Estimating Light Edge Velocity Based on Retinal Ganglion Cell Spike Trains

by

Adam Eisenman

B.S. Electrical Engineering, 2005
Minor Mathematics

GEORGIA 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 Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

Author . . . .
Department of Electrical Engineering and Computer Science
May 23, 2007

Certified by . . . .
John L. Wyatt
Professor
Thesis Supervisor

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

Estimating Light Edge Velocity Based on Retinal Ganglion Cell Spike Trains

by

ADAM EISENMAN

B.S. Electrical Engineering, 2005

Minor Mathematics

GEORGIA INSTITUTE OF TECHNOLOGY

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

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 response measured from the retinal ganglion cells (RGCs). We present various algorithms for doing so and present sensitivity analysis of such algorithms. We test the performance of the algorithms on data recorded from retina and on data produced by simulation. We find that we are able to extract enough information about the edge velocity from ON and OFF RGCs when the edge of light is wide. However, our best algorithm's performance decays significantly as the edge of light gets narrower. This leads us to develop algorithms that use ON-OFF directionally selective (DS) cells in conjunction with non-directional ON and OFF cells to produce better estimates of the velocity for narrow edges of light. In addition, we develop a model to simulate the response of a DS cell to 1-dimensional light motion.

Thesis Supervisor: John L. Wyatt
Title: Professor

Acknowledgements

There do not exist words which are magnificent or fitting enough to explain to my parents, **Dorita and Bruce Eisenman**, how thankful I am for being their son. They have provided me with just the right amount of everything, including love, education, morals, and thoughtfulness. I thank you with all my heart. In particular, I would like to elaborate on the major influences each one of them has had on me. My father is an exemplary one, always attentive to every one of my actions. I feel that he has lived his youth once, and again through me. He was the source of my scientific curiosity (in particular, I remember his teaching me the Pythagorean Theorem while I was only in second grade; I didn't understand a thing he said). He brainwashed me from the moment I was born with the ideal of being a "Ramblin' Wreck from Georgia Tech" and a "Hell of an Engineer". I'm happy to say that now I am one. My beautiful mother taught me how to be a good human and to always believe in myself. I understand how much it hurts her for me to be away from Panama, but she has withstood the pain in order to let me fulfill my dreams. Thanks! In addition to this, she taught me all about "cosquillitas," one of the fine pleasures in life.

To my little brother Benji: As I am your role model in certain aspects, you are mine in many others, including charisma, joyfulness, and keeping a warm heart. I truly believe that you will accomplish even more than your brother.

To my grandparents, Sarita and Milton Cohen-Henriquez & Esther and Abraham Eisenman, whom I respect immensely, thank you for keeping me in your thoughts.

To my advisor **John Wyatt**: Beside from being one of the sharpest people I have met, I consider you an excellent friend. Thanks for all the personal time you dedicate to your students; thanks for each and every time you walked in to show me a problem that you thought I would like; and thanks for not letting things go to "hell in a hand basket." In particular, I would like to highlight that the probability of Prof. Wyatt being available to help me when I need it is 1. Without your constant dedication to me, this document would not have been possible.

To **Shelley, Steve, and Karl**: Your expertise and patience have made the ex-

perimental work for this thesis possible. I really appreciate how generous you have been with your time. Above all, I would like to highlight Shelley's kindness and infinite knowledge of DS cells, Steve's patience and experimental expertise, and Karl's coolness and care. Thanks for being such good people.

Many thanks to Professor Al Oppenheim, my academic advisor, for teaching me so much about MIT and building a small ladder to enter it safely; to Professor Berthold Horn for his wise advice, tricks on least-squares methods, and for having the patience to discuss algorithm ideas with me; to Professor Polina Goland for giving me the opportunity to TA 6.041 my first semester here: you saved me!

To **Stavros 'Malaka' Valavanis**: You deserve a lot of respect, and you have mine. I want to acknowledge that all the theoretical developments in the thesis were developed by Stavros and I jointly. Almost everything in this thesis has a great chunk of Stavros in it. You are the spike-sorting beast. Man, we had some fun times in the office, Miami, restaurants, parties, etc. I will never forget the crazy singing, screaming, joking, plate-throwing and dancing in the office. Kalimerachta!

To my crazy KGs: Alex, Antonio, and Danny; each of you provided a very important piece of my MIT experience. Particularly, Alex provided the musical genius and constant willingness to go out, Antonio the alcohol, parties, coffee times and lunches, and Danny the lesson that anything is good to eat, the more expired the better. Thanks for being so good to me!

To the members of 'La Familia' (Nash, Mayer, Jakes, Manush, Esteban, Vos, Helmut, and Jorge), you have always believed in me, pushed me forward, and entertained me like nobody else ever will. You are my brothers forever!

I have left the last paragraph to acknowledge the person who owns my heart: Violeta 'Violulis' Morhaim. Nunca me he sentido tan bien en toda mi vida. Tú me completas, me iluminas, y me llenas de ganas de vivir. Eres la mujer más linda que ha sido creada (a la par de mi madre, jeje). Tu belleza interna amplifica lo irresistible que eres. Gracias por creer en mí. Éste es sólo el comienzo.

To Hashem: Thank you for your blessings.

Adam Eisenman — May, 2007

This Thesis is dedicated to
my grandmother *Mary Nevah de Eisenman* Z"L and
my beloved aunt *Debbie Eisenman Zalzer* Z"L

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> The best scientist is open to experience and begins with romance - the idea that anything is possible.
>
> —RAY BRADBURY

A few years back I became very curious about how information is communicated in real neural systems. I became deeply passionate about making an attempt at understanding the neural code. Through the Boston Retinal Implant Project, Professor Wyatt offered me the opportunity to pursue my passion. Along the way I also had the opportunity to build up knowledge which would serve a very interesting goal: to restore partial vision to blind people. The main purpose of this thesis is to shed light onto the puzzling field of retinal neural coding. Although we do not establish any particular truth about the meaning of the cell ensemble spike-time statistics, we are able to use the cell spike-times to our advantage in the process of decoding the parameters of a visual stimulus. In addition, we give analytical evidence in favor of the hypothesis that ON-OFF directionally selective[1] (DS) cells provide necessary local motion information about a visual stimulus.

The Boston Retinal Implant Project (see Figure 1-1) serves as a springboard to study retinal neural coding. This project's objective is to restore partial vision in patients with Retinitis Pigmentosa and Macular Degeneration. These particular

[1] Directionally selective cells maximally respond to motion in their preferred direction, barely respond to motion in their null (opposite of preferred) direction, and respond intermediately for directions in between the preferred and null [13]. DS cell physiology will be treated in more detail throughout the thesis.

conditions affect the retinal photoreceptor layer, leaving it dysfunctional in most cases. However, these degenerative diseases leave the ganglion cell layer almost entirely functional [7]. The latter layer is responsible for transmitting the visual information to the brain. Hence, by electrically stimulating the ganglion cell layer, one could obtain effective visual perceptions in the brain .

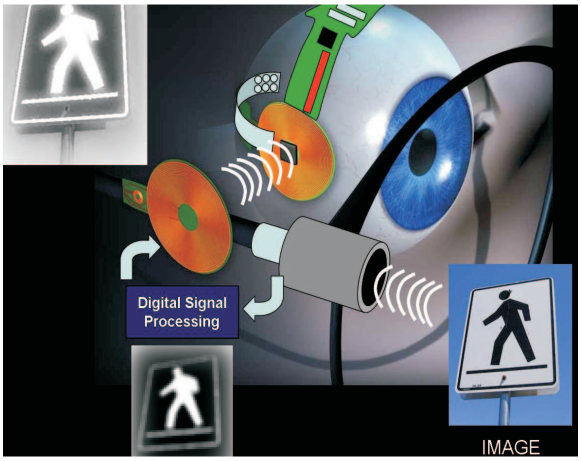

Figure 1-1: The Boston Retinal Implant Concept - from www.bostonretinalimplant.org. Scenical images are recorded by a camera which is mounted on a pair of glasses. This information is processed and electrical stimulus information is then transmitted to the implant chip, contained inside the eye. The chip electrically stimulates the retinal cells using an MEA, and a visual perception is achieved.

Considering how one would perform the coding of the MEA's (Multi-Electrode Array) stimulating signals in a retinal implant, we would like to understand the structure of the stochastic map from optical signals to retinal ganglion cell firing. Once this mapping can be modeled accurately, it can be possible to produce a coding scheme that can be decoded by the brain. More explicitly, we know that retinal ganglion cells are the only retinal cells that feed signals to the brain and that this connection is only feed-forward [3]. Theoretically, this implies that if we were able to replicate the consequential spatio-temporal spiking pattern caused by a light pattern in every one of the ganglion cells of a healthy retina via electrode stimulation, the brain would perceive the correct light pattern.

Another interesting problem that leads us to understand the aforementioned one is what we call the 'Inverse Problem'. The statement of the Inverse Problem is as follows: Given a set of spatio-temporal retinal ganglion cell (RGC) electrical events (i.e. action potential sequences), what can we understand about the video that was shown to the retina? The work contained in this thesis will shed light onto the understanding of the Inverse Problem. 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. In a sense, this process is the inverse of that carried out to stimulate RGCs for an implant, hence its well-deserved name. The estimation process is depicted below in Figure 1-2.

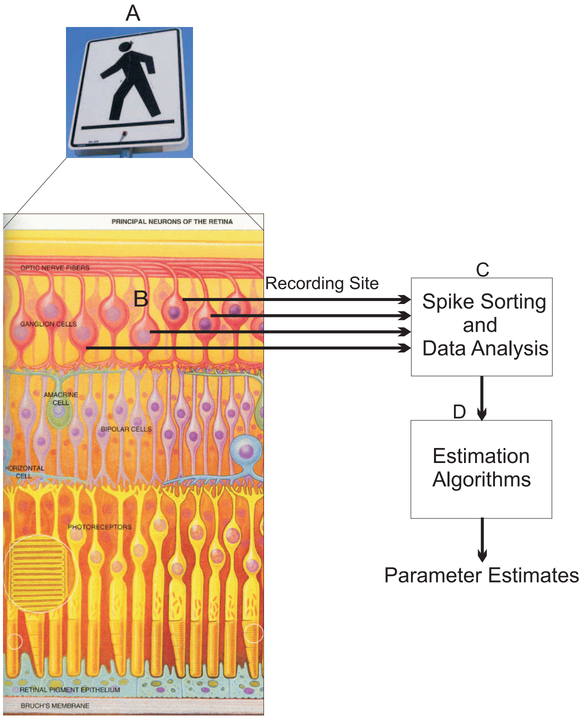

Figure 1-2: Depiction of the process of estimating visual stimulus parameters from RGC recordings. **A)** Image is projected onto the photoreceptor layer, passing through the transparent RGC layer. **B)** Electrical activity of the RGC ensemble is recorded. **C)** Action potentials coming from the cell ensemble are assigned to specific 'units'; then the spike times are used to determine input parameters to the estimation algorithm. **D)** Input parameters coming from **C** are used to produce output parameters which are estimates describing the visual stimulus from **A**.

In this chapter we present a general description of the problem which we address. Subsequently, we give reasons for using rabbit retina in our experiments and present an outline of the experimental procedures to gather the necessary data. Lastly, we describe the methodology which is implemented to extract the information from the data.

1.1 Problem Description

In this thesis we focus on one of the simplest statements of an inverse problem. We wish to make estimates of the speed and direction of a moving edge of light, which is focused on the photoreceptor layer of a piece of rabbit retina. These estimates are made merely by analyzing the occurrence times of action potentials produced by a subset of the retinal ganglion cell ensemble of the retinal piece. The speed and direction of the edge is constant throughout the time of motion. The luminance intensity which we call BRIGHT or ON is held constant, as is the luminous intensity which we call DARK or OFF. We stimulate the retinal piece with broad and narrow edges of light. In the case of broad edges, which we call **curtains**, they are either bright over a dark background or dark over a bright background. In the case of narrow edges, which we call **bars**, they are bright over a dark background. The bars are finite in length, and thus have a leading and a trailing edge. Due to this property, a single bar causes an ON effect followed by and OFF effect on the retina.

Throughout the thesis we make claims about how well we are able to estimate the speed and direction of a moving edge based only on RGC spike times. As action potentials from the same cell are considered stereotypical [8], all of the visual information is encoded in their occurrence times. We show that the information in the spike times is sufficient to make very good estimates ($\sim 5\%$ error) as long as the moving light edge is broad. We predict and confirm that the estimates get much worse as the width of the light edge is decremented. Then we proceed to make the estimates better for narrow edges of light (bars) by taking advantage of the directional information which ON-OFF DS cells provide. However, we *do not* claim that the brain implements algorithms similar to the ones we propose.

