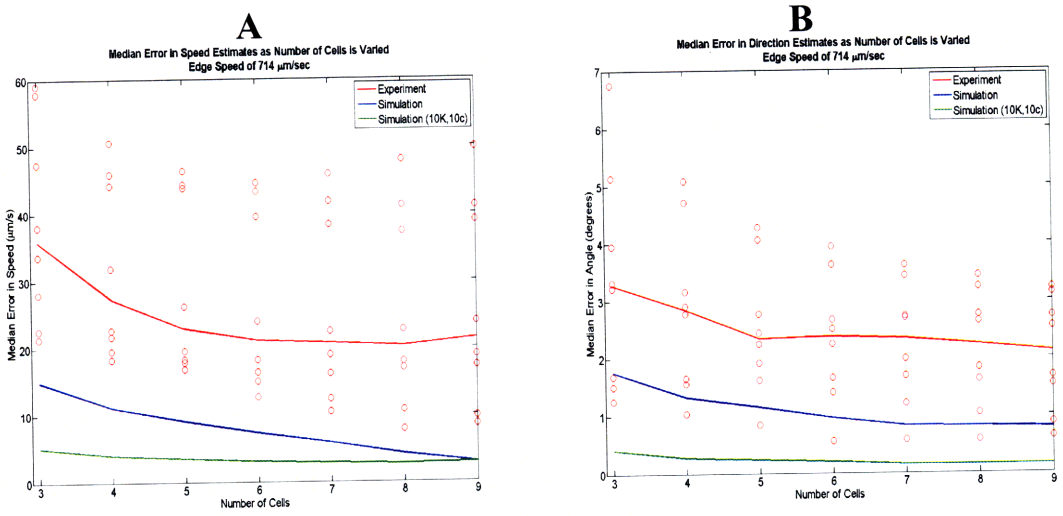

Figure 7-1: A and B depict the median of the absolute values of the errors in speed and direction estimates, respectively, vs. number of cells used for the likelihood algorithm. The dots and red line represent experimental data. Each dot corresponds to a single value of N and a single direction of edge motion. Each dot represents the median of the absolute values of the errors in the estimate, averaged over $\binom{9}{N}$ trials (all possible N-cell combinations which can be picked from the set of nine cells). The red line represents the median value of the eight red dots in the estimate as N is varied. The blue line represents the results from simulation: that is, for each of the nine cells we have modeled, we artificially generate a spike train in response to each stimulus. We subsequently solve the "inverse" problem by maximizing the joint likelihood of these artificially generated spike trains from each cell. The green line represents the results from simulation where we have increased each cell's c and K to $10c$ and $10K$.

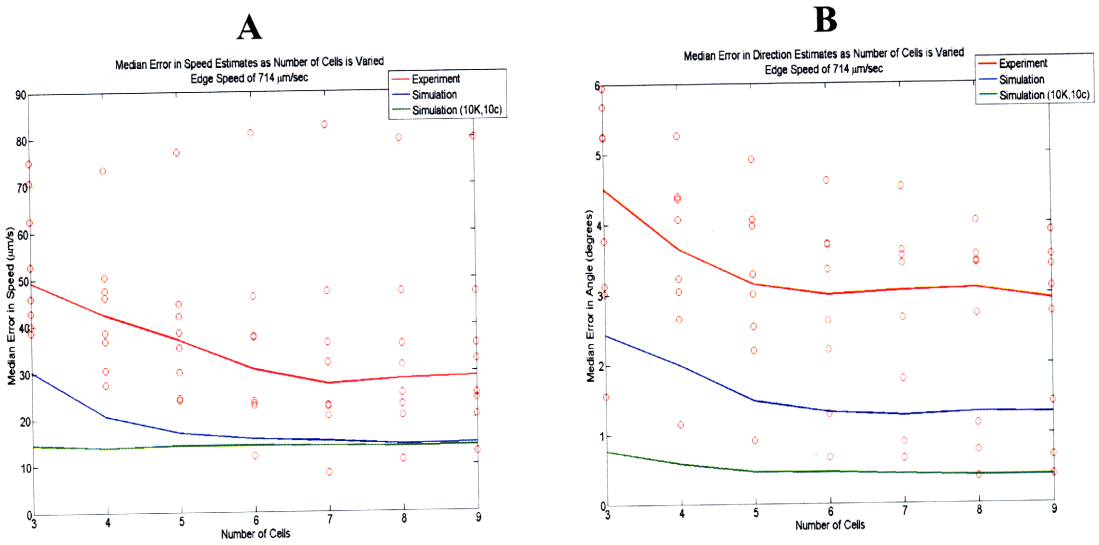

Figure 7-2: A and B depict the median of the absolute values of the errors in speed and direction estimates, respectively, vs. number of cells used for the global firing time information algorithm. The dots and red line represent experimental data. Each dot corresponds to a single value of N and a single direction of edge motion. Each dot represents the median of the absolute values of the errors in the estimate, averaged over $\binom{9}{N}$ trials (all possible N-cell combinations which can be picked from the set of nine cells). The red line represents the median value of the eight red dots in the estimate as N is varied. The blue line represents the results from simulation: that is, for each of the nine cells we have modeled, we artificially generate a spike train in response to each stimulus. We subsequently solve the "inverse" problem by maximizing the joint likelihood of these artificially generated spike trains from each cell. The green line represents the results from simulation where we have increased each cell's c and K to $10c$ and $10K$.

7.2 Relationship Between (V,θ) Estimates and the Spacing Between the Cells Used

As presented in the simulations in chapter 5 of Eisenman 2007 [1], the global firing time information algorithm estimates deteriorate as the distance between the cell pairs used decreases; this is reasonable because as distance between cells decreases, the errors in the location and crossing time estimations of a cell play a bigger role thus degrading the estimates for (V,θ). On the other hand, the likelihood algorithm uses the entire spike train response of a cell and thus, we expect the (V,θ) estimates to be more robust to "clustered" cells. Before continuing, we present the notion of "cell clustering" which we use in this chapter.

7.2.1 The Minimum Bounding Circle

The minimum bounding circle for a set of N points is defined as the smallest circle which encloses these N points; note that at least two of these points lie on the circumference of their minimum bounding circle and that some of these N points may lie in the interior of the minimum bounding circle. Figure 7-3 illustrates the concept of a minimum binding circle for $N=5$ points. The smaller the minimum bounding circle is for N points, the more "clustered" these points are.

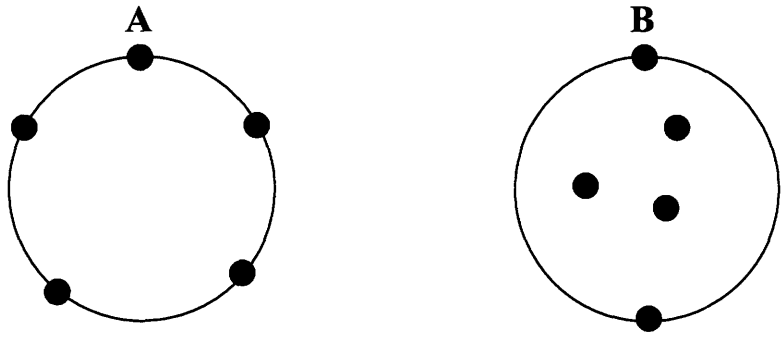

Figure 7-3: This figure illustrates the concept of a minimum bounding circle for 5 points in the plane. Note that the points in panel A and the points in panel B have the same minimum bounding circle.

7.2.2 Relationship Between (V,θ) Estimates and the Minimum Bounding Circle

As mentioned in section 7.2, for the likelihood algorithm, we do not expect worse estimates from cells that are included in a smaller minimum bounding circle than cells which are included in a bigger minimum bounding circle. Figure 7-4 depicts the median of the absolute values of the errors in speed (panel A) and direction (panel B) estimates as a function of the "clustering" of the cells used in the likelihood algorithm. The dots and red line represent experimental data. Each "column" of dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 3-cell combinations are picked from the set of all cells that are enclosed between minimum bounding circles of radius *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. For example, suppose that we are looking at the results for r=480um in panel A. Each of the eight dots in that "column" represents the median of the absolute values of the errors in speed for a certain direction over all possible choices of 3-cell combinations which lie within minimum bounding circles of radii *smaller* than 480um and outside minimum bounding circles of radii *larger* than 440um. The blue line represents the results from simulation: that is, for each of the nine cells which we have modeled, we artificially generate a spike train in response to each stimulus.

As is evident from figure 7-4, both the speed and angle estimates do not vary significantly as the radius of the minimum bounding circle is increased. The median of the absolute values of the errors in speed over all stimuli is around 30um/sec (4% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 3°. Thus, even for a small minimum bounding circle and a small number of cells, our estimates are fairly close to the truth.

It is interesting to compare these results with those obtained via the global firing time information algorithm. Figure 7-5 depicts the median of the absolute values of the errors in speed (panel A) and direction (panel B) estimates as a function of the "clustering" of the cells used in the global firing time information algorithm. The dots and red line represent experimental data. Each "column" of blue dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 3-cell combinations are picked from the set of all cells that are enclosed between minimum bounding circles of radius *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. For example, suppose that we are looking at the results for r=480um in panel A. Each of the eight dots in that "column" represents the median of the absolute values of the errors in speed for a certain direction over all possible choices of 3-cell combinations which lie *within* minimum bounding circles of radii *smaller* than 480um and outside minimum bounding circles of radii *larger* than 440um. The blue line represents the results from simulation: that is, for each of the nine cells which we have modeled, we artificially generate a spike train in response to each stimulus.

As is evident from figure 7-5, the median of the absolute values of the errors in speed slightly decrease as the "clustering" decreases; however, the errors in the angle estimates do not vary significantly as the "clustering" decreases; moreover, for all radii of the minimum bounding circles, the global firing time algorithm underperforms compared to the likelihood algorithm. The median of the absolute values of the errors in speed over all stimuli decreases from around 80um/sec (for cells *inside* minimum bounding circle radii smaller than 350um) to around 50um/sec (for cells *inside* minimum bounding circles with radii *smaller* than 650um and *outside* minimum bounding circles with radii *larger* than 600um); the median of the absolute values of the errors in angle over all stimuli is around 4°. By comparing figures 7-4 and 7-5, it is clear that the likelihood algorithm outperforms the global firing time information algorithm. Moreover, it is evident from

figure 7-4 that the likelihood algorithm estimates are robust to small minimum bounding circles.