1.2 Experimental Setup

New Zealand white rabbits were chosen as the experimental subjects for the following reasons: 1) The rabbit retina has been studied extensively; its anatomy and physiology are well understood relative to those of other animals. 2) The structure of the rabbit's retina is similar to the human retina in many ways. 3) The physiology of rabbit ON-OFF DS cells is well understood.

In order to record action potentials from RGCs we use a square (with corner electrodes removed), 60-electrode MEA which is placed on the RGC layer of the rabbit retinal piece. Given that a single electrode can record action potentials from multiple RGCs, we are required to assign each action potential waveform to a particular cell. We make these assignments by projecting all waveforms measured on a given electrode onto the principal components[2] that they define as a group. Subsequently, we perform clustering in the space of their first three principal components. Each cluster corresponds to a cell proximate to the adjacent electrode.

A further description of the process required to record RGC action potentials in response to visual stimuli is found in the *Experimental Procedures* chapter.

1.3 Extracting Information From RGC Firing Patterns

We make estimates of speed and direction by focusing on two different effects caused on the RGCs by the moving light edge. The first effect type is non-directional, and is caused on a cell when sudden changes of brightness occur over its receptive field. In particular, ON cells react to changes from dark to bright, OFF cells react to changes from bright to dark, and ON-OFF cells react to both types of changes. There are many cells which are not directionally selective which respond robustly to motion in any direction. The response types of these cells are many times transient (brief firing burst), as seen in Figure 1-3. Other cells respond more sustainedly (extended

[2] Principal component analysis simplifies a dataset for analysis by reducing its dimensionality.

firing burst), as seen in Figure 1-4. Both of these figures contain Peri Stimulus Time Histograms (PSTH). A PSTH is an average firing rate computed over several trials of the same stimulus. The PSTH is computed by binning spikes in time.

The second effect which we use to make the estimates is directional. DS cells respond robustly to motion in their preferred direction, weakly to motion in their null direction, and intermediately for directions in between. Based on this phenomenon we make a plot (displayed in Figure 1-5) of the average number of spikes that an ON-OFF DS cell fires for motion of a narrow bars over its receptive field in 8 different directions. For each direction, the average is computed over 3 trials.

1.3.1 Using Non-DS Cell Information to Estimate Speed and Direction

If one is able to find the positions of the cells' receptive field centers, it seems plausible to use ON, OFF and ON-OFF cells which have a transient-type response similar to the one shown in Figure 1-3 to make estimates of speed and direction. Since these cells' responses are very localized in time, they provide much information about where the curtain was at the time they fired. For example, the cell in the aforementioned figure responds very early to motion of a down-moving OFF edge, and very late to motion of an up-moving OFF edge. Given that this cell lies in the upper portion of the image projection area, these response characteristics make sense. Contrastingly, cells which respond similarly to the response seen in in Figure 1-4 do not provide us with localized information.

1.3.2 Using DS Cell Information to Estimate Speed and Direction

Given that ON-OFF DS cells respond directionally to the motion of a bar, as portrayed in Figure 1-5, it seems plausible to obtain directional information about a moving bar based on ON-OFF DS cell responses. For example, if a bar was moved exactly over the receptive field of an ON-OFF DS Cell and the cell responded very

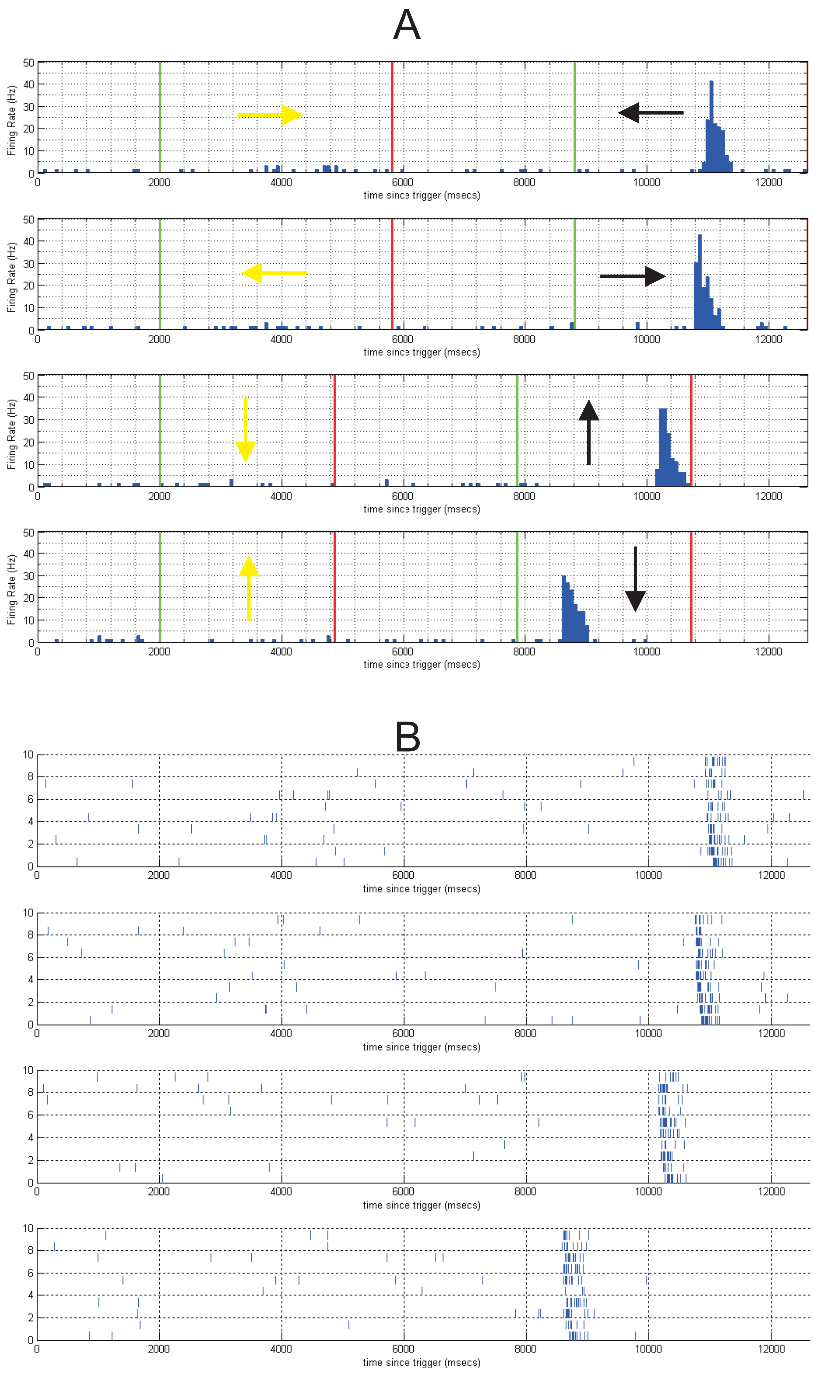

Figure 1-3: Example Response of a Transient OFF Cell to Motion in **4** Directions. **A)** Peri Stimulus Time Histogram (PSTH) computed over **10** trials of the response of a transient OFF cell to the motion of an ON curtain in **4** directions and the motion of an OFF curtain in **4** directions. Yellow 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 stop moving. **B)** The cell's response to the same stimulus repeated **10** times. Each spike train that lies within a green and red line corresponds to a single trial of a moving curtain.

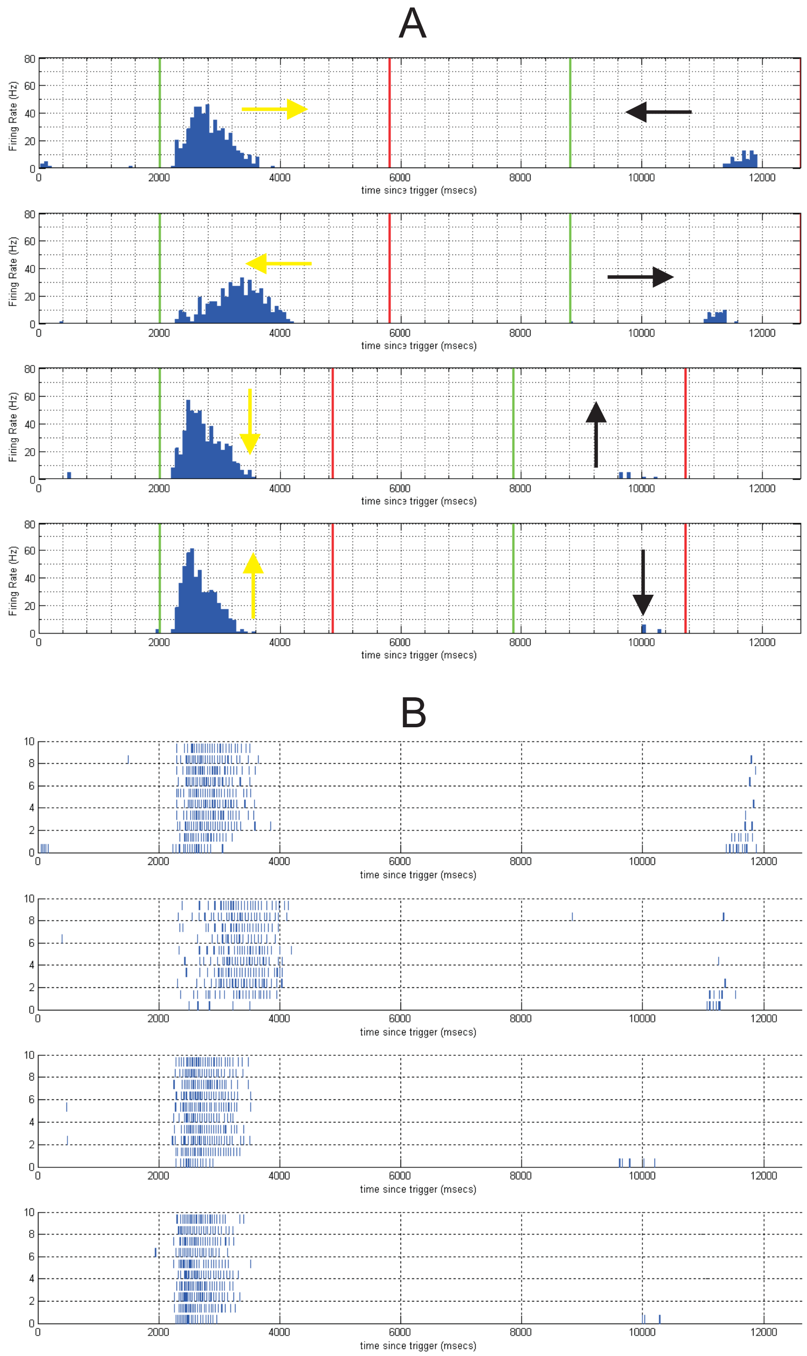

Figure 1-4: Example Response of a Sustained ON Cell to Motion in **4** Directions. **A)** Peri Stimulus Time Histogram (PSTH) computed over **10** trials of the response of a transient OFF cell to the motion of an ON curtain in **4** directions and the motion of an OFF curtain in **4** directions. Yellow 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 stop moving. **B)** The cell's response to the same stimulus repeated **10** times. Each spike train that lies within a green and red line corresponds to a single trial of a moving curtain.

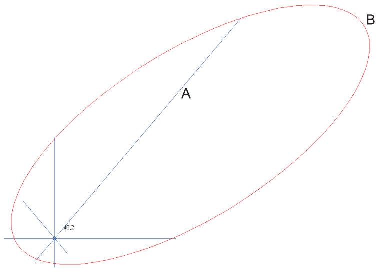

Figure 1-5: Response of an ON-OFF DS cell for motion of a bar in 8 different directions. **A)** Each blue line represents the average number of spikes fired for motion of a narrow bar over the receptive field of this ON-OFF DS cell. The leading and trailing edges of the bar cause an ON and OFF effect respectively. **B)** We fit the end points of the 8 blue lines with an ellipse (shown in red). The ellipse represents the expected number of spikes for all possible directions.

lightly, it is likely that the bar was moving in a direction close to its null direction. On the other hand, if the cell responds robustly, it is likely that the bar was moving in a direction close to its preferred direction. By comparing how many spikes the DS cell actually fired with the expected number of spikes the cell fires in each direction, one can obtain information about what direction the bar was moving in.

1.4 Estimating Speed and Direction From Simulated Data and Experimental Data

Throughout the *Theoretical Developments* chapter, we propose algorithms to make the estimates of speed and direction of moving curtains and bars. We also carry through an analysis of the noise sensitivities of the estimates as a function of the noise in the measured parameters which we input to the algorithms. Based on the sensitivity analysis and the results of simulating the response of cells to moving edges, we suggest which algorithms should be used for the experimental data.

In the *Data Processing Methods* chapter, we explain the methods that were used to estimate the cells' receptive field center locations, the times between the moments when pairs of cells fired, the DS polar firing plots, and the number of spikes each DS cell fired for motion in each direction. Finally, in the *Results* chapter, we show that on average, as one increases the number of cells used and the maximum distance separating the cells from one another, the estimates for speed and direction of moving curtains get better. We also show that the estimates of speed and direction of moving bars are much worse than for curtain motion when using non-directional cells. However, when the directional information from ON-OFF DS cells is introduced, the estimates of bar direction get far better, as we pay a small price in the quality of bar speed estimates. This small price leads us to suggest a model which extracts speed information from the response of DS cells. Thus, such a model and algorithm are outlined in the chapter *Conclusions and Further Work.*

Chapter 2

Survey of the Current Literature

Here we present a survey of the current literature about the Inverse Problem and about ON-OFF DS cell modeling. This served as a guide to outline methods that had been used in the past and to describe what had already been accomplished. It was used to provide insight into what can be done differently to advance the field.

We show a method which has traditionally been used to approach the Inverse Problem. This method makes use of optimal LTI filters to map from ensemble cell firing rates to image sequence estimates. Lastly, we show a method which has been used to estimate the speed of a moving curtain. In this thesis it is our intent to add on to this by estimating angle along with estimating speed. Introducing this new dimension (estimating angle) adds a lot of complexity to the problem but can also add a lot of insight into the question of what we are able to extract from the information that the RGCs grant us.

2.1 ON-OFF Directional Selective Cell Modeling

Based on the paper by Fried et al. 2005 [2], which describes how the excitatory and inhibitory inputs to DS cells are themselves directionally selective, we are able to create a 1-dimensional (1D) model that reflects the basic characteristics of the mechanisms of DS cell firing patterns. This paper shows that directionality in these cells appears mostly due to three phenomenon: 1) The somas of cells that deliver inhibition (Starburst Amacrine cells) are spatially offset in the direction of the DS cell's preferred direction whereas the somas of cells that deliver excitation (Bipolar cells) are located close to the the dendritic field of the DS cell. Therefore, when a bar is moving in the preferred direction, excitation is delivered before inhibition. When a bar is moving in the null (opposite to preferred) direction, inhibition is delivered before excitation. 2) For an object moving in the preferred direction, inhibitory signals (delivered by starburst amacrine cells) are suppressed by other starburst amacrine cells that are spatially offset in the null direction. 3) Excitatory signals (delivered by Bipolar cells) for movement in the null direction are suppressed by cells that are spatially offset in the preferred direction. This mechanism is depicted in Figure 2-1.

2.2 The Inverse Problem

The problem of decoding the neural code of the retina has been studied multiple times in the past (Lettvin et al. 1959 [5], Warland et al. 1997 [12], Frechette et al. 2005 [1]). 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. This analysis is done in Stanley et al. 1999 [9], and is used to reconstruct natural scenes.

2.2.1 Reconstructing Natural Stimuli Using LTI Filters

In order to reconstruct natural scenes from the response of neurons in LGN. As shown in [9], we are given a set of M cells whose responses are being measured. We bin each neuron's firing rate in response to a set of m-sequence and natural scene stimuli. This gives us a discretized firing rate. Let the firing rate of each neuron be of length N and denote the firing rate of neuron i by $r_i[n], n = 0, ..., N-1$, where $r_i[k]$ = [firing rate of neuron i at time k]. Moreover, let the number of pixels in the visual space be S and denote the stimulus value (between -1 and 1) at pixel i by $s_i[n], n = 0, ..., N-1$, where $s_i[k]$ = [intensity of pixel i at time k]. The stimulus (i.e. a new stimulus) is

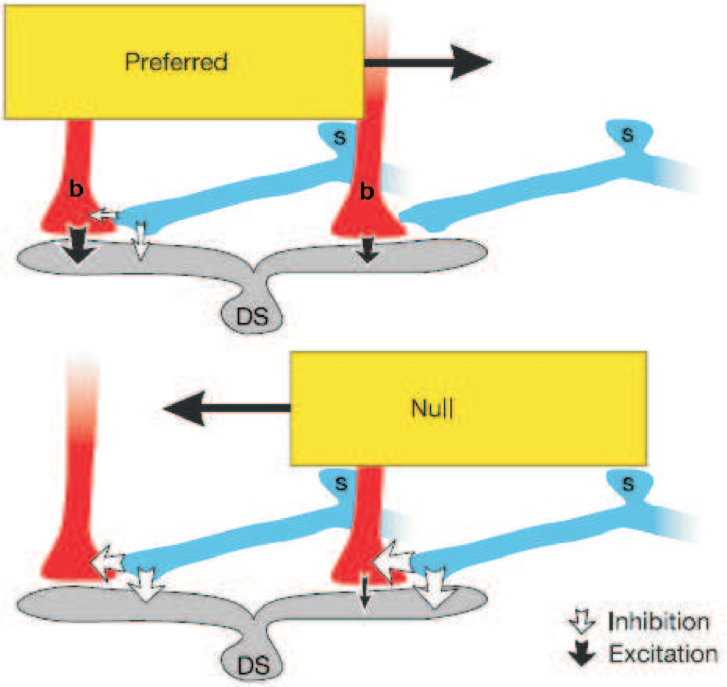

Figure 2-1: Circuitry underlying the DS response. The processes of starburst cells (s, blue) that point in the null direction provide inhibition to DS cell dendrites (DS, grey). Starburst processes respond best to movement away from the cell body, which makes the inhibitory input delivered to DS cells larger for movement in the null direction (bottom panel) than for movement in the preferred direction (top panel). An additional inhibition acts presynaptically to reduce excitation for null direction movement. Although this presynaptic inhibition is depicted as coming from starburst cells, the results do not rule out the existence of another type of cell. The excitatory input to DS cells probably comes from bipolar cells (b, red) and may also have a cholinergic component from other starburst cells. For movement in the null direction, the inhibitory input reaches each subregion of the DS cell ahead of the stimulus edge and therefore before excitation. For movement in the preferred direction, inhibition lags behind excitation. This figure and caption appeared in Fried et al. [2]

then reconstructed using

$$\hat{s}_j[n] = \sum_{i=1}^{M} \sum_{m=-(L-1)}^{L-1} h_{i,j}[m] r_i[n-m] \qquad \text{for} \quad j = 0, ..., S-1 \tag{2.1}$$

where $\hat{s}_j[n]$ is the estimate of the stimulus pixel j at time n, and $h_{i,j}[n]$ is the optimal[1] filter of length $2L-1$ to map the response of cell i to pixel j. If we view the filter $h_{i,j}[n]$ as a column vector composed of $2L-1$ elements, we can express it as:

$$h_{i,j} = \mathbf{R}_{i1}^{-1}\mathbf{p}_{1j} + \mathbf{R}_{i2}^{-1}\mathbf{p}_{2j} + \ldots + \mathbf{R}_{iM}^{-1}\mathbf{p}_{Mj} \tag{2.2}$$

where $\mathbf{R}_{ij}$ is a $(2L-1) \times (2L-1)$ covariance matrix containing the empirical covariances between the response of cell i and the response of cell j at relative lags of $-(L-1), ..., (L-1)$ and $\mathbf{p}_{ij}$ is a column vector of length $2L-1$ that contains the covariances of the response of cell i with stimulus pixel j at relative lags of $-(L-1), ..., (L-1)$.

Figure 2-2 depicts the reconstruction of natural scene videos as done in [9].

2.2.2 Estimating the Speed of a Moving Curtain using Parasol Cells in Macaque Monkey Retina

In addition to reconstructing stimuli by using LTI filters, Frechette et al. adopt another sensible approach [1]. In their paper, estimates of the speed of a moving edge (which we call moving curtain) are made by looking at the delay between the responses of pairs of parasol cells. Their purpose is to find estimates of a specific stimulus parameter (the speed of a moving edge). Their model for finding the time delay between the times at which each parasol cell responds to a moving edge is based on a cross-correlation of the smoothed spike trains of each pair of RGCs. They conclude that given this model, temporal structure in spike trains provided more precise speed estimates than time-varying firing rates and that correlated activity

[1] Optimal in the sense that it minimizes the mean-squared error of the reconstruction over the training set, i.e., over the m-sequence and natural stimuli used to characterize the filters. Note: these filters incorporate the correlation from cell to cell and from cell to pixel.

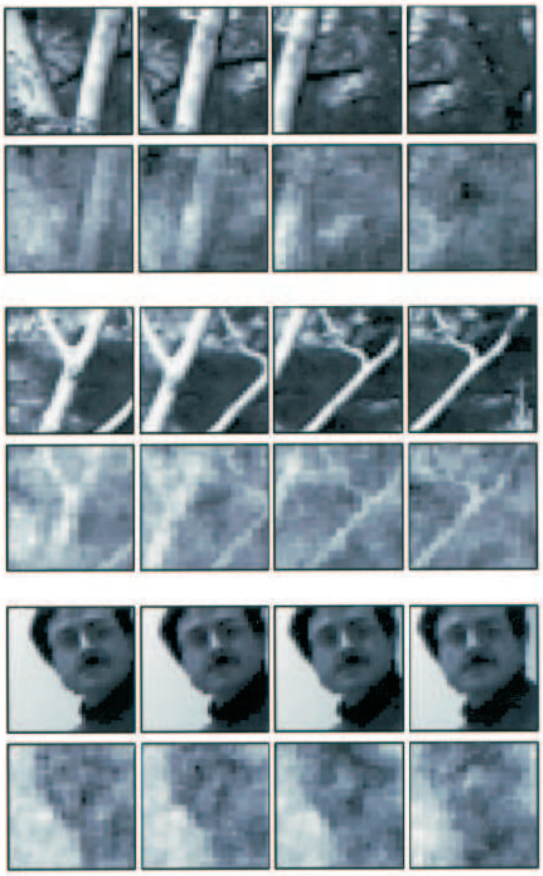

Figure 2-2: Comparison between the actual and the reconstructed images in an area of 6.4° × 6.4°. Each panel shows four consecutive frames (interframe interval, 31.1 msec) of the actual (top) and the reconstructed (bottom) movies. Top panel, Scenes in the woods, with two trunks of trees as the most prominent objects. Middle panel, Scenes in the woods, with smaller tree branches. Bottom panel, A face at slightly different displacements on the screen. This figure and caption appeared in Stanley et al. [9]

between RGCs had little effect on speed estimates. Figure 2-3 depicts the procedure used to estimate the speed.

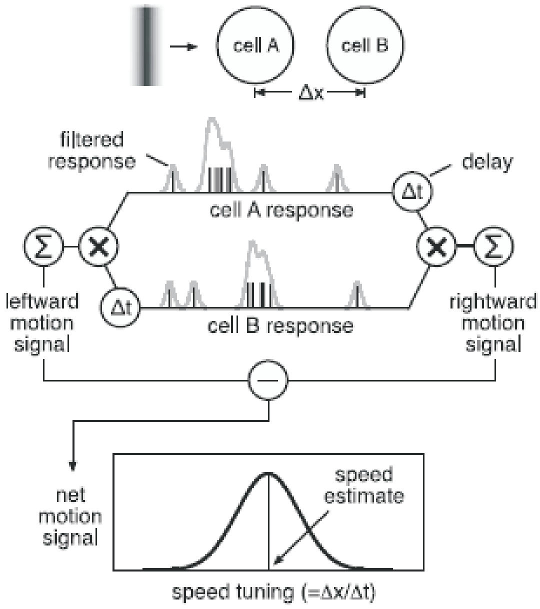

Figure 2-3: The algorithm for estimating bar speed from ensemble RGC activity is depicted schematically, operating on hypothetical spike trains (black ticks) obtained from 2 cells in response to a bar moving from left to right. Each spike train is low-pass filtered in time (gray traces). The filtered response from cell A is delayed by a fixed amount corresponding to the speed tuning and is multiplied pointwise by the filtered response from cell B. The result is summed over time to yield a rightward motion signal. A leftward motion signal is obtained by delaying the response from cell B instead. For multiple cells, all pairwise net motion signals are summed. The speed tuning that yields the maximum net motion signal is used as an estimate of stimulus speed. This figure and caption appeared in Frechette et al. [1]

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.

The experimental set-up procedures and spike waveform analysis described in this chapter are an adaptation of those described in [10], [11].

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.