Figure 7-6 depicts the median of the absolute values of the errors in speed and direction estimates obtained via the likelihood algorithm vs. the radius of the minimum bounding circles which enclose 5-cell combinations; note that it is exactly analogous to figure 7-4 except that it uses 5-cell combinations instead of 3-cell combinations. Figure 7-7 depicts the median of the absolute values of the errors in speed and direction estimates obtained via the global firing time information algorithm vs. the radius of the minimum bounding circles which enclose 5-cell combinations; this is exactly analogous to figure 7-5 except that it uses 5-cell combinations instead of 3-cell combinations.

For the likelihood algorithm which uses 5-cell combinations (figure 7-6), both the speed and angle estimates do not vary significantly as the radius of the minimum bounding circle is increased. The median of the absolute values of the errors in speed over all stimuli is around 40um/sec (5.5% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 3°. Thus, even for a small minimum bounding circle, our estimates are very close to the truth. For the global firing time information algorithm which uses 5-cell combinations (figure 7-7), both the speed and angle estimates do not vary significantly as the radius of the minimum bounding circle is increased; moreover, its estimates are very close to those of the likelihood algorithm. The median of the absolute values of the errors in speed over all stimuli is around 40um/sec (5.5% of 714um/sec); the median of the absolute values of the errors in angle over all stimuli is around 3°.

From the results presented in this chapter, we can conclude that the likelihood algorithm estimates are fairly robust to a small number of cells and to cells that are "clustered", especially when compared to the global firing time information algorithm estimates. This is as expected since the likelihood algorithm makes use of the entirety of each cell's spike train responses whereas the global firing time information algorithm uses only the first order statistics of each cell's spike train responses; thus, since the likelihood algorithm uses more information than the global firing time information algorithm, we expect the likelihood method to outperform the global firing time information algorithm, especially when the number of cells being used is small. As the

real data which we have is limiting, we further explore the behavior of the likelihood algorithm through simulations which are presented in the next chapter and compare it to the behavior of the global firing time information algorithm.

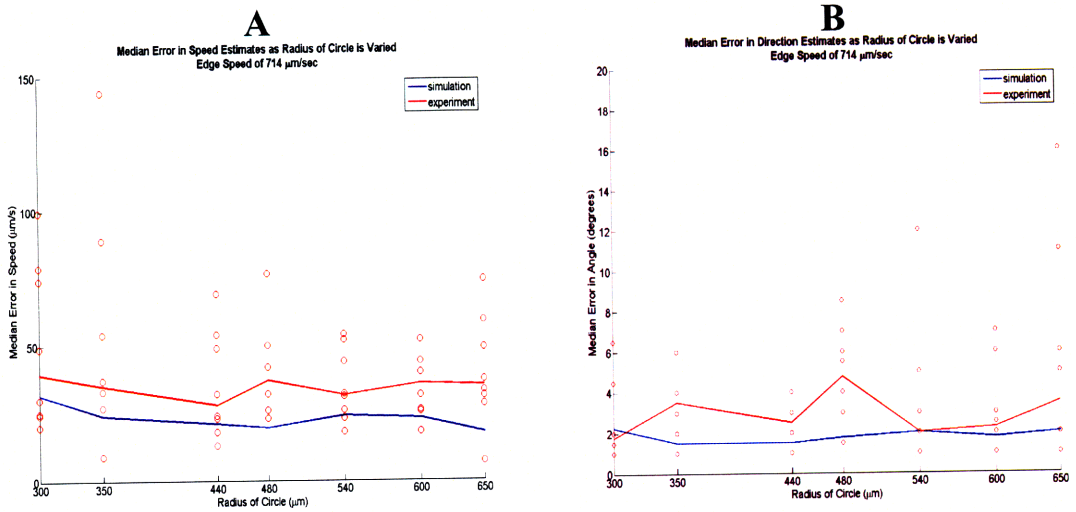

Figure 7-4: 3-cell combinations: A and B depicts the median of the absolute values of the errors in speed and direction estimates respectively as a function of the "clustering" of the cells used in the likelihood algorithm. The dots and red line represent experimental data. Each "column" of red dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 3-cell combinations from the set of all cells that are enclosed between minimum bounding circles of radii *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. The blue line represents the results from simulation.

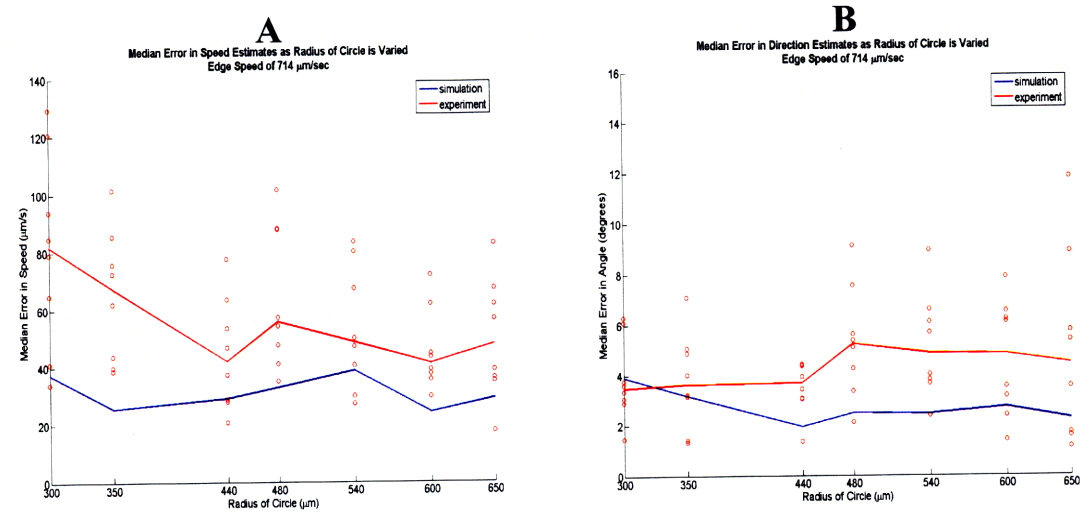

Figure 7-5: 3-cell combinations: A and B depicts the median of the absolute values of the errors in speed and direction estimates respectively as a function of the "clustering" of the cells used in the global firing time information algorithm. The dots and red line represent experimental data. Each "column" of red dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 3-cell combinations from the set of all cells that are enclosed between minimum bounding circles of radii *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. The blue line represents the results from simulation.

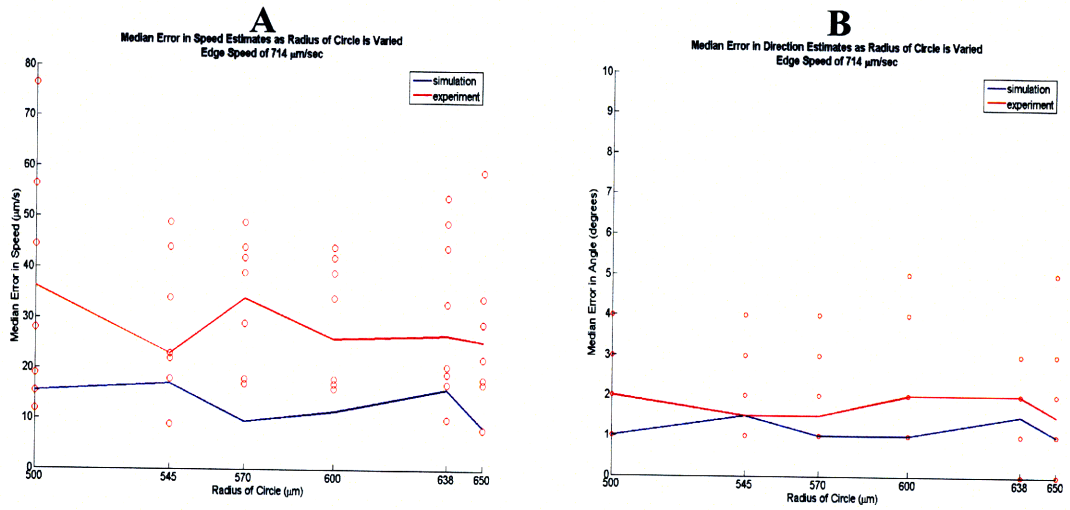

Figure 7-6: 5-cell combinations: A and B depicts the median of the absolute values of the errors in speed and direction estimates respectively as a function of the "clustering" of the cells used in the likelihood algorithm. The dots and red line represent experimental data. Each "column" of red dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 5-cell combinations from the set of all cells that are enclosed between minimum bounding circles of radii *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. The blue line represents the results from simulation.

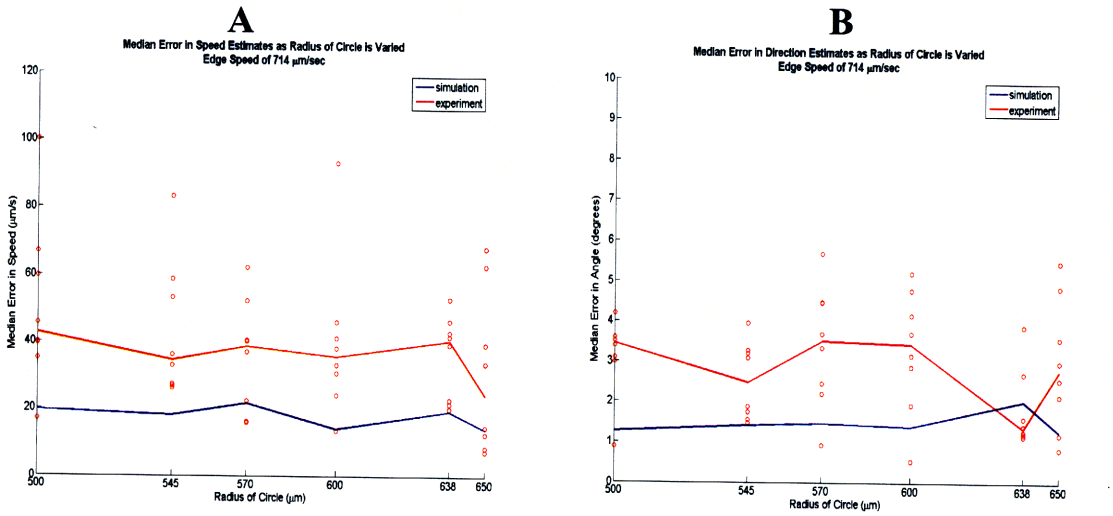

Figure 7-7: 5-cell combinations: A and B depicts the median of the absolute values of the errors in speed and direction estimates respectively as a function of the "clustering" of the cells used in the global firing time information algorithm. The dots and red line represent experimental data. Each "column" of red dots corresponds to estimates made using cells *within* a minimum bounding circle of the corresponding radius (denoted r_n) but *outside* a minimum bounding circle of the smaller radius to the left (denoted r_{n-1}). Each dot corresponds to a single direction of edge motion and it represents the median of the absolute values of the errors in the estimate in which all possible 5-cell combinations from the set of all cells that are enclosed between minimum bounding circles of radii *larger* than r_{n-1} and *smaller* than r_n. The red line represents the median value of the eight red dots (i.e. over edges in 8 distinct directions) in the estimate as the radii of the minimum bounding circles are varied. The blue line represents the results from simulation.

7.3 Data Fidelity

The results presented in the previous sections of this chapter were obtained by "training" on the first two trials of each stimulus and "testing" on the third. In order to remove any concern that our results might depend on which "testing" and "training" trials we are using, we present the results for the likelihood algorithm of "training" on the first and third trials and "testing" on the second trial as well as the results of "training" on the second and third trials and "testing" on the first trial. A and B depict the median of the absolute values of the errors in speed and direction estimates, respectively, vs. number of cells used for the likelihood algorithm. Figure 7-8 depicts the median of the absolute values of the errors in speed (panel A) and direction (panel B) estimates as a function of the number of cells used for the likelihood algorithm. The green line in panel A (B) is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 1 and 2 and "testing" on trial 3; note that the green line in panel A (B) is exactly the same line as the red line in panel A (B) of figure 7-1. The red line in panel A (B) of figure 7-8 is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 2 and 3 and "testing" on trial 1; the blue line in panel A (B) of figure 7-8 is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 1 and 3 and "testing" on trial 2. It is evident that the results are very similar for all 3 cases; the median absolute values of the errors in speed and angle estimates follow the same decreasing trend as the number of cells being used increases. Thus, we are very confident that our results for the likelihood algorithm are independent of which combination of "testing" and "training" trials we use.

Our next objective is to check whether there were any experimental biases present in our data set. To this end, we examine whether the errors in the speed and angle estimates are close to zero mean for the case where we are using all nine cells; moreover, we "train" and "test" on the same trials because we are interested in getting the best possible estimates of the actual speed and angle of each stimulus. For example, if the angle estimate is always greater than the "real" angle, it may be the case that our stimulus angle was different from what we though it was; such a finding would warrant some statistical analysis so as to estimate any experimental bias that may exist. Figure 7-9

presents errors in speed and direction estimates, respectively, for each of the eight edge directions; these errors were obtained by using all 9 cells in the likelihood algorithm. For each direction, the green data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 1; the blue data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 2; the red data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 3. For example, consider the data point for the direction of 180° in panel B: if we "train" and "test" on trial 1, the error obtained from the likelihood algorithm using all 9 cells is around +1.2° (i.e. the algorithm gives an estimate of about 181.2°); if we "train" and "test" on trial 2, the error obtained from the likelihood algorithm using all 9 cells is also around +1° (i.e. the algorithm gives an estimate of about 181°); if we "train" and "test" on trial 3, the error obtained from the likelihood algorithm using all 9 cells is around -0.2° (i.e. the algorithm gives an estimate of about 179.8°). From figure 7-9, it is evident that the errors in both speed and direction are not biased; some are positive and some are negative. Furthermore, the deviations from the values we assume to be true (714um/sec and $0^0, \pm 45^0, \pm 90^0, \pm 135^0$, and 180^0) are small and approximately zero mean. Thus, we conclude with fair certainty that the experiment was not biased.

In the spirit of checking for experimental biases, we also checked whether the "upper" red dots in figures 7-1, 7-2, 7-4, 7-5, 7-6, and 7-7 corresponded to the same direction of motion; if for example the highest error always corresponded to the same direction, this would raise questions about the accuracy of our knowledge of that particular stimulus direction. However, we found that different stimuli corresponded to different "upper" red dots and thus, we are fairly certain of the accuracy of our knowledge about each stimulus' direction of motion.

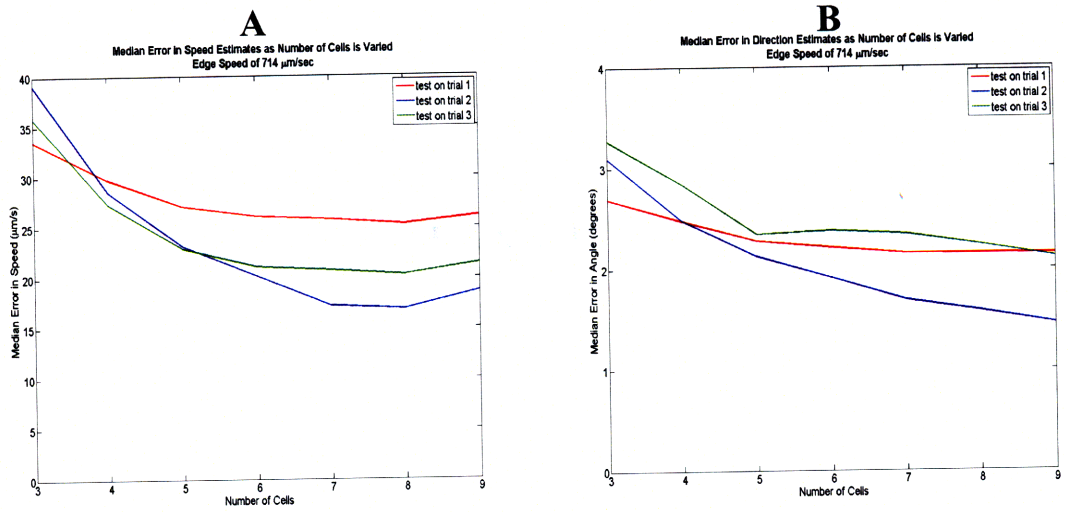

Figure 7-8: A and B depict the median of the absolute values of the errors in speed and direction estimates, respectively, vs. number of cells used for the likelihood algorithm. The green line is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 1 and 2 and "testing" on trial 3; note that the green line is exactly the same line as the red line of figure 7-1. The red line in this figure is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 2 and 3 and "testing" on trial 1; the blue line is the median of the absolute values of the errors in speed (angle) as a result of "training" on trials 1 and 3 and "testing" on trial 2.

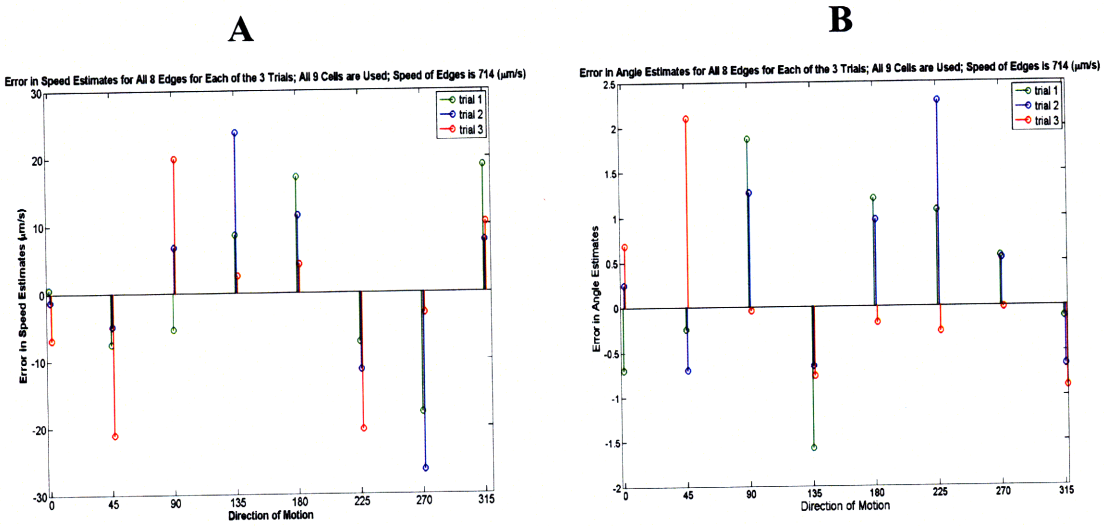

Figure 7-9: A and B depict the errors in speed and direction estimates, respectively, for each of the eight edge directions; these errors were obtained by using all 9 cells in the likelihood algorithm. For each direction, the green data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 1; the blue data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 2; the red data point represents the errors in speed (panel A) and direction (panel B) estimates obtained by "training" and "testing" on trial 3.

Chapter 8

Likelihood Algorithm Simulations

8.0 Acknowledgements and Introduction

Before beginning this chapter, I would like to acknowledge the importance of Jessica's role in implementing the simulation results which are presented in this chapter. I would also like to acknowledge Shamim's help in revising the global firing time algorithm. Their coding skills are truly extraordinary! I am grateful that we were able to collaborate so well.

Given the experimental results of chapter 7, we were keen on further exploring the behavior of the likelihood algorithm; specifically, we examined the fidelity of the (V,θ) estimates as a function of the number of cells being used and as a function of the "clustering" of the cells being used. In this chapter, we present simulation results of the likelihood algorithm and compare them to simulation results of the global firing time algorithm; in these simulations, we are able to control the number of cells and their locations. In this chapter, we also hope to examine whether the results presented in chapter 7 are consistent with the simulation results.

Section 8.1 presents the setup of the simulations; section 8.2 presents the results of the simulations for both the likelihood and global firing time information algorithms; section 8.3 compares the running time of the two algorithms.