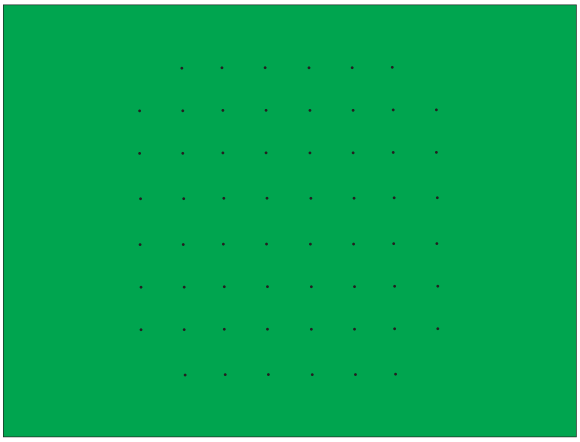

Figure 3-1: MEA and image projection area drawn to scale. The MEA's side has a length of 1.4mm. Each electrode has a 10μm diameter. The shape of the image Projection Area is rectangular (height 2.038mm, width 2.718mm)

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.

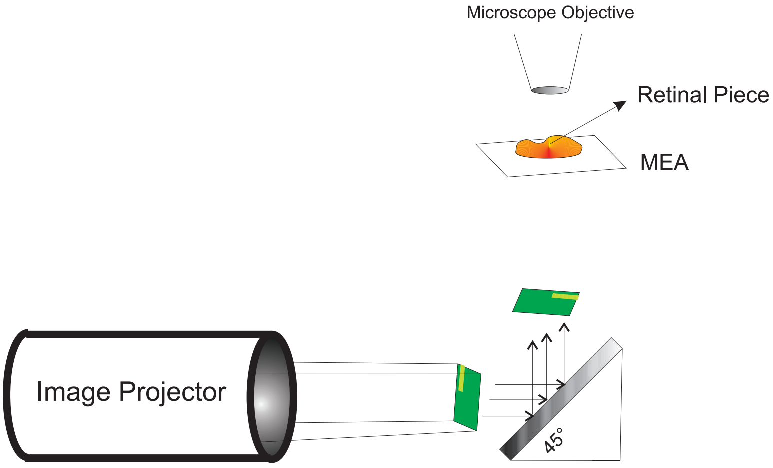

Figure 3-2: The image, produced by the image projector is deflected off of a flat mirror which is inclined at 45°. The image passes through the transparent (the effect of the 10 μm non-transparent electrodes is very small) MEA, and focuses on the photoreceptor layer of the retina. The MEA makes contact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

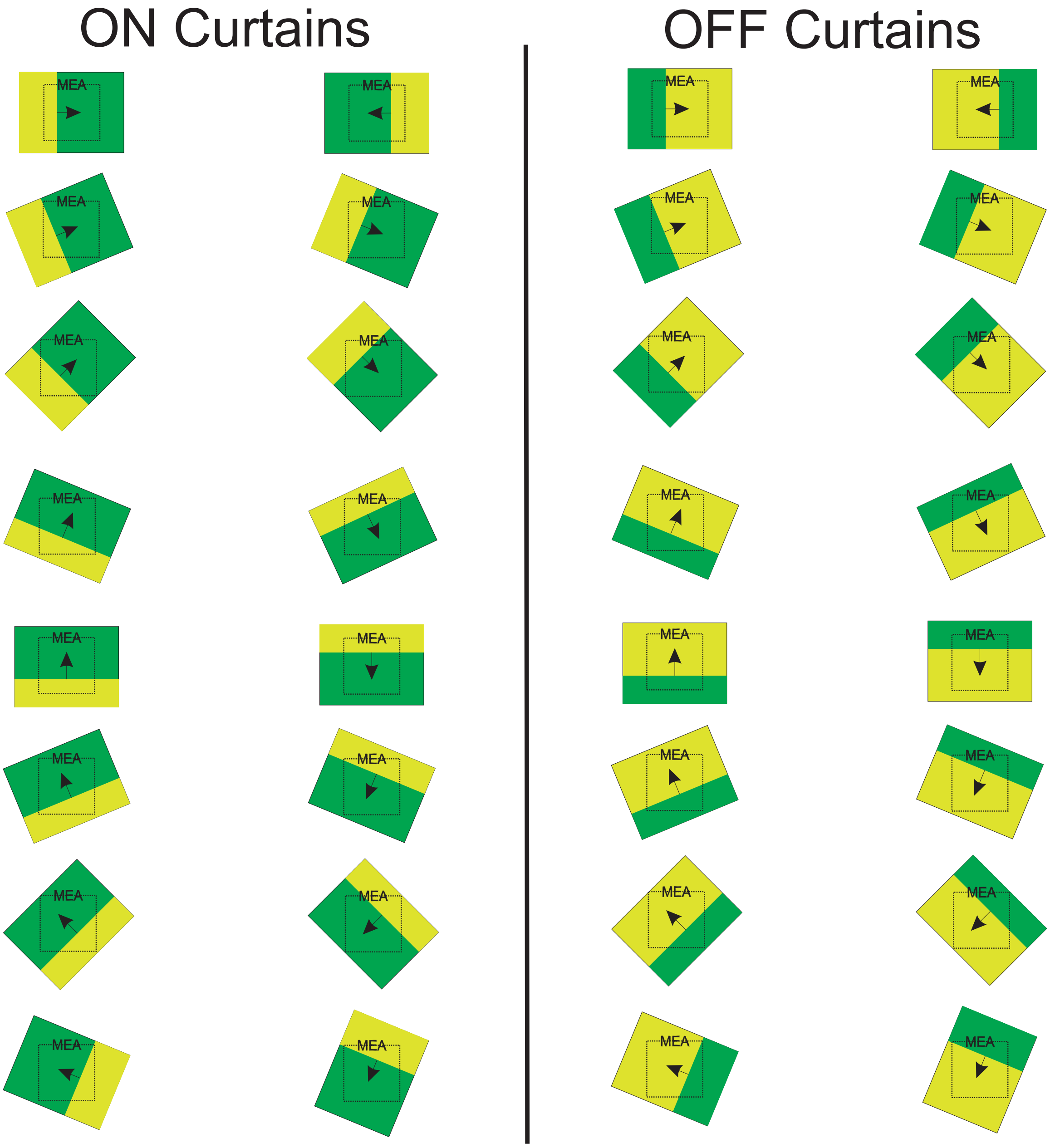

Figure 3-3: The MEA is always fixed. The projected image is rotated with respect to the MEA (by rotating the projection device) to obtain the effect of curtain motion in various directions. The left half of the figure depicts motion of a bright curtain over a darker background (ON effect). The right 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.

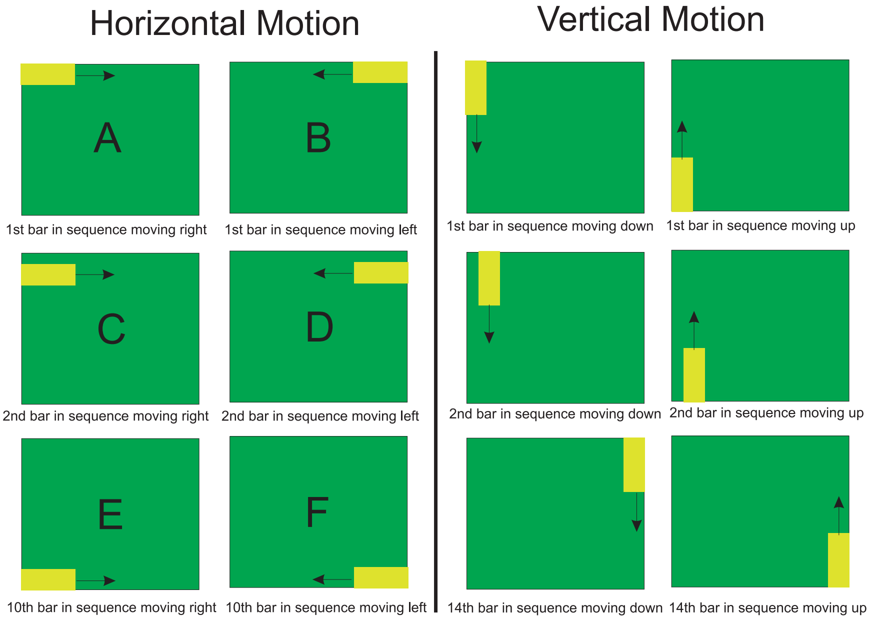
Horizontal Motion
Vertical Motion
A
1st bar in sequence moving right
B
1st bar in sequence moving left
1st bar in sequence moving down
1st bar in sequence moving up
C
2nd bar in sequence moving right
D
2nd bar in sequence moving left
2nd bar in sequence moving down
2nd bar in sequence moving up
E
10th bar in sequence moving right
F
10th bar in sequence moving left
14th bar in sequence moving down
14th bar in sequence moving up

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 [6]. 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 [4], 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.

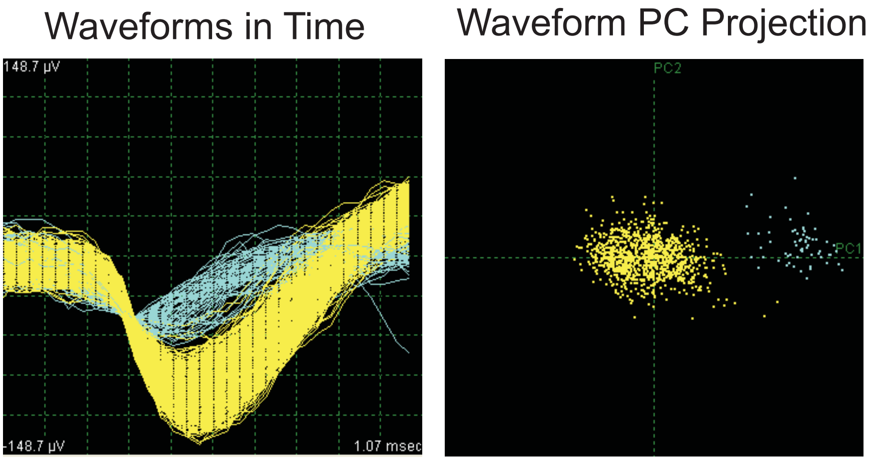

Figure 3-5: On the left of the figure, we see the samples of action potential waveforms coming from two different cells recorded on the same electrode. On the right, we see the projection of each of these transformed vectors onto the 2-D space defined by the first two principal components of the data 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

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 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

[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).

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:

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

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

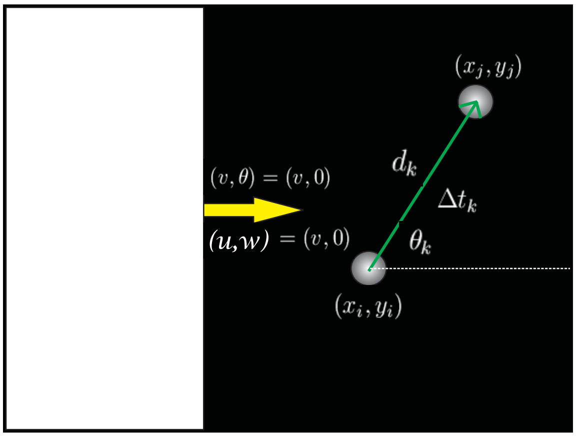

Figure 4-1: Depiction of cell pair, vector from cell i to cell j, time between cell i and cell j, distance between cell i and cell j, and edge velocity vector.

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.

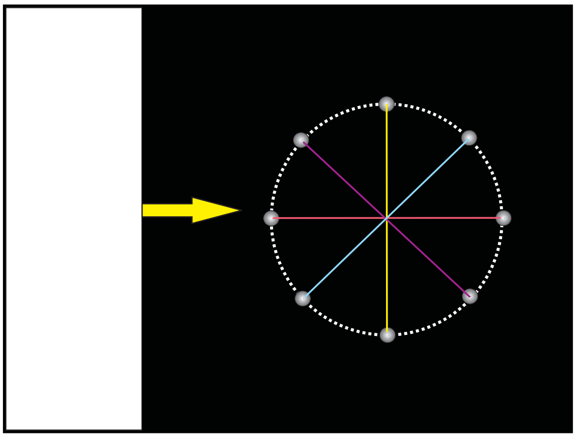

Figure 4-2: $2N$ Cells uniformly placed on a circumference (Here $N = 4$). Each cell is paired up with the cell which is a diameter across from it.

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 Nx$3N$ 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,

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

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

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:

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

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

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

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 $\sum_{i=1}^{N}\cos^2(2\theta_i-2\theta) = \dfrac{N}{2}$ as well.

Furthermore,

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

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.

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

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

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

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 $E[\delta^2_{\Delta x}] = E[\delta^2_{\Delta y}]$.

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 $\sigma^2_{\Delta t}$, δ_{θ_i}'s have a common variance $\sigma^2_{\theta_k}$, and δ_{d_k}'s have a common variance σ^2_d. Now, from 4.21,

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

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

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

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,

$$
\begin{aligned}
Var(\delta_\theta) = \sigma_\theta^2 \approx \frac{1}{N^2}\Bigg(& \frac{v^2}{R^2}\sum_{i=1}^{N}\sin^2(\theta_i-\theta)\sigma_{\Delta t}^2 \\
& + 4\sum_{i=1}^{N} sin^4(\theta_i-\theta)\sigma_{\theta_k}^2 + \frac{1}{4R^2}\sum_{i=1}^{N} sin^2(2\theta_i-2\theta)\sigma_d^2\Bigg) && (4.38)\\
& = \frac{1}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{3}{2}\sigma_{\theta_k}^2 + \frac{1}{8R^2}\sigma_d^2\right) && (4.39)\\
& = \frac{1}{N}\left(\frac{v^2}{2R^2}\sigma_{\Delta t}^2 + \frac{3\sigma_p^2}{4R^2} + \frac{1}{4R^2}\sigma_p^2\right) && (4.40)\\
& = \boxed{\frac{1}{2NR^2}\left(v^2\sigma_{\Delta t}^2 + 2\sigma_p^2\right)} && (4.41)
\end{aligned}
$$

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 $\sigma^2_{f_k}$ becomes:

$$\boxed{\sigma^2_{f_k} \approx \cos^2(\theta_k - \theta) \cdot 2\sigma^2_p + d^2_k \sin^2(\theta_k - \theta) \cdot \frac{2\sigma^2_p}{d^2_k} + v^2\sigma^2_{\Delta t} = 2\sigma^2_p + v^2\sigma^2_{\Delta t}} \tag{4.44}$$

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:

$$\sigma^2_{f_k} \approx 2\sigma^2_p + 2v^2\sigma^2_{\Delta t} + \frac{\cos^2(\theta_k - \theta) \cdot 8\sigma^4_p}{d^2_k} \tag{4.45}$$

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':

$$u' = \frac{1}{a_1b_2 - a_2b_1}(b_2\Delta t_1 - b_1\Delta t_2) \qquad w' = \frac{1}{a_1b_2 - a_2b_1}(-a_2\Delta t_1 + a_1\Delta t_2) \tag{4.51}$$

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

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

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

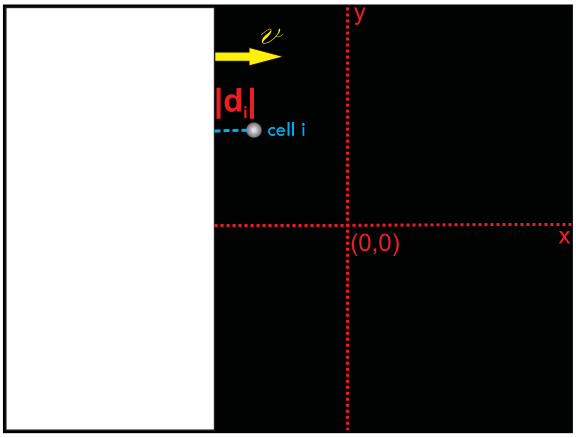

Figure 4-3: This is a picture taken at time t_i (the moment at which cell i fires). It can be seen that the edge is not over cell i's RF center at this moment. The timing error in this picture is then $\frac{d_i}{v}$.

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

where

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

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

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

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:

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

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:

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

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

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

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:

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

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,

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

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

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

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

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

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

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 $(d_k \cos(\theta_k), d_k \sin(\theta_k), \Delta t_k)$.

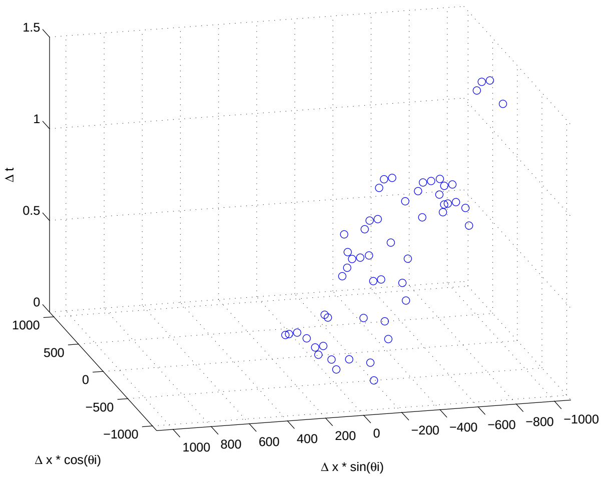

Figure 4-4: Location of points in 3D space. It can be seen that the points approximately lye in a hyperplane, as expected. This would be seen more easily if we were able to rotate the axes.

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

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

$\mathbf{C}$ 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

$$\begin{aligned} \begin{bmatrix} \delta_\alpha \\ \delta_\beta \end{bmatrix} &\approx (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C}\underline{\boldsymbol{\delta\lambda}} \\ &= \frac{1}{2R^2N} \begin{bmatrix} -\sum_{i=1}^{N} d_i(\alpha\cos^2(\theta_i) + \beta\frac{1}{2}\sin(2\theta_i))\delta_{d_i} + \sum_{i=1}^{N} d_i^2(\alpha\frac{1}{2}\sin(2\theta_i) - \beta\cos^2(\theta_i))\delta_{\theta_i} + \sum_{i=1}^{N} d_i\cos(\theta_i)\delta_{\Delta t_i} \\ -\sum_{i=1}^{N} d_i(\alpha\frac{1}{2}\sin(2\theta_i) + \beta\sin^2(\theta_i))\delta_{d_i} + \sum_{i=1}^{N} d_i^2(\alpha\sin^2(\theta_i) - \beta\frac{1}{2}\sin(2\theta_i))\delta_{\theta_i} + \sum_{i=1}^{N} d_i\sin(\theta_i)\delta_{\Delta t_i} \end{bmatrix} \end{aligned} \tag{4.89}$$

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 $(v_{min}, \ \theta_{min})$ 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 $\sigma_p = 100\mu\text{m}$, and $\sigma_{\Delta t} = 0.1\text{sec}$. 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 $300 - 3000\mu\text{m/sec}$ 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 $300\mu\text{m/sec}$, and we do not expect to be able to estimate speeds and directions accurately for speeds greater than $3000\mu m/sec$.

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

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

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

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

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

Simulations

Simulating the response of RGCs to motion of curtains and bars serves us three purposes: 1) To predict how well the decoding algorithms we propose would do in situations which were not carried out experimentally, 2) To observe how the algorithms' performance on simulated data (which involves assumptions about independence and specific noise distributions) compares to the performance on real data, 3) To verify that the assumptions we make for the theoretical calculations (such as placing the cells on a circumference) are close to simulated situations where the cells are not restricted to lie on a circumference.

In this chapter we describe the assumptions which we made to simulate the behavior of ON, OFF, and ON-OFF cells in reaction to the motion of curtains and fixed length bars of various widths. We also describe the assumptions made to simulate the response of DS cells due to the motion of bars. Next, we present the results of estimating the speed and direction of a moving curtain using the various algorithms described in the chapter *Theoretical Developments.* In addition, we present the results of estimating the motion of bars of various widths. We first make the bar motion parameter estimates by using purely ON, OFF, and ON-OFF cells, and then show that the estimates get better as we include the responses of ON-OFF DS cells into the estimation procedure. All throughout, we show plots that depict the RMS (root mean-squared) error in the estimates as a result of using combinations of various numbers of cells along with various restrictions on the radial extent of the cell locations in the plane. We make comparisons of the algorithms' performance when they are run assuming various possible noise levels (σ^2's) in the measured parameters (i.e. cell locations, firing times, number of spikes fired by DS cells in response to a particular bar).

5.1 Setting Up Simulations of the Responses of ON, OFF, and ON-OFF Non-DS Cells to Curtain Motion

For a particular simulation, we select a radius value R, which defines a circle in which we drop the cells. In addition, we select a number of cells N which are dropped into the circle. For each simulation in the set we drop the cells into the circle at random (uniformly), as seen in Figure 5-1. By construction, since a particular curtain only has one effect (either ON or OFF), we assume that all cells inside the circle respond to a particular curtain type (ON or OFF). Given the curtain speed, direction, and starting time, we calculate the times at which each cell should fire according to the position of their receptive field centers. Next, we add zero-mean Gaussian noise of variance σ_p^2 and σ_t^2 to the cell position coordinates and firing times, respectively (all noise values are selected independently). The noisy position and firing time values of the cell ensemble are handed to the estimation algorithms.

5.2 Setting Up Simulations of the Responses of ON, OFF, and ON-OFF Non-DS Cells to Bar Motion

The responses of ON, OFF, and ON-OFF Non-DS cells to a moving bar are simulated in exactly the same way as in response to moving curtains. However, only cells which are in the strip defined by the motion of the bar (see Figure 5-2) respond. Similarly,

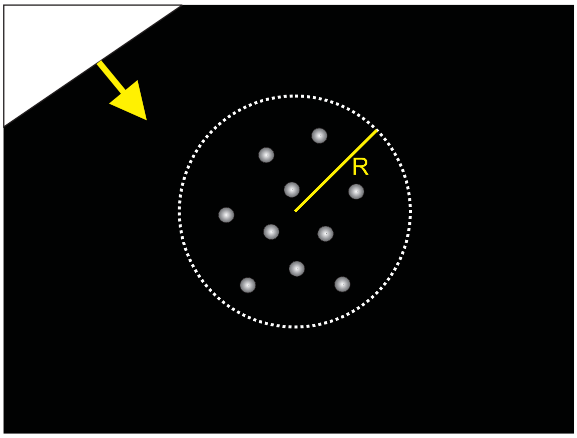

Figure 5-1: Simulating the Response of N Cells Inside a Circle of Radius R, to a Moving Curtain. In this picture $N = 10$.

for a particular simulation, the cells are placed at random (uniformly) inside the strip. As a bar has a leading and a trailing edge, it evokes ON and OFF responses in the cells along its path. Nonetheless, we always simulate either the response of ON and ON-OFF cells to the leading edge (ON effect), or the response of OFF and ON-OFF cells to the trailing edge (OFF effect). We do so because we would like to compare the fidelity of the estimates of speed and direction of a thin moving bar to the estimates of speed and direction of a moving curtain.

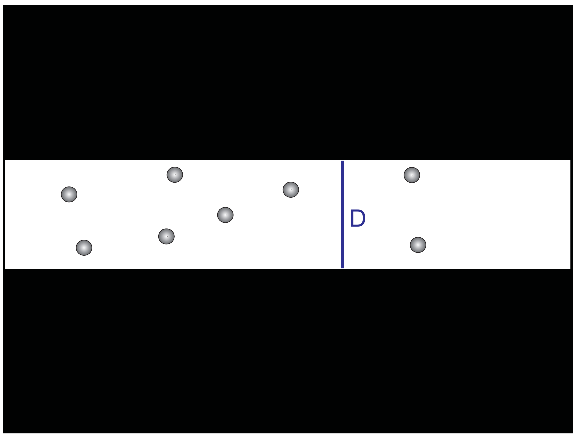

Figure 5-2: Simulating the Response of N Cells Inside a strip of height D, to a moving bar. The white strip represents the path traversed by the moving bar. In this picture $N = 8$.

5.3 Setting Up Simulations of the Responses of ON-OFF DS Cells to Bar Motion

In order to simulate the response of ON-OFF DS Cells to the motion of bars we produce a single parameter — the total number of spikes fired for motion in a particular direction due to the ON *and* OFF effect of the bar. A few assumptions are inherent in the above decision. We assume that regardless of the speed (assuming the speed is in the range of speeds we experiment with), the number of spikes fired for motion in a particular direction is about the same. To decide how many spikes a DS cell fires for motion in a particular direction we do the following: 1) Choose the major and minor axis lengths of an ellipse at random (uniformly); the major axis length is restricted within the range 15 to 30; the minor axis length is restricted within the range 7.5 to 15. 2) Choose the inclination angle of the ellipse defined in 1) at random (uniformly between 0 and 2π). 3) Calculate the shortest distance from the ellipse focus (of the two foci, we refer to the one which is on the negative x-axis before the ellipse is rotated, assuming the ellipse center is at (0,0)) to the ellipse contour in the direction at which the bars is moving. 4) The length found in 3) is the number of spikes that the DS cell fires for motion in the particular direction. By this procedure, in simulation, each function $h_k(\theta)$, defined in the chapter *Theoretical Developments*, is defined by an ellipse chosen at random (see Figure 5-3). In simulation, the value S_k (number of spikes the cell actually fired), also defined in *Theoretical Developments*, is obtained by adding noise to $h_k(\theta_{bar})$, where θ_{bar} is the angle at which the bar is moving. The added noise is zero-mean Gaussian, with standard deviation proportional to $h_k(\theta_{bar})$. Each simulated DS cell's $h_k(\theta)$ represents its polar firing profile. The simulated values of the S_k's (one for each DS cell) along with each DS cell's $h_k(\theta)$ are handed to the estimation algorithms.