8.1 The Simulation Setup

In order to control the locations of each cell exactly, we do not randomly place them within a disc; we place them uniformly on the circle that forms the boundary of the disc. Thus, we are free to control the radius of the circle of which we place the cells, as well as the number of cells we place on the circle; figure 8-1 below illustrates the setup:

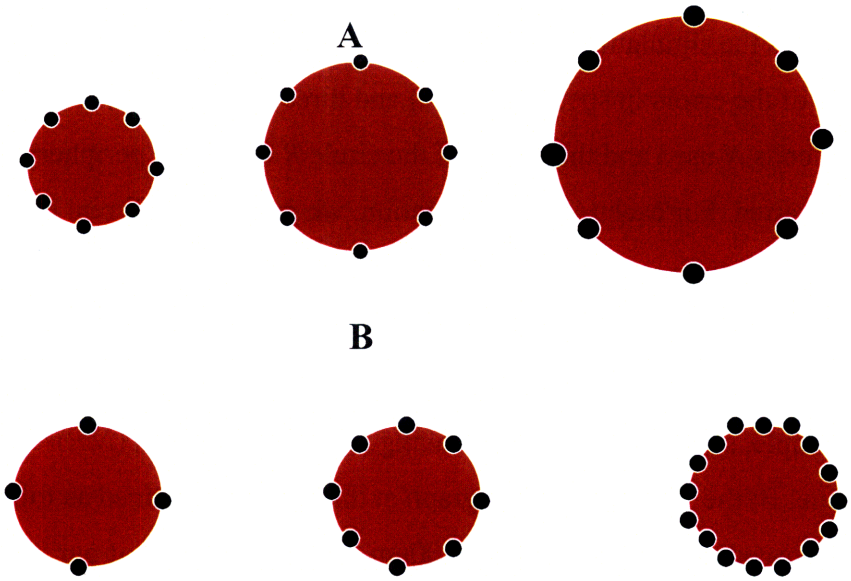

Figure 8-1: This figure illustrates the variables over which we have control in the simulations. In panel A, the number of cells on the circle is kept constant and the radius is varied. In panel B, the radius of the circle is kept constant and the number of cells is varied.

The number of cells N on the periphery of a circle varied from 3 to 15; the radius R of these circles varied from 100um to 800um. The model parameters c and K of the cells were chosen to be the average values of those obtained from real data. As for the extent of each cell's receptive fields, we imposed the condition that it be circular (i.e. $\sigma_x = \sigma_y$) where σ_x and σ_y were chosen to be the average values of those obtained from real data. For each (N,R) combination, we present 8 different edges moving at 714um/sec at 8 different angles ($0^0, \pm 45^0, \pm 90^0, \pm 135^0,$ and 180^0); for each of these edges, each of the N cells responds by generating a spike train according to its model. We then estimate the speed and angle of the edge by maximizing the joint likelihood of the spike train responses of these N cells (in accordance with the method presented in section 6.3). For

each (N,R) combination, we repeat the above procedure 30 times: we then take the median of the absolute value of the resulting 240 errors (8 different edges presented 30 times each).

8.2 Likelihood Algorithm Simulation Results

Figure 8-2 depicts the simulation results for the likelihood algorithm. The medians of the absolute value of the errors in speed (panel A) and direction (panel B) are presented as the number of cells N used and the radius of the circle R on whose periphery these cells are placed are varied. For each (N,R) combination, we present 8 different edges moving at 714um/sec at 8 distinct angles ($0^0, \pm 45^0, \pm 90^0, \pm 135^0$, and 180^0); we repeat this above procedure 30 times For each (N,R) combination, the error which is plotted is the median of the absolute value of the resulting 240 errors (8 different edges presented 30 times each). For example, suppose that we are looking at the combination (N,R)=(5 cells, 200um) in panel B; the value of the 3-D graph at that point is obtained as follows: 5 cells are uniformly spaced on a circle of radius 200um. We present each of our 8 edges 30 times and get 30 estimates of the angle of each edge; thus, for each edge's angle, we get 30 errors. We thus get a total of 240 errors for 8 edges. The value of the graph is the median of the absolute value of these 240 errors.

As is evident from panels A and B of figure 8-2, the errors in the likelihood algorithm decrease as the number of cells used increases and the radius of the circle whose periphery they are placed on increases. However, this effect is not significant for data points with both R greater than 400um and N greater than 7 simultaneously. It is also noteworthy that the worst case errors in speed (40 um/sec, which is 5.5% of 714um/sec) and direction (8°), which occur for the combination (N,R)=(3 cells, 100um), are relatively small. Moreover, if we examine the cross sections at $N=3$ and $N=5$ cells for radii between 300um and 650um, the dependence of the (V,θ) estimates on R is small and non-monotonic. This observation is consistent with the results presented in figures 7-5 and 7-7 of chapter 7 of this thesis. Figure 8-3 depicts the data of figure 7-5 (3-cell combination results for likelihood algorithm) overlaid with the cross section of figure 8-2

where $N=3$ and R is between 300um and 650um. The blue (simulation with real cell positions) and red (experimental data results) lines are exactly the ones of figure 7-5; the green line is the cross section of interest described above from figure 8-2. It is evident from both panels of figure 8-2 that the green line tracks the blue line closely. The effect of increasing R is a little more pronounced in the green line than in the blue line. This makes sense because for the green line, the cells lie uniformly spaced on a circle of radius R whereas for the blue line, the cells lie *within* a circle of radius R. Thus, increasing R increases *all* of the distances between the cells for the green line case; however, it may not increase *all* of the distances between the cells for the blue line case.

Figure 8-4 depicts the data of figure 7-7 (5-cell combination results for likelihood algorithm) overlaid with the cross section of figure 8-2 where $N=5$ and R is between 500um and 650um. The blue (simulation with real cell positions) and red (experimental data results) lines are exactly the ones of figure 7-7; the green line is the cross section of interest described above from figure 8-2. It is evident from both panels of figure 8-2 that the green line tracks the blue line closely. However, in contrast to the situation in figure 8-3, the effect of increasing R is similar in both the green and blue line cases. This may be due to the fact that the range of R is small. Indeed, in figure 8-2, for $N=5$, the decreasing trend of the errors as R increases is evident. However, due to the restrictive real cell locations (blue line), we are not able to explore a wider range of R values.

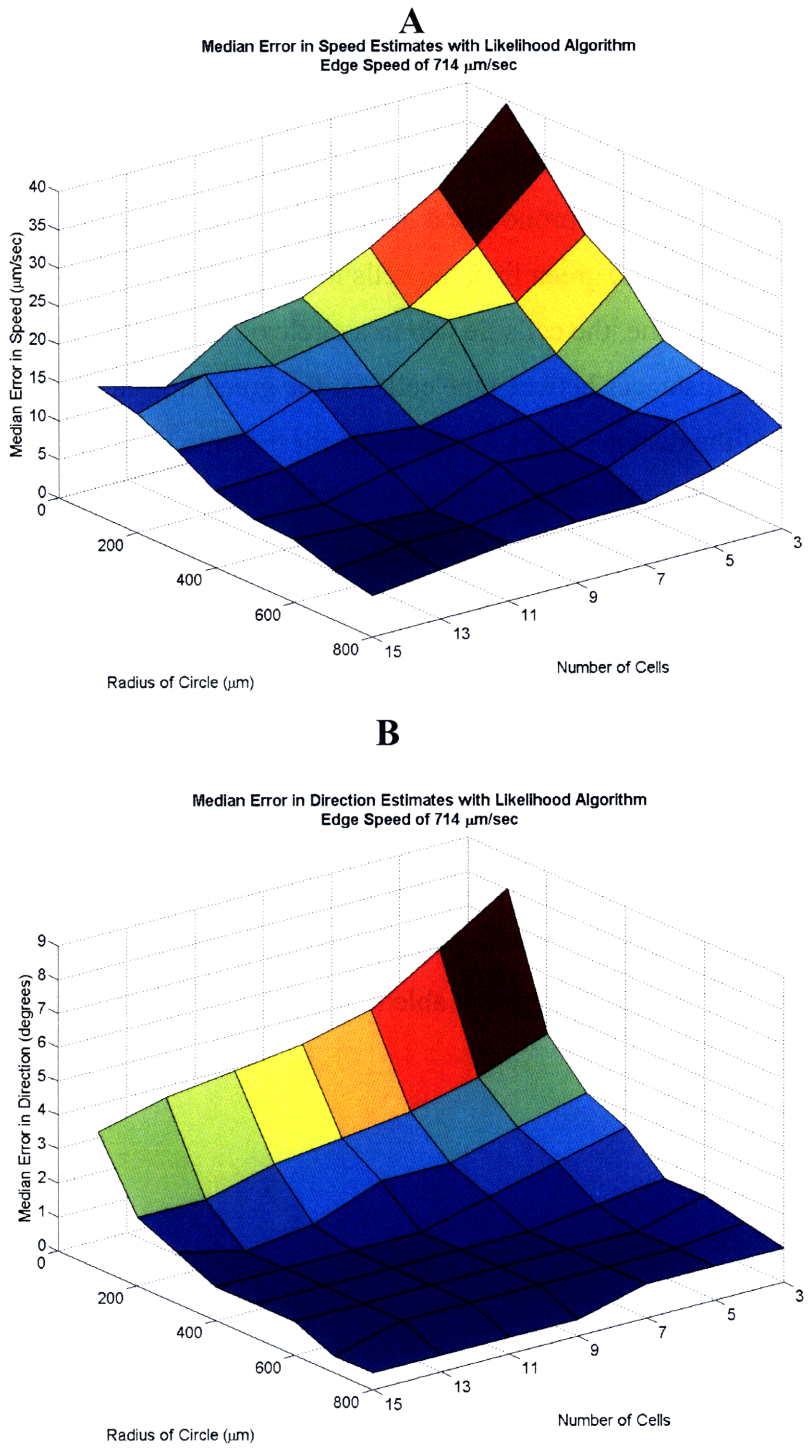

Figure 8-2: A and B show the errors in estimating speed and direction, respectively, by using the likelihood algorithm. The speed of the curtain is 714um/sec. For each N (number of cells) and R (radius of circle) combination, we present 8 different edges moving at 714um/sec at 8 different angles ($0^0, \pm 45^0, \pm 90^0, \pm 135^0,$ and 180^0); we repeat this above procedure 30 times For each (N,R) combination, the error which is plotted is the median of the absolute value of the resulting 240 errors (8 different edges presented 30 times each).