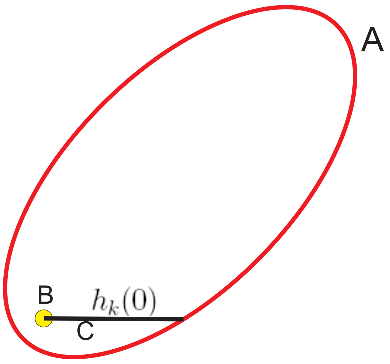

Figure 5-3: DS Cell Polar Plot Example. The DS cell polar firing profile (A) is represented by the red ellipse. This DS cell's "preferred direction" of firing is 45°. The yellow dot (B) denotes the location of the focus (out of the two foci) which is used. As an example, we draw a black line (C) which denotes the segment of distance $h_k(0)$ drawn from the ellipse's focus to the ellipse contour. The length of this segment represents the number of spikes (absent of noise) fired by the DS cell for motion in the direction $\theta = 0$.

5.4 Moving Curtain Simulations

In this section we present the results of simulating the motion of a curtain (ON and OFF effects are simulated equivalently) at fixed speeds of $v = 714\mu m/sec$ and $v = 1428\mu m/sec$. By construction of the simulation environment the direction in which the curtain moves is irrelevant. We calculate the RMS (root mean square) error in the estimates of speed and direction over 300 simulation runs, where the errors are defined to be $(\hat{v} - v_{true})$ and $(\hat{\theta} - \theta_{true})$, respectively. We make plots of the RMS error in the estimates as R (radial extent of cell locations; also half the maximum distance between cells) and N (the number of cells used to make the estimates) are varied. Subsequently, we compare the results of estimating the speed and direction of a curtain to theory, by simulating the response of cells evenly spaced on a circumference. At the end of this section, we compare the estimate fidelity of all treated algorithms.

5.4.1 Estimating Velocity Vector Directly Without Weighing

We present the results of estimating speed and direction based on the algorithm described in the *Theoretical Developments* chapter, "Estimating Velocity Vector Directly" (i.e., equation 4.114). The results shown in Figures 5-4 and 5-5 are for motion of a curtain at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$. We do not reweigh the estimates coming from each two-pairing based on their noise sensitivity. The results of weighing are shown in the next subsection. In the next subsection, we also make a comparison of the estimate fidelity between both methods.

The plots have been interpolated for ease of view. We do not observe an obvious trend in the speed error value as N and R increase, but we observe that for both simulated speeds as R increases, the direction error decays. However, we quickly notice that even in scenarios under which we would expect good estimates (i.e., large number of cells, large radial extent), this algorithm's estimates of speed and direction give very large errors. In large part, the estimate infidelity is due to the effect of estimates coming from two-pairings which provide outlying estimates. It should be noted that this effect could be diminished by picking the median of the estimates coming from each two-pairing instead of the mean. In the next subsection we present the results of weighing the terms in the sum of individual estimates coming from each two-pairing (as seen in Equation 4.55) to notice that the estimates get much better.

5.4.2 Estimating Velocity Vector Directly With Weighing

In this subsection we show the results of estimating the velocity vector directly by assigning weights to each individual estimate (based on a calculation of the estimate's variance), as described in equation 4.55 of the *Theoretical Developments* chapter. The results shown in Figures 5-6 and 5-7 are for motion of a curtain at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$.

It is apparent in the speed estimate plot that as N (number of cells used) and R (radial extent of cell locations) increase, the RMS error of the estimate decreases. This trend is also noticeable in the direction estimate plots for both speeds, however the decrease in RMS error vs. increasing R is not as prevalent as in the speed estimate plots. Since we did not perform an analysis on the sensitivity of the speed and direction estimates for this algorithm, we do not compare the simulation results to any theoretically derived sensitivities. However, we do make this comparison further

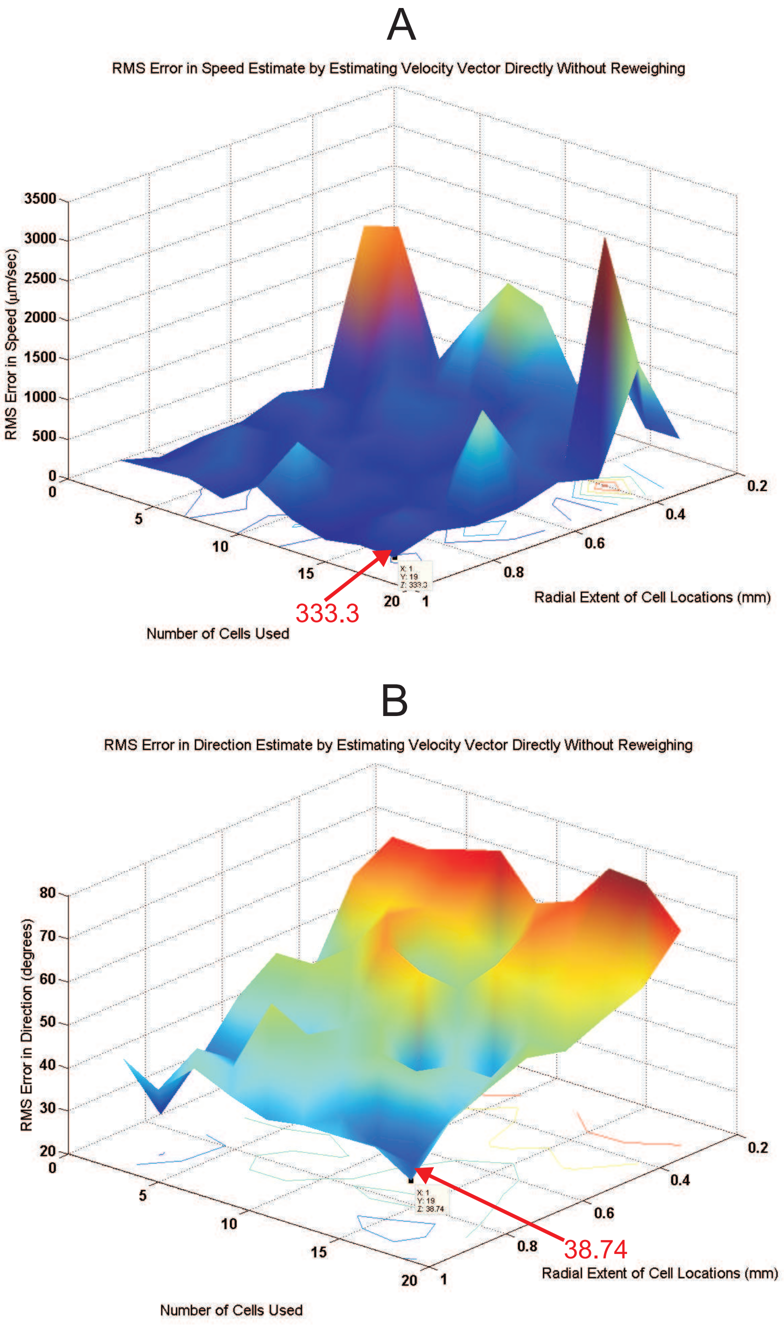

Figure 5-4: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector directly without weighing. The speed of the curtain is $v = 714\mu m/sec$. The errors which are plotted are RMS errors averaged over 100 trials, all independent of each other (i.e., the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data. The reason why we average over 100 trials instead of 300, and why we only simulate up to 19 cells is that the number of two-pairings grows very rapidly with the number of cells which makes the simulation time become too long.

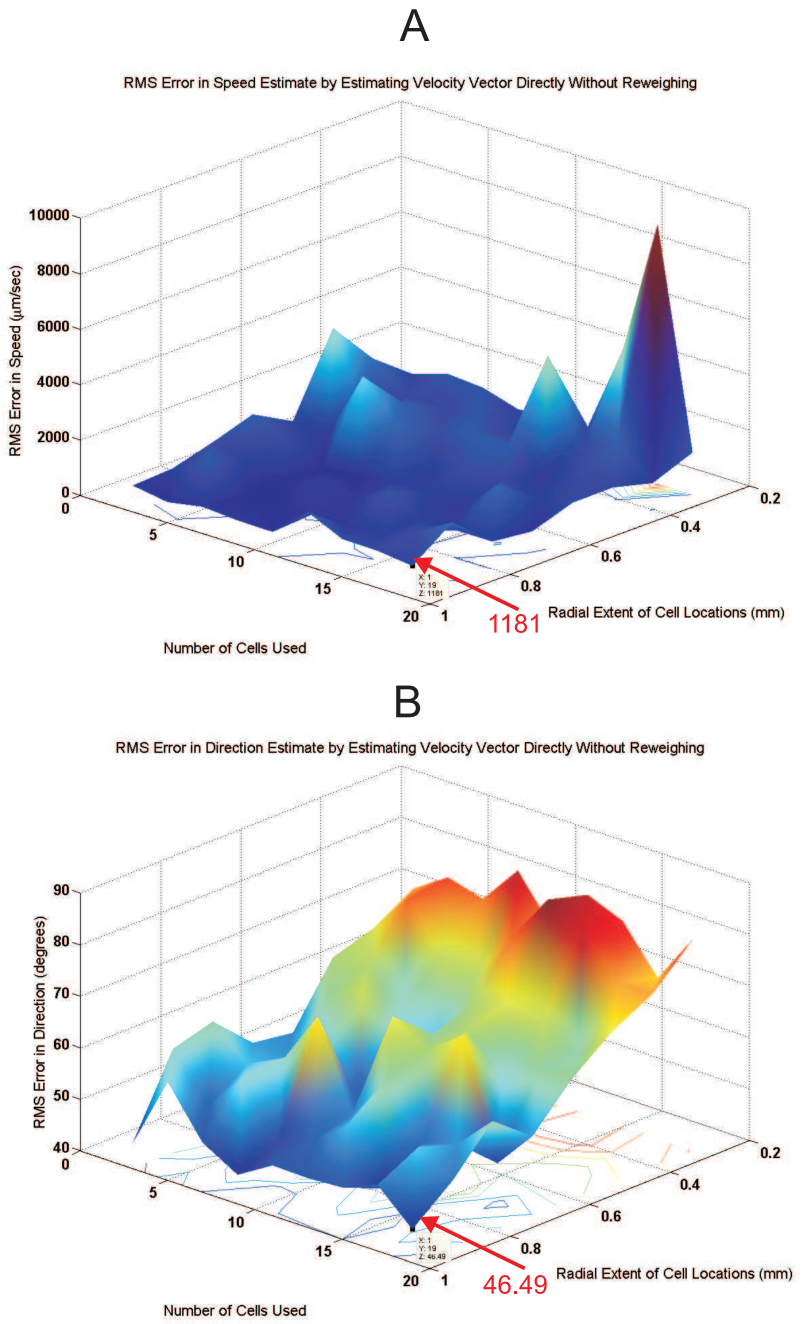

Figure 5-5: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector directly without weighing. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 100 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} =$ 0.141 sec, which reflect the STDs of the errors we get from real data. The reason why we average over 100 trials instead of 300, and why we only simulate up to 19 (vs. 25 for other algorithms) cells is that the number of possible two-pairings grows very rapidly with the number of cells which makes the simulation time become too long.

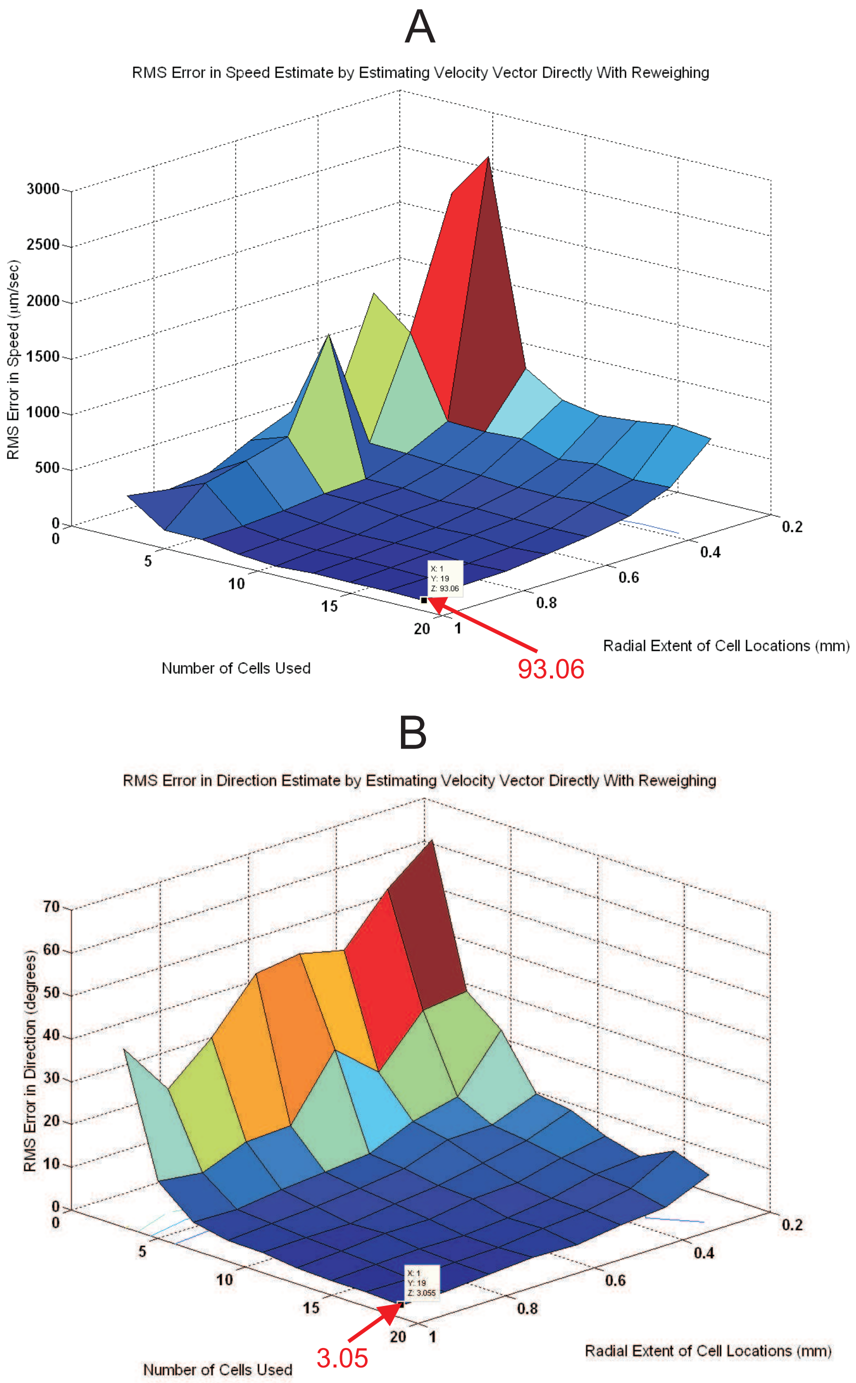

Figure 5-6: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector directly with weighing the terms in the sum. The speed of the curtain is $v = 714\mu m/sec$. The errors which are plotted are RMS errors averaged over 100 trials, all independent of each other (i.e., the cells are replaced inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data. For the same reasons as before we average over 100 trials instead of 300 and simulate only up to 19 cells, instead of 25.

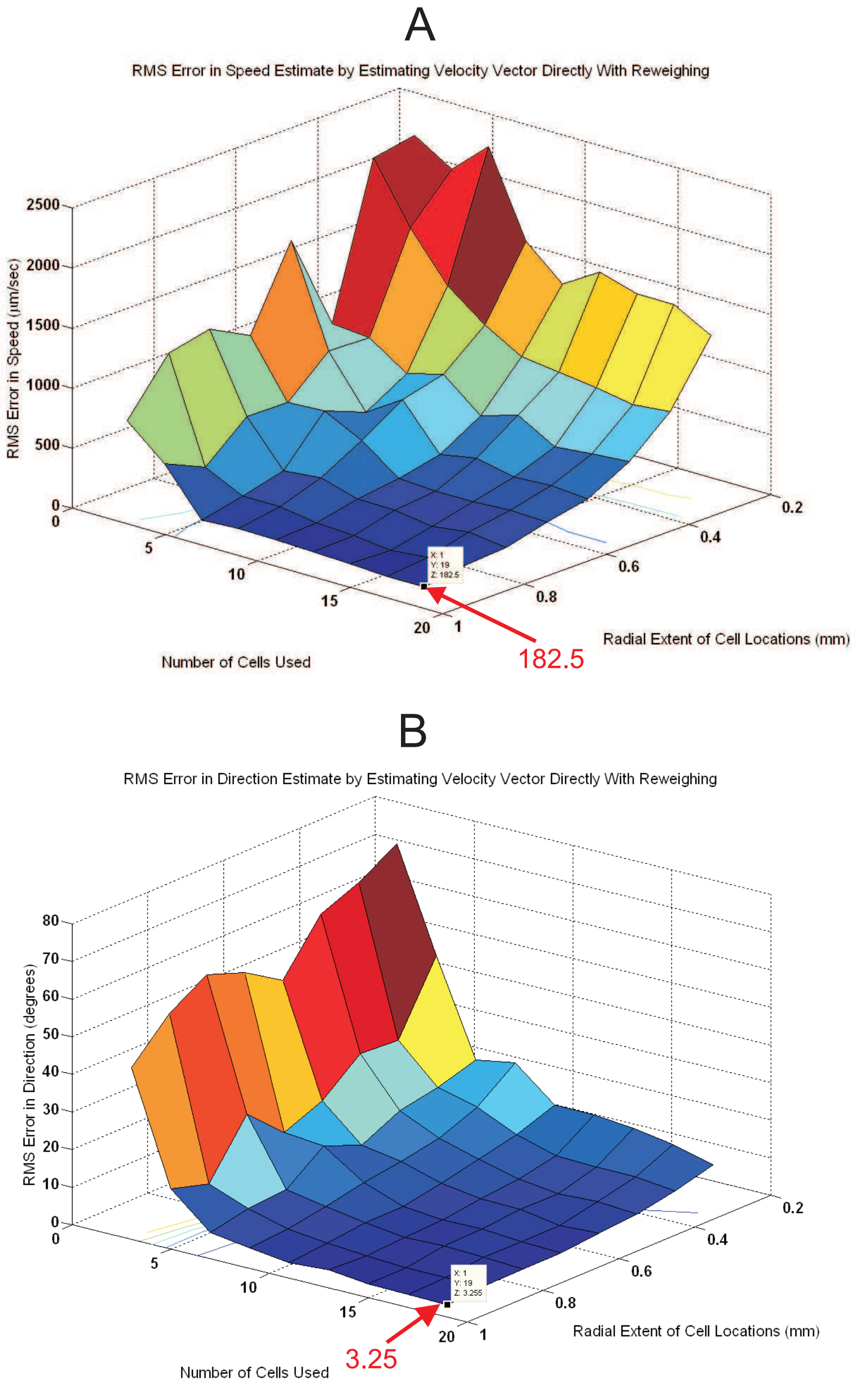

Figure 5-7: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector directly with weighing the terms in the sum. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 100 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data. For the same reasons as before we average over 100 trials instead of 300 and simulate only up to 19 cells, instead of 25.

in this section after we simulate the performance of the CosCos and Newton-Raphson algorithms.

It is worth noting that at least for the largest values of N and R, the estimates of speed and direction improve greatly when the weights are introduced. The dimension of the moving curtains we run during experiments and the number of cells which we typically get responses from are in the neighborhood of these values of N and R. Therefore, the simulations suggest that if we were to pick between the no-weight algorithm and the weighted version to estimate speed and direction, we should pick the latter.

In addition, we note that doubling the speed of the moving curtain causes the estimates of speed and direction to get worse in RMS, as we intuitively expected.

5.4.3 Estimating Velocity Vector Using Global Firing Time Information

In this subsection we show the results of estimating speed and direction by estimating the rectangular components of the velocity vector. We do so by using global firing time information, as described in the corresponding algorithm in the *Theoretical Developments* chapter. The results, shown in Figures 5-8 and 5-9, are for motion of a curtain at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$.

These plots are qualitatively very similar to the results in the previous subsection. We notice the same general trends (decrease in RMS error as we increase R and N). Once again, as the speed is doubled, the RMS errors get worse.

5.4.4 Estimating Speed and Direction by the CosCos Algorithm

In this subsection we show the results of estimating speed and direction by the CosCos algorithm, presented in the *Theoretical Developments* chapter. The results, shown in Figures 5-10 and 5-11, are for motion of a curtain at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$.

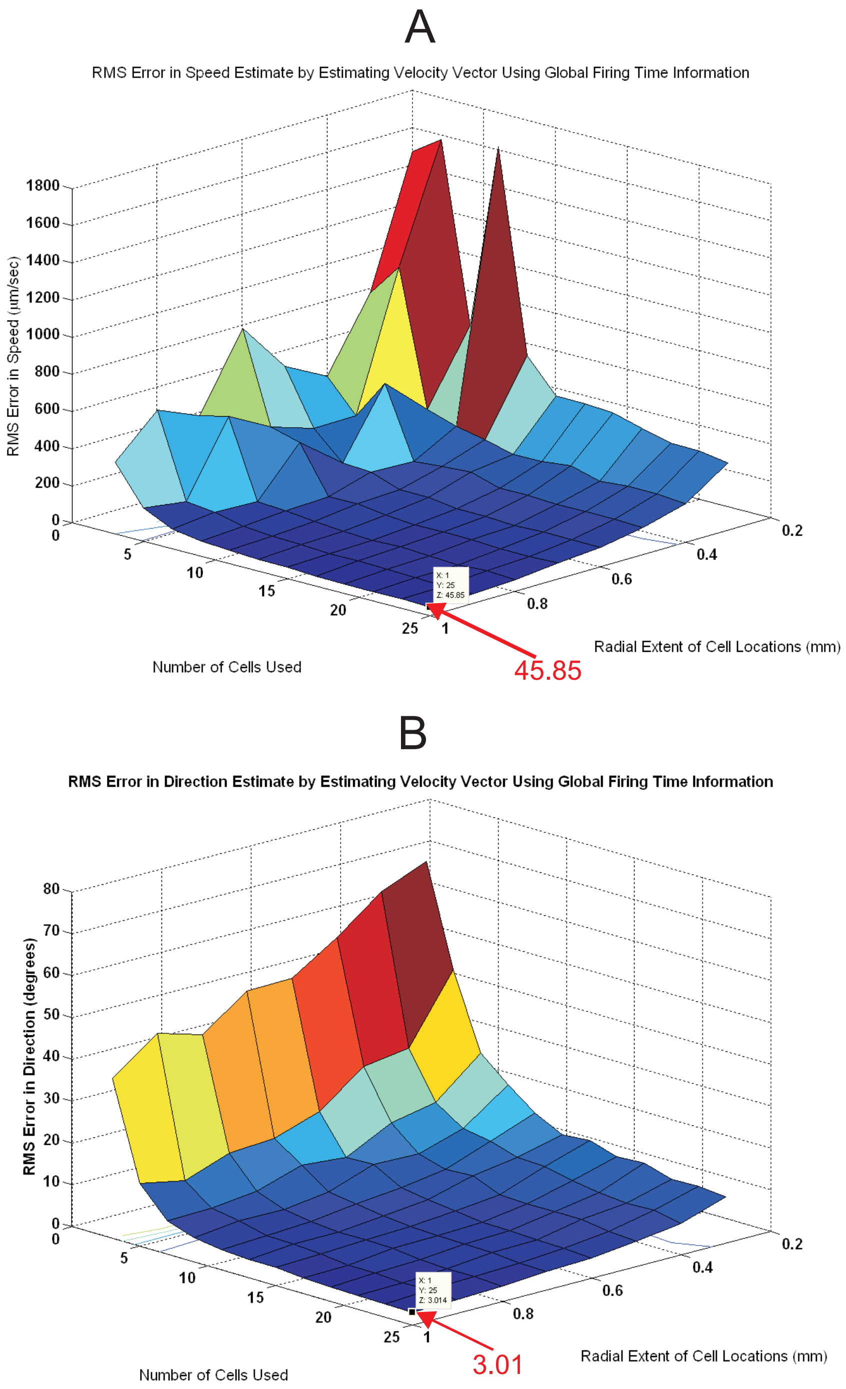

Figure 5-8: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector using global firing time information. The speed of the curtain is $v = 714\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} =$ 0.141 sec, which reflect the STDs of the errors we get from real data.