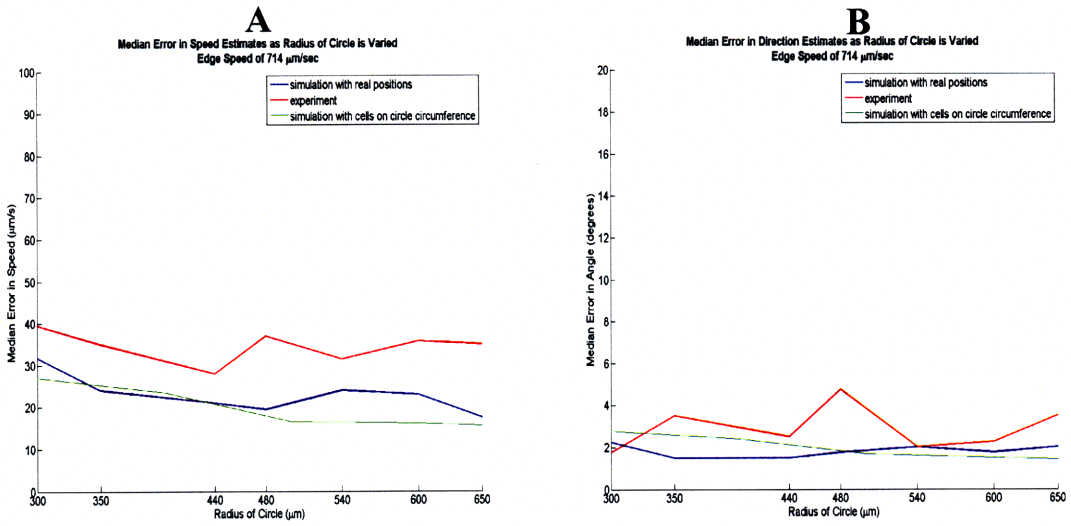

Figure 8-3: *N=3*, 300um<*R*<*650um:* Panels A and B depict the cross section of the corresponding panels of figure 8-2 where *N=3* and *R* is between 300um and 650um (green line). These cross sections are overlaid with the blue (simulation with real cell positions) and red (experimental data results) lines from figure 7-5 (3-cell combination results for likelihood algorithm).

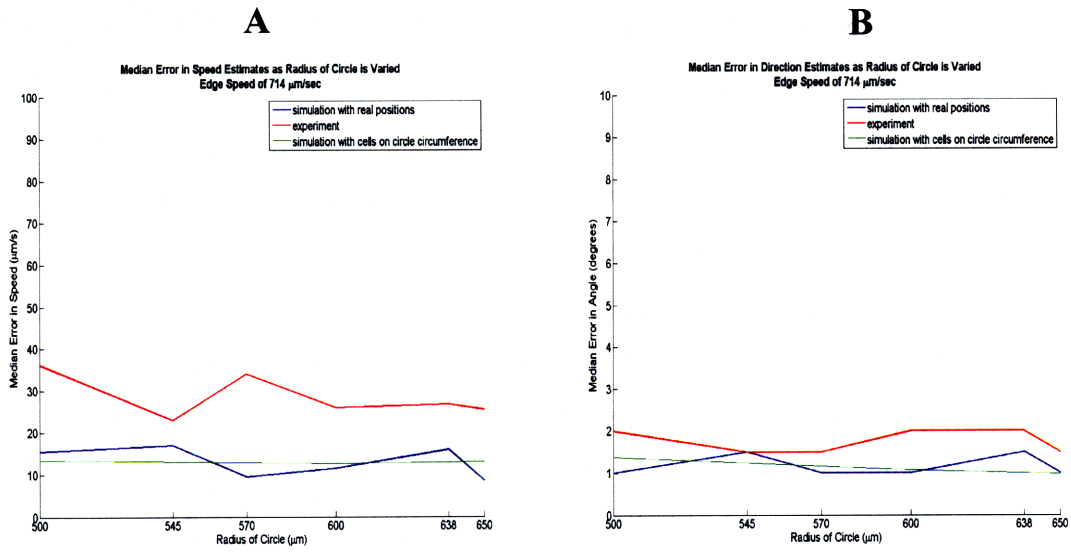

Figure 8-4: *N=5*, 500um<*R*<*650um* Panels A and B depict the cross section of the corresponding panels of figure 8-2 where *N=5* and *R* is between 500um and 650um (green line). These cross sections are overlaid with the blue (simulation with real cell positions) and red (experimental data results) lines from figure 7-7 (5-cell combination results for likelihood algorithm).

8.3 Global Firing Time Algorithm Simulation Results

Figure 8-5 depicts the simulation results for the global firing time information algorithm. The medians of the absolute value of the errors in speed (panel A) and direction (panel B) are presented as the number of cells N used and the radius of the circle R on whose periphery these cells are placed are varied. For each (N,R) combination, we present 8 different edges moving at 714um/sec at 8 distinct angles ($0^0, \pm 45^0, \pm 90^0, \pm 135^0$, and 180^0); we repeat this above procedure 30 times For each (N,R) combination, the error which is plotted is the median of the absolute value of the resulting 240 errors (8 different edges presented 30 times each). For example, suppose that we are looking at the combination (N,R)=(5 cells, 200um) in panel B; the value of the 3-D graph at that point is obtained as follows: 5 cells are uniformly spaced on a circle of radius 200um. We present each of our 8 edges 30 times and get 30 estimates of the angle of each edge; thus, for each edge's angle, we get 30 errors. We thus get a total of 240 errors for 8 edges. The value of the graph is the median of the absolute value of these 240 errors.

As is evident from panels A and B of figure 8-5, the errors in the global firing time information algorithm decrease as the number of cells used increases and the radius of the circle whose periphery they are placed on increases. However, this effect is not significant for data points with both R greater than 400um and N greater than 7 simultaneously. If we examine the cross sections at $N=3$ and $N=5$ cells for radii between 300um and 650um, the dependence of the (V,θ) estimates on R is small and non-monotonic. This observation is consistent with the results presented in figures 7-6 and 7-8 of chapter 7 of this thesis. Figure 8-6 depicts the data of figure 7-6 (3-cell combination results for global firing time information algorithm) overlaid with the cross section of figure 8-5 where $N=3$ and R is between 300um and 650um. The blue (simulation with real cell positions) and red (experimental data results) lines are exactly the ones of figure 7-6; the green line is the cross section of interest described above from figure 8-5. It is evident from both panels of figure 8-6 that the green line tracks the blue line closely. The effect of increasing R is a little more pronounced in the green line than in the blue line. This makes sense because for the green line, the cells lie uniformly spaced on a circle of radius R whereas for the blue line, the cells lie *within* a circle of radius R. Thus,

increasing R increases *all* of the distances between the cells for the green line case; however, it may not increase *all* of the distances between the cells for the blue line case.

Figure 8-7 depicts the data of figure 7-8 (5-cell combination results for global firing time information algorithm) overlaid with the cross section of figure 8-5 where $N=5$ and R is between 500um and 650um. The blue (simulation with real cell positions) and red (experimental data results) lines are exactly the ones of figure 7-8; the green line is the cross section of interest described above from figure 8-5. It is evident from both panels of figure 8-7 that the green line tracks the blue line closely. However, in contrast to the situation in figure 8-6, the effect of increasing R is similar in both the green and blue line cases. This may be due to the fact that the range of R is small. Indeed, in figure 8-5, for $N=5$, the decreasing trend of the errors as R increases is evident. However, due to the restrictive real cell locations (blue line), we are not able to explore a wider range of R values.

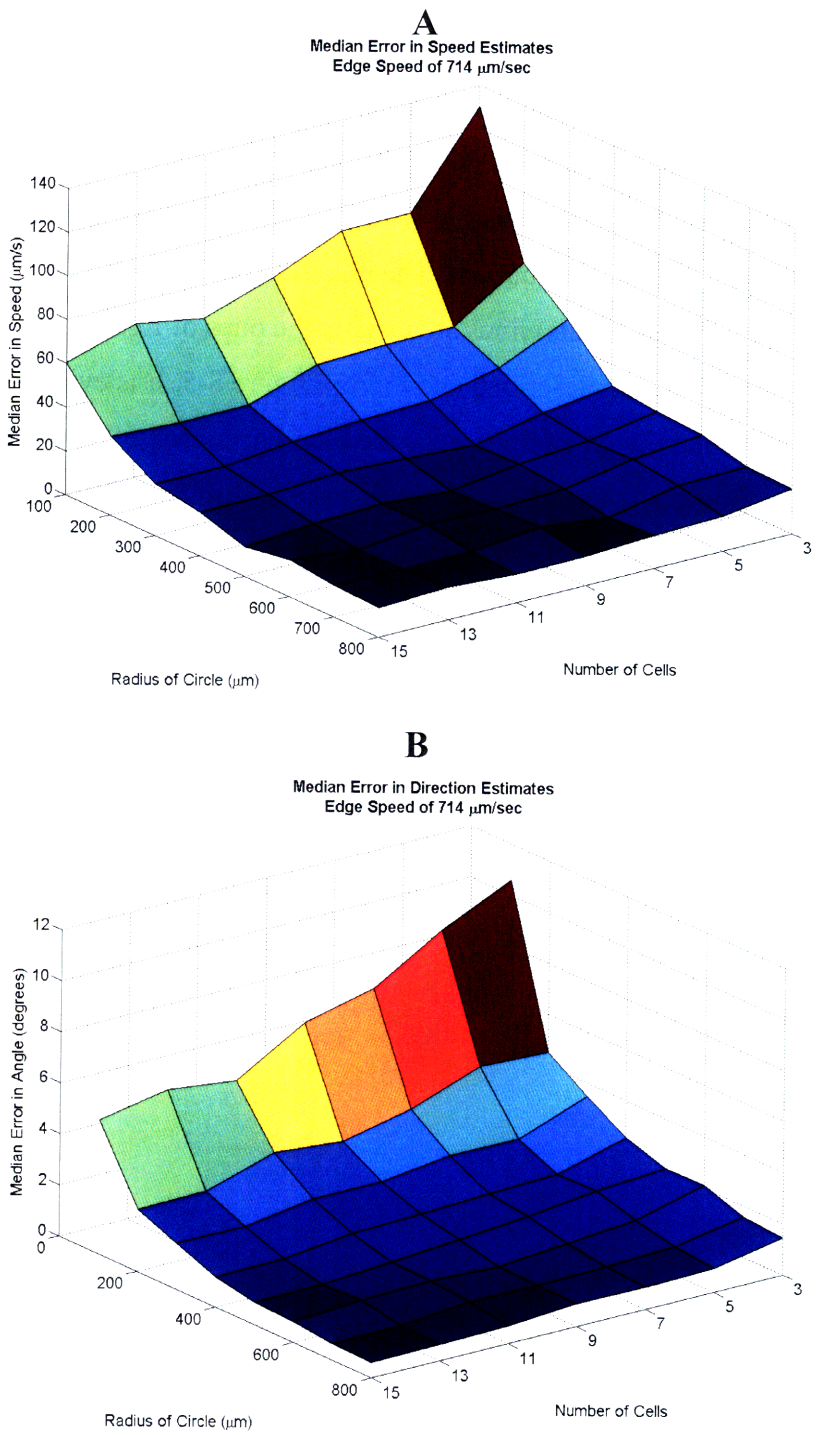

Figure 8-5: A and B show the errors in estimating speed and direction, respectively, by using the global firing time information algorithm. The speed of the curtain is 714um/sec. For each N (number of cells) and R (radius of circle) combination, we present 8 different edges moving at 714um/sec at 8 different angles ($0^0, \pm 45^0, \pm 90^0, \pm 135^0,$ and 180^0); we repeat this above procedure 30 times For each (N,R) combination, the error which is plotted is the median of the absolute value of the resulting 240 errors (8 different edges presented 30 times each).

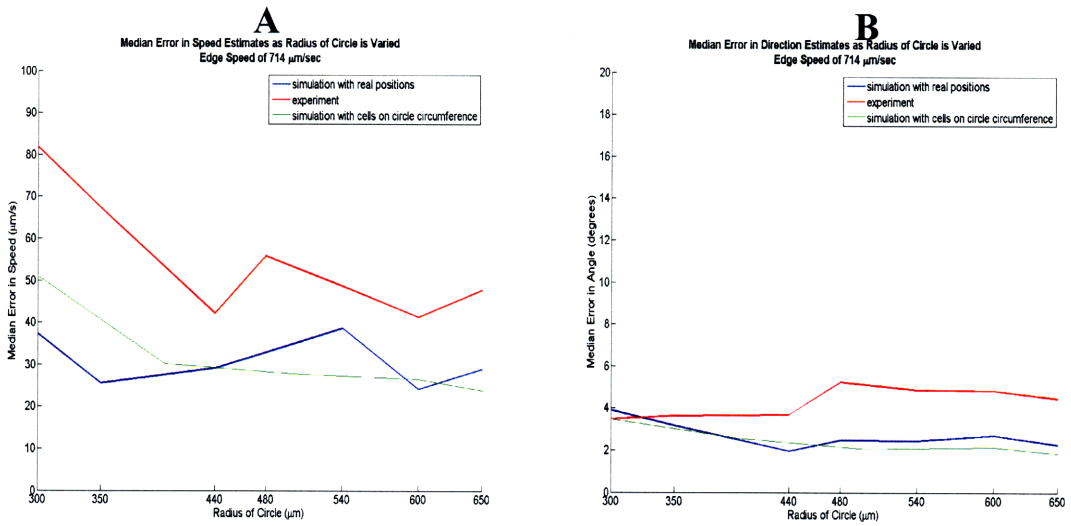

Figure 8-6: *N=3*, 300um<*R*<*650um:* Panels A and B depict the cross section of the corresponding panels of figure 8-5 where *N=3* and *R* is between 300um and 650um (green line). These cross sections are overlaid with the blue (simulation with real cell positions) and red (experimental data results) lines from figure 7-6 (3-cell combination results for global firing time information algorithm).

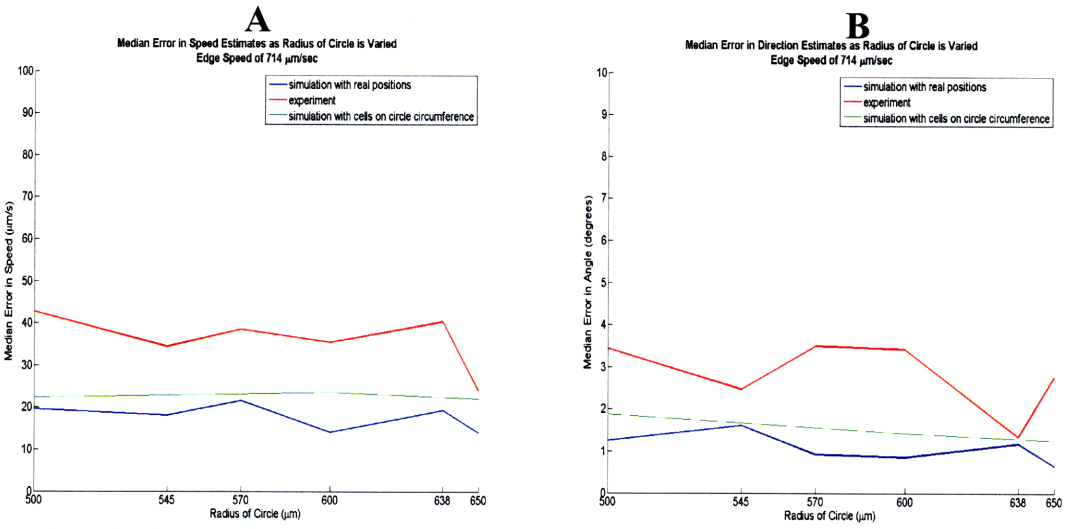

Figure 8-7: *N=5*, 500um<*R*<*650um* Panels A and B depict the cross section of the corresponding panels of figure 8-5 where *N=5* and *R* is between 500um and 650um (green line). These cross sections are overlaid with the blue (simulation with real cell positions) and red (experimental data results) lines from figure 7-8 (5-cell combination results for global firing time information algorithm).

8.4 Algorithm Performance Comparison

It is evident from figures 8-2 and 8-5 that the likelihood algorithm outperforms the global firing time information algorithm for all N and R. Figure 8-8 compares the median of the absolute values of errors in speed (panel A) and the median of the absolute values of errors in direction (panel B) between the likelihood algorithm and the global firing time information algorithm. Panel A (B) is essentially panel A (B) of figure 8-5 minus panel A (B) of figure 8-2. In other words, for each (N,R) combination, the values in panel A (B) of figure 8-8 are the corresponding values of the global firing time information minus the corresponding values for the likelihood algorithm. For example, consider the point (N,R)=(3 cells, 100um) in panel A of figure 8-8: the difference of the medians of the absolute values of the speed errors at that point (approximately 90um/sec) is the median of the absolute values of the speed errors obtained via the global firing time algorithm (approximately 130um/sec) minus the median of the absolute values of the speed errors obtained via the likelihood algorithm (approximately 40um/sec).

Figure 8-9 also divides the medians of the absolute values of errors in speed (panel A) and the medians of the absolute values of errors in direction (panel B) between the likelihood algorithm and the global firing time information algorithm. Panel A (B) is essentially panel A (B) of figure 8-5 *divided by* panel A (B) of figure 8-2. In other words, for each (N,R) combination, the values in panel A (B) of figure 8-9 are the corresponding values of the global firing time information divided by the corresponding values for the likelihood algorithm. A ratio greater than one corresponds to the global firing time algorithm giving a *larger* error than the likelihood algorithm; a ratio less than one corresponds to the global firing time algorithm giving a *smaller* error than the likelihood algorithm. For example, consider the point (N,R)=(3 cells, 100um) in panel A of figure 8-9: the ratio of the medians of the absolute values of the speed errors at that point (approximately 3) is the median of the absolute values of the speed errors obtained via the global firing time algorithm (approximately 130um/sec) divided by the median of the absolute values of the speed errors obtained via the likelihood algorithm (approximately 40um/sec).

It is noteworthy that for *all* (N,R) combinations, the errors in both speed and direction obtained by the global firing time information algorithm are larger than those obtained via the likelihood algorithm. Moreover, the likelihood algorithm *significantly* outperforms the global firing time information algorithm for low values of R when estimating speed and low values of N and R when estimating angle. For large values of N and R, the difference between the two algorithms is very small. The above difference between the two algorithms is consistent with our intuition: the likelihood algorithm uses the entire spike train response of each cell whereas the global firing time information algorithm uses only the median of each spike train response of each cell (as it only cares about the time at which the edge passes over the center of each cell). Thus, the likelihood algorithm uses much more information than the global firing time information algorithm; this additional information is related to the size of the receptive field of each cell. Thus, in the likelihood algorithm, each cell has an opinion about the relationship between V and θ. For example, let us consider the case where a cell has a circular receptive field ($\sigma_x = \sigma_y$): then, if the width of its spike train response is "wide", the cell "knows" that the edge is moving slowly; conversely, if the width of its spike train response is "narrow", the cell "knows" that the edge is moving fast. On the other hand, the global firing time information algorithm only uses the median of a cell's spike train response and thus cannot distinguish between a fast moving edge and a slow moving edge when given information from a single cell. As another example, consider the case where the cell has a very eccentric receptive field. In the likelihood algorithm, the cell has an opinion about the relationship between V and θ: for example, if its spike train response is "wide", the cell "knows" that the edge cannot both be moving fast and along its minor axis. Thus, given the above, we expect the likelihood algorithm to significantly outperform the global firing time information algorithm when a small number of cells are used since in the likelihood algorithm, each individual cell has an opinion of the relationship between V and θ. As shown in the simulations presented, this is indeed the case.

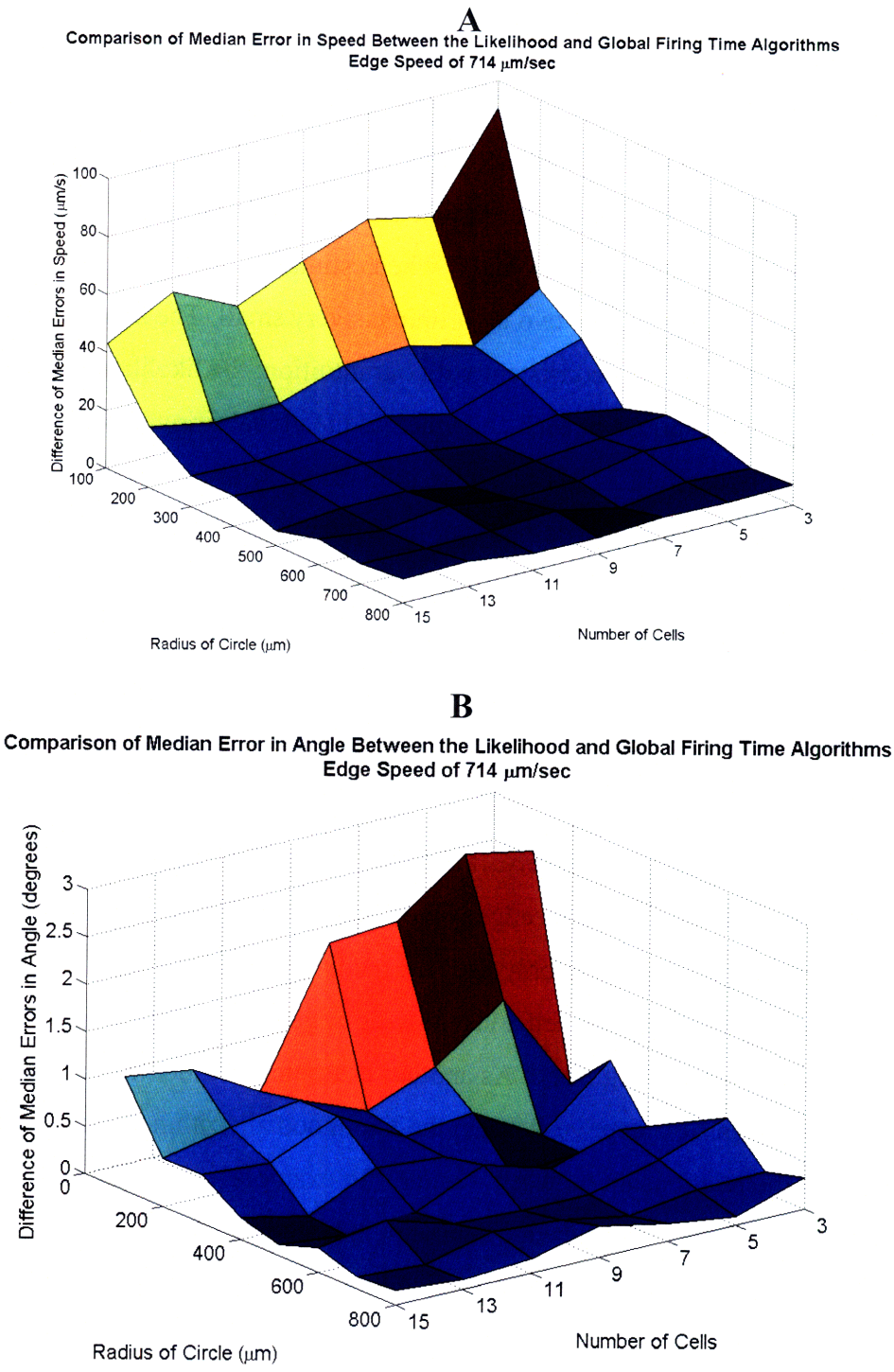

Figure 8-8: This figure compares the median of the absolute values of the errors in speed (panel A) and the median of the absolute values of the errors in direction (panel B) between the likelihood algorithm and the global firing time information algorithm. Panel A (B) is essentially panel A (B) of figure 8-5 minus panel A (B) of figure 8-2. In other words, for each (N,R) combination, the values in panel A and B are the corresponding values of the global firing time information minus the corresponding values for the likelihood algorithm.

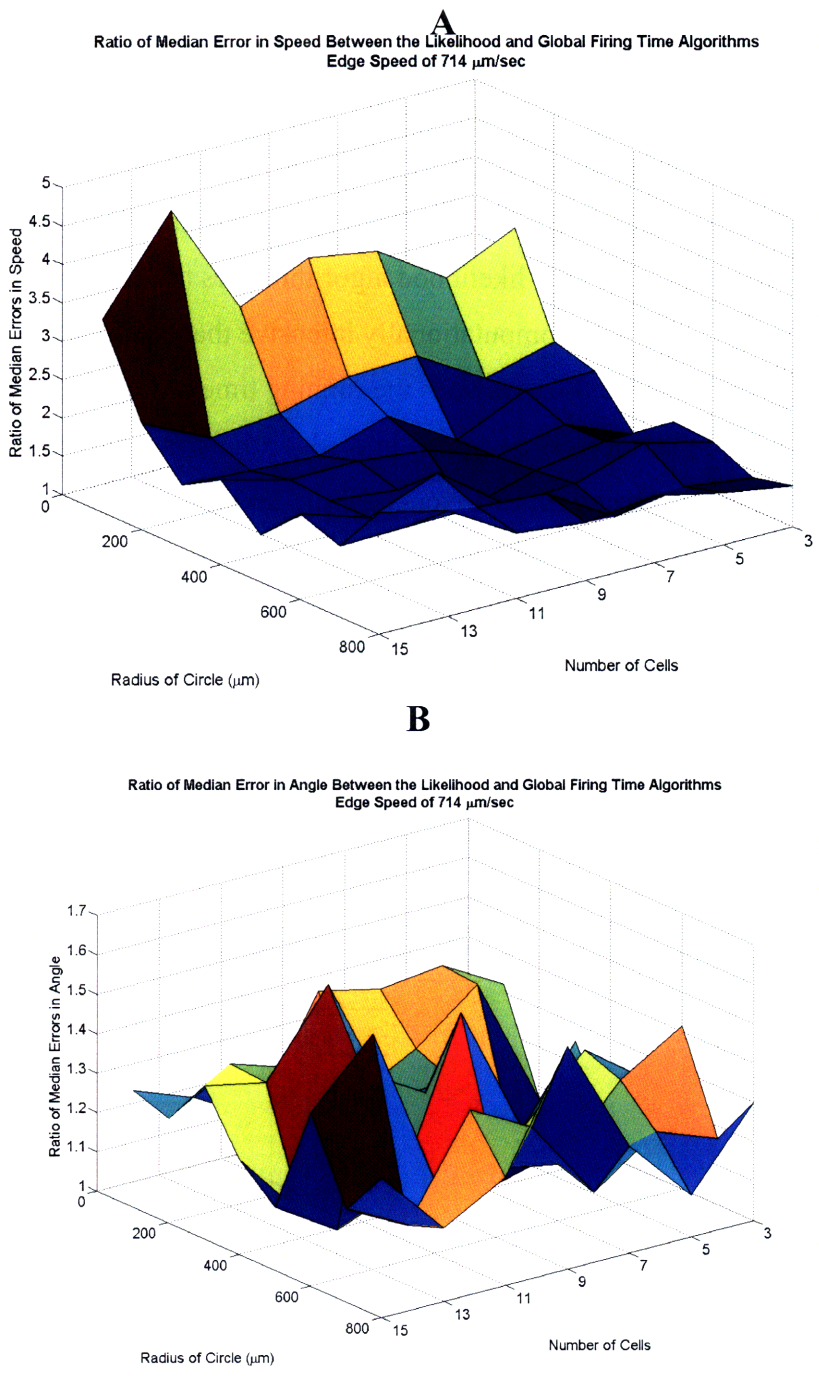

Figure 8-9: This figure depicts the comparison ratios of the median of the absolute values of the errors in speed (panel A) and the median of the absolute values of the errors in direction (panel B) between the likelihood algorithm and the global firing time information algorithm. Panel A (B) is essentially panel A (B) of figure 8-5 divided by panel A (B) of figure 8-2. In other words, for each (*N,R*) combination, the values in panel A and B are the corresponding values of the global firing time information divided by the corresponding values for the likelihood algorithm.

8.5 Comparison of Running Times

The simulations presented in the previous section show that the likelihood algorithm gives better estimates of (V,θ) than the global firing time information algorithm. However, there is a tradeoff: as the likelihood algorithm uses the entire spike train response of each cell, it is more computationally intensive than the global firing time information algorithm. Figure 8-6 compares the running time of the likelihood (blue line) and the global firing time information (green line) algorithms as a function of the number of cells. Note that these running times are only for the Inverse Problem. Note also that the computer used has a speed of 3.8GHz. The global firing time information algorithm is approximately constant in time as a function of the number of cells being used whereas the likelihood algorithm is approximately linear in time as a function of the number of cells being used. Thus, if we are using a large number of cells, the global firing time information algorithm is preferable to the likelihood algorithm as the global firing time algorithm information runs in constant time and gives estimates very close to those obtained by the likelihood algorithm.

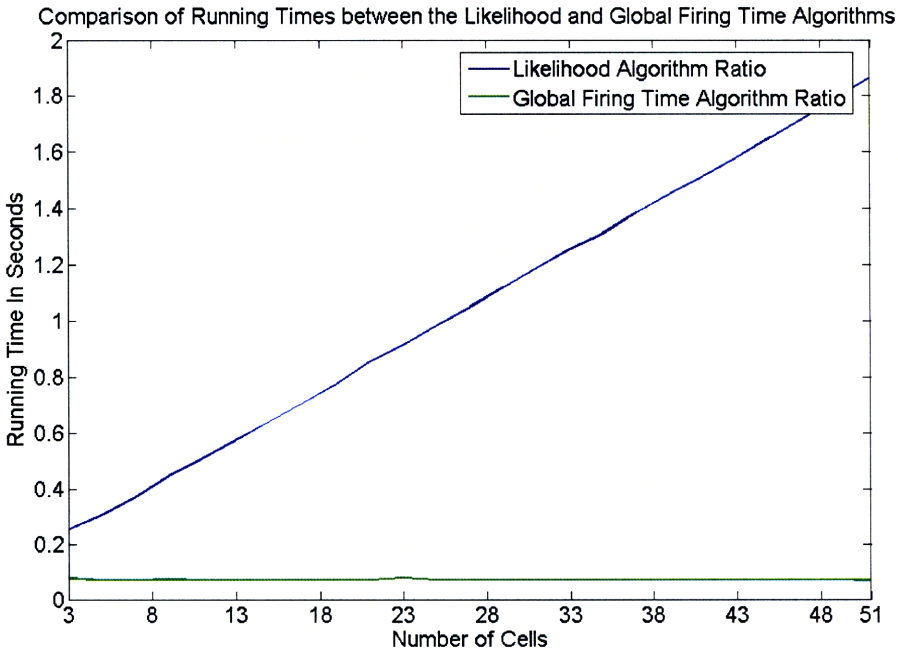

Figure 8-10: This figure compares the running time of the likelihood (blue line) and the global firing time information (green line) algorithms as a function of the number of cells. The likelihood algorithm is approximately linear in time as a function of the number of cells being used and the global firing time information algorithm is approximately constant in time as a function of the number of cells being used.

Chapter 9

Conclusions and Further Work

In this thesis, we have studied an instance of an Inverse Problem which involves estimating the speed and direction of moving edges of light moving at fixed speeds and directions on the photoreceptor layer of a rabbit retinal patch. We obtained retinal ganglion cell responses by recording their action potentials in response to these moving edges of light. After assigning each action potential to a specific cell, we obtained the set of all action potential occurrence times (i.e. the spike train response). The algorithms we employed to estimate the speed and direction of a moving edge take as input the spike train response of every cell. We developed two types of estimation algorithms: the first type of algorithms, presented in chapter 4, is based on a least squares approach. These algorithms use only the first order statistics of each cell's spike train response. The logic of these algorithms is based on the intuition that edge motion information can be inferred from the relative response times of cells. The second type is developed in chapter 6: in essence, it is an estimation algorithm which utilizes maximum likelihood estimation: in contrast to the algorithms of chapter 4, this algorithm does not use least squares. For each cell, a model is obtained through "training". The likelihood of each cell's response is then computed and the joint likelihood of all cells' responses is maximized. In addition to developing estimation algorithms, novel work on weighting least squares was done. Professor Wyatt and I, with the input of Professor Megretski, found a method that minimizes a *weighted sum of the variances* of a set of parameter estimates which we are interested in. Section 9.1 summarizes the conclusions of each chapter of this thesis; section 9.2 proposes relevant further work which could serve as a continuation of this pioneering effort in understanding the retinal neural code.

9.1 Main Conclusions of Thesis Chapters

This section summarizes the conclusions of this thesis. In chapter 4, four least squares based algorithms were developed in order to estimate the speed and angle of a moving edge. Three of these algorithms, the "CosCos", the "Newton-Raphson Minimization", and the "Estimating Velocity Vector Directly" algorithms use information from cell pairs. In these algorithms, each cell pair gives an "opinion" of the speed and angle of the moving edge and the difference in "opinions" between the cell pairs is reconciled through least squares. The "Global Firing Time Information" algorithm takes as input the time which each cell "says" that the edge passed over it and compares it to the time at which the edge "should have" passed over each cell. Thus, each cell's time discrepancy is an "opinion" of the speed and angle of the moving edge: the difference in "opinions" between the cells is reconciled through least squares. The main conclusion of chapter 4 is that the speed and angle estimates for all four algorithms are sensitive to the "clustering" of the cells and to the number of cells used. For the CosCos and Newton-Raphson minimization algorithms, for the case where the cells lie uniformly spaced on the circumference of a circle, it was analytically derived that the variance of the errors in both speed and direction estimates is inversely proportional to both the number of cells used and to the square of the radius of the circle on whose circumference the cells are placed on.

In chapter 5, we proved that the BLUE estimator, in addition to minimizing the *variance of the weighted sum* of the elements in our parameter estimates $\hat{x}$, also minimizes the *weighted sum of the variances* the elements in our parameter estimates $\hat{x}$. Moreover, we found a family of weight matrices W which all lead to the BLUE estimator.

In chapter 6, we developed an estimation algorithm which uses the entirety of each cell's spike train response to a moving edge. The algorithm consists of creating a receptive field model for each cell through "training": the model parameters for a cell are obtained by maximizing the likelihood over the model parameters of observing the cell's responses to the training data. In order to estimate the speed and angle of an unknown edge, we maximize the joint likelihood (with respect to the speed and angle) of the responses of all cells which are stimulated by this edge.

In chapter 7, we tested the likelihood algorithm performance on real data; moreover, we compared it to the performance of the global firing time information algorithm. For the both algorithms, we observed that the errors in the estimates decreased as the number of cells used increased. The worst errors for the likelihood and global firing time information algorithms were for the case where 3 cells were used (5% and 7% respectively in speed, 3° and 4.5° respectively in angle); the best case was when all 9 cells were used (3% and 4% respectively in speed, 2° and 3° respectively in angle). It is encouraging that even when using only three cells, the errors were very small for both algorithms. Also, for both algorithms, we observed that there was not much of a dependence of the estimates on the "clustering" of the cells used. This is mainly due to the limitations of the data. Presumably, if we could use more cells that were more uniformly distributed on the MEA, we would be able to see the errors decrease as the cell "clustering" decreased. Overall, the likelihood algorithm outperforms the global firing time information algorithm for all cases. Another important conclusion from this chapter is that the assumption of inhomogeneous Poisson firing is not very costly. This can be seen by comparing the simulated spike train results (blue lines of figures 7-1 and 7-2) to the real spike train results (red lines of figures 7-1 and 7-2). As explained in chapter 7, the difference between the blue and red lines gives an upper bound on the error from the inhomogeneous Poisson firing approximation.

In chapter 8, we explore the behavior of the two algorithms through a "simulated" retina. We place the cells uniformly on the circumference of a circle where we are free to vary the number of cells and the radius of the circle. Both algorithms' performance increases as the radius of the circle is increased and as the number of cells being used increases. Moreover, the likelihood algorithm slightly outperforms the global firing time information algorithm in all cases. However, the computation time needed for the global firing time information algorithm is much smaller than that of the likelihood algorithm. The global firing time information algorithm is a constant time algorithm with respect to the number of cells being used whereas the likelihood algorithm is linear in time with respect to the number of cells being used.

9.2 Further Work

This section is divided into short term further work goals (section 9.2.1) and long term directions (section 9.2.2).

9.2.1 Short Term Further Work

The following topics (in no particular order) are worth exploring in the short term:

1. Develop cell models which include a covariance term.
2. Generate trials from cell models with *1000c* and *1000K*. The error should go to 0 as c and K are increased further.
3. Perform a sensitivity analysis on the likelihood algorithm. It would be nice to be able to weigh each cell's spike train differently when maximizing the joint likelihood. Find out what makes a spike train "good": should ones with more spikes be trusted more than those with few spikes?
4. Create simulations analogous to those of chapter 8 where c, K, σ_x, and σ_y are varied and explore these effects on the fidelity of the estimates.
5. Create a more realistic model for cells which incorporates the ON-center excitatory and OFF-surround inhibitory effects.
6. Instead of estimating a cells lag with the method in section 6.1.3, make it a model optimization parameter.
7. Create cell models for other cell types such as DS cells. Presumably, this could be done by making the K parameter a function of the angle of edge motion.
8. Create a cell model analogous to the one in chapter 6 which can be used for estimating the speed and direction of moving bars.
9. Apply the likelihood model results to the data of the experimental date 02/09/07. The data quality is not as good as that of 04/06/07 but if no new data is obtained, this data is workable. The 02/09/07 data has two usable curtain speeds. Note that

the format of the triggers is different-some coding needs to be done to make it in the same format as the data in 04/06/07.

9.2.2 Long Term Directions

Below, I have outlined the long term directions which I think that this pioneering research should go down:

1. Create a generalized model which holds for all 13 ganglion cell types. The model should help us be able to distinguish cell types according to a set of parameter values. For example, in the cell model of chapter 6, if we train a cell and we get large σ_x and σ_y, and for another cell, we get small σ_x and σ_y, it is more likely that the first cell is an alpha cell and the second cell is a beta cell.
2. Estimate more stimulus parameters. Contrast should be the next parameter added.
3. Explore whether cells near each other exhibit synchronous firing. Can we use cell synchrony in our estimation process?
4. Stimulate the retina with images of moving balls and try to find ways of estimating the number of balls being shown, their speed, and their direction of movement.
5. Stimulate the retina with moving edges with a time varying speed and see if the speed can be tracked.
6. Stimulate the retina with moving or still objects of different shape (i.e. triangles, squares, circles, etc.) and devise algorithms which can distinguish between the shapes. A first step would be to extend the model of chapter 6 to work for moving bars and then stimulate the retina with moving bars of different widths. There would then be 3 estimation parameters: the speed of the bar, its direction of motion, and its width.

Bibliography

[1] A. Eisenman. *Estimating Light Edge Velocity Based on Retinal Ganglion Cell Spike Trains*. MIT Thesis, Dept. of Electrical Engineering and Computer Science, 2007.

[2] E.N. Brown, L.M. Frank, D.Tang, M.C. Quirk, and M.A. Wilson. *A Statistical Paradigm for Neural Spike Train Decoding Applied to Position Prediction from Ensemble Firing Patterns of Rat Hippocampal Place Cells*. J. Neuroscience, 18(18):7411-7425, 1998.

[3] C.G. Broyden. *The Convergence of a Class of Double-rank Minimization Algorithms*. IMA Journal of Applied Mathematics, 6(3):222-231, 1970.

[4] S.M. Carcieri, A.L. Jacobs, and S. Nirenberg. *Classifcation of Retinal Ganglion Cells: A statistical Approach.* J. Neurophysiology, 90:1704-1713, 2003.

[5] S.H. Devries and D.A. Baylor. *Mosaic Arrangement of Ganglion Cell Receptive Fields in Rabbit Retina.* J Neurophysiology, 78: 2048-2060, 1997.

[6] E.S. Frechette, A. Sher, D. Petrusca Grivich, A.M. Litke, and E.J. Chichilnisky. *Fidelity of the Ensemble Code for Visual Motion in Primate Retina*. J. Neurophysiology, 94:119-135, 2005.

[7] S. Grossberg, E. Mingolla, and W.D. Ross. *Visual Brain and Visual Perception: How does the Cortex do Perceptual Grouping*. Trends in Neurosciences, 20(3):106-111, 1997.

[8] K. Guillory, S. Shoham, R. Normann. *Discrete Stimulus Estimation from Neural Responses in the Turtle Retina*. Vision Research, 46: 1876–1885, 2006.

[9] N.E. Medeiros and C.A. Curcio. *Preservation of Ganglion Cell Layer Neurons in Age-Related Macular Degeneration*. Investigative Ophthalmology & Visual Science, 42(3):795-803, 2001.

[10] Rebecca L. Rockhill, Frank J. Daly, Margaret A. MacNeil, Solange P. Brown, and Richard H. Masland. *The Diversity of Ganglion Cells in a Mammalian Retina*. J. Neuroscience, 22(9):3831-3843, 2002.

[11] Garrett B. Stanley, Fei F. Li, and Yang Dan. *Reconstruction of Natural Scenes from Ensemble Responses in the Lateral Geniculate Nucleus*. J. Neuroscience, 19(18):8036-8042, 1999.

[12] W. Truccolo, U.T. Eden, M.R. Fellows, J.P. Donoghue and E.N. Brown. *A Point Process Framework for Relating Neural Spiking Activity to Spiking History Neural Ensemble, and Extrinsic Covariate Effects*. J Neurophysiology 93: 1074-1089, 2005.