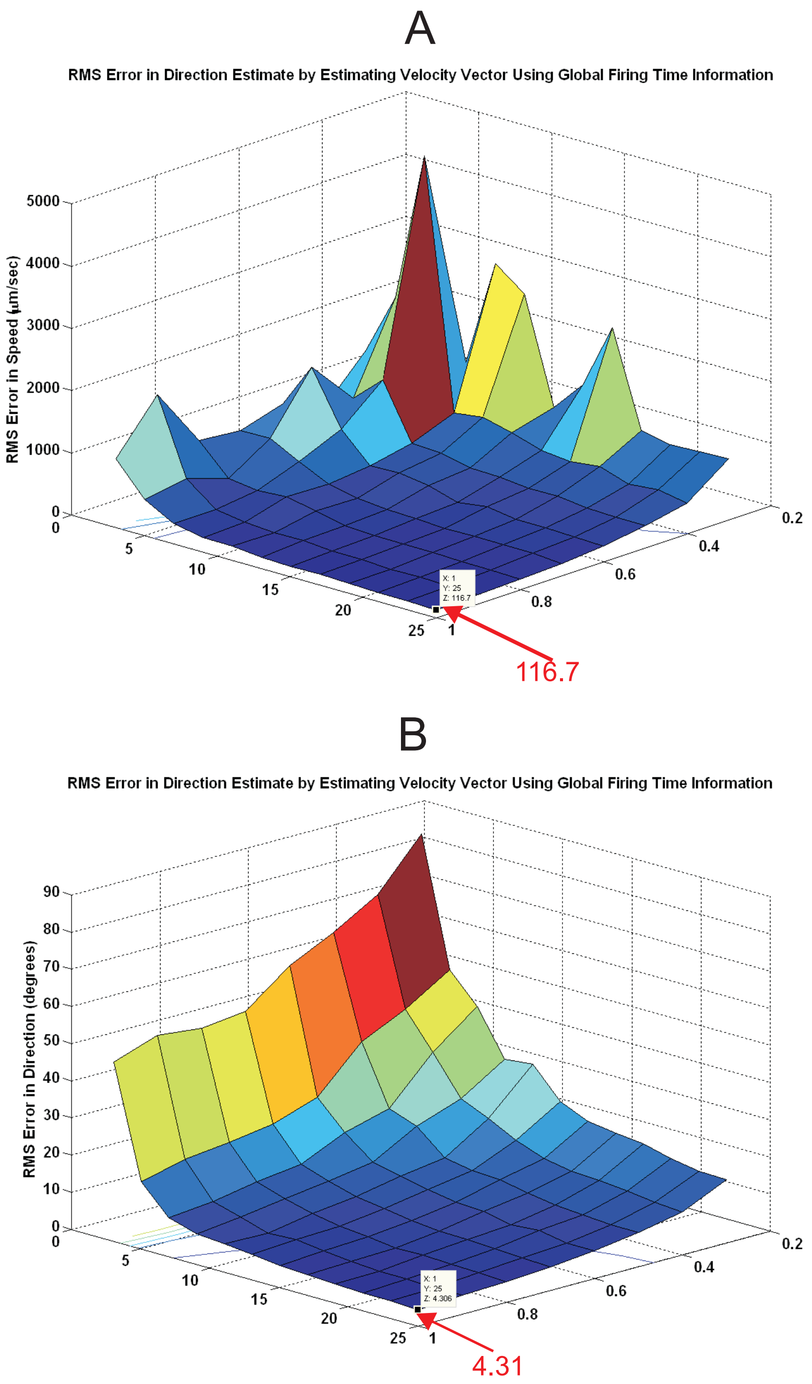

Figure 5-9: **A** and **B** show the errors in estimating speed and direction, respectively, by estimating the velocity vector using global firing time information. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

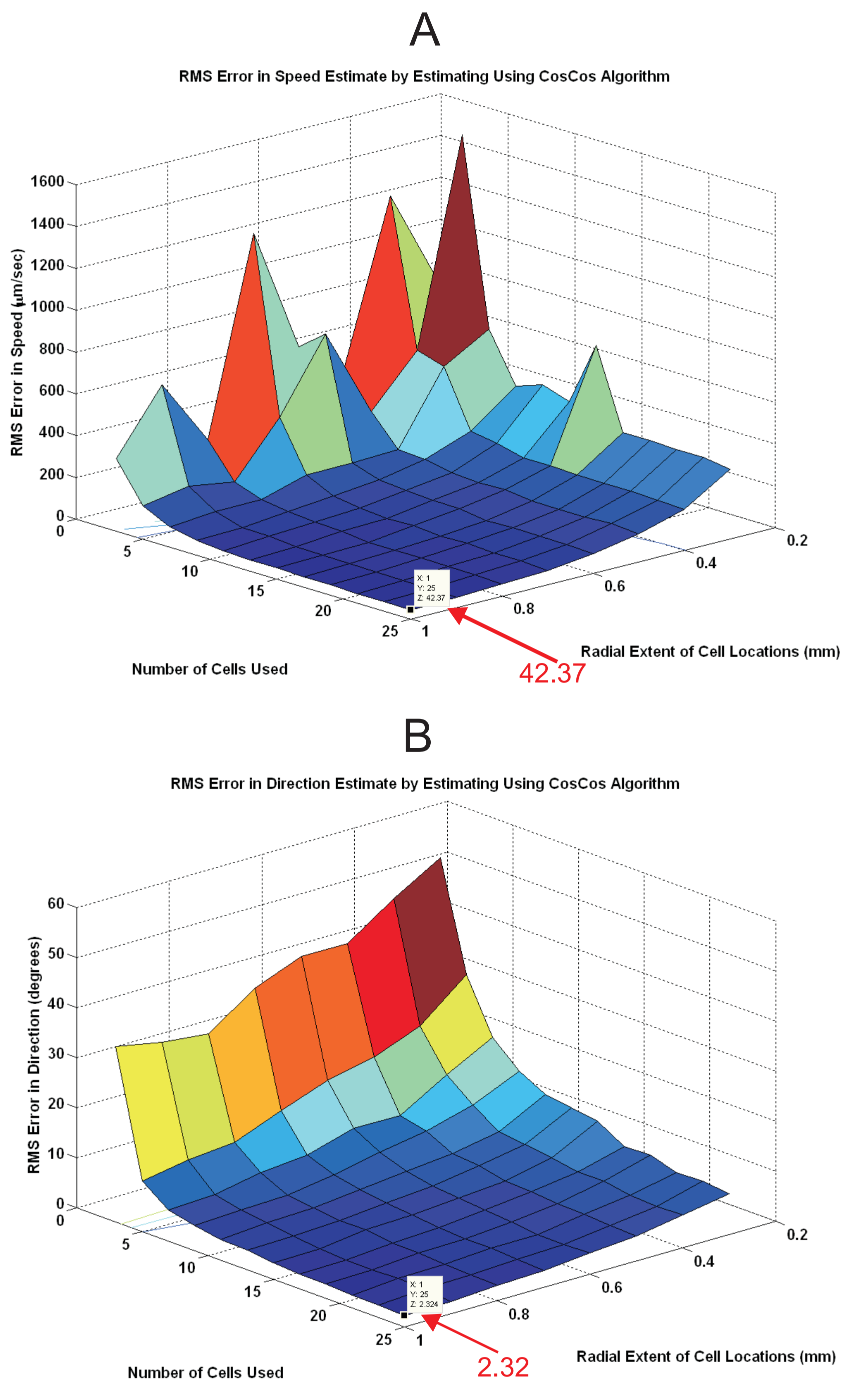

Figure 5-10: **A** and **B** show the errors in estimating speed and direction, respectively, by using the CosCos algorithm. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

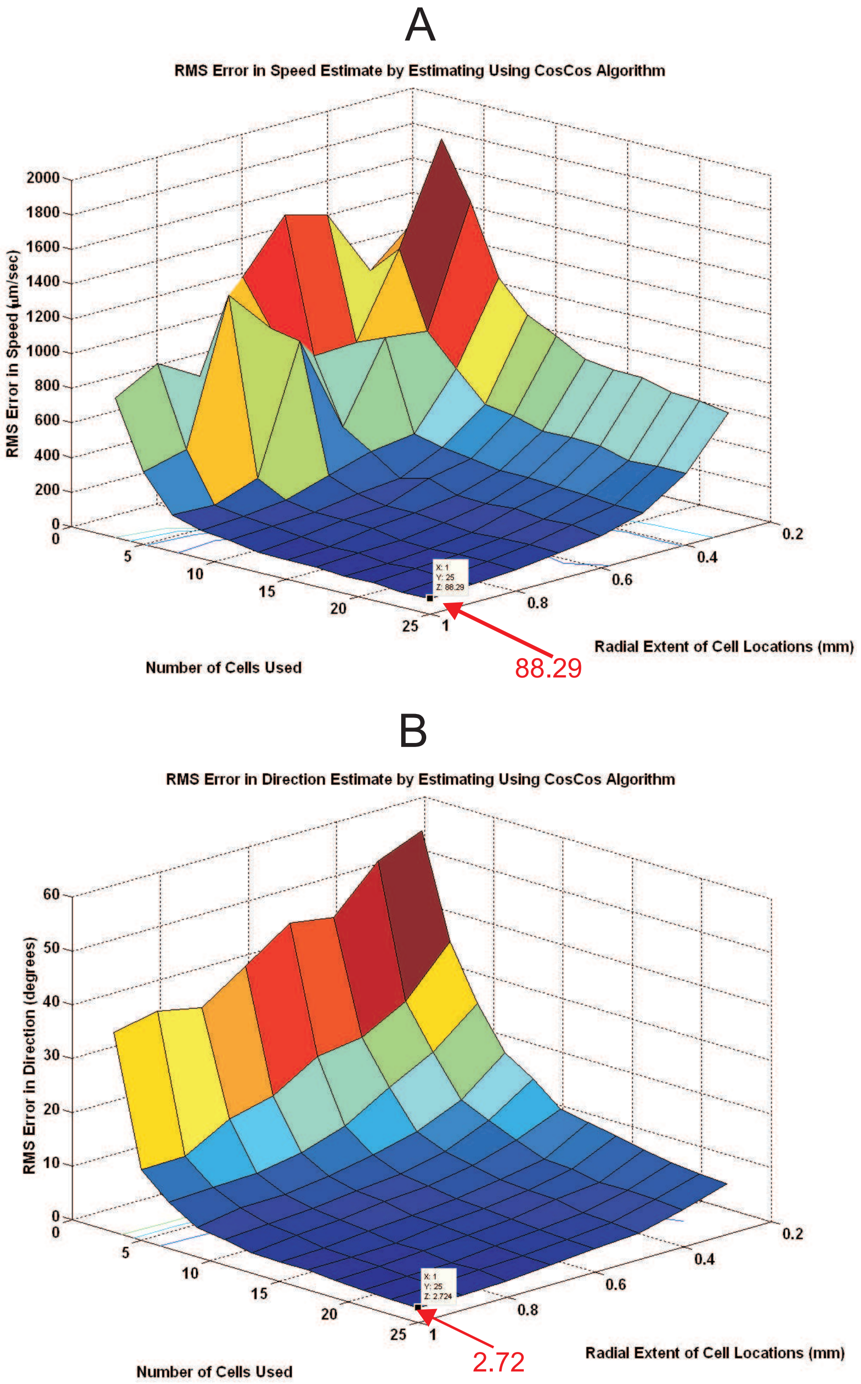

Figure 5-11: **A** and **B** show the errors in estimating speed and direction, respectively, by using the CosCos algorithm. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

Once again, it is apparent that as N and R increase, the RMS errors in the estimates of speed and direction decrease. In addition, we see that the estimates of speed and direction get worse as the speed is doubled.

5.4.5 Estimating Speed and Direction by the Newton-Raphson Algorithm

In this subsection, we show the results of estimating speed and direction by the Newton-Raphson algorithm, presented in the *Theoretical Developments* chapter. The results, shown in Figures 5-12 and 5-13, are for motion of a curtain at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$.

It is apparent that as N and R increase, the RMS errors in the estimates of speed and direction decrease. In contrast to the CosCos algorithm, it appears that the estimates of direction are more sensitive to the cells being contained in a very small radius. Also for this algorithm, we see that the estimates of speed and direction get worse as the speed is doubled.

As was derived in the *Theoretical Developments* chapter, if N pairs of cells were evenly spaced on a circumference of radius R, the variance of the speed and direction estimates, estimated by using the Newton-Raphson and the CosCos algorithms, both approximately decrease as $\frac{1}{NR^2}$. In the case that the cells are not restricted to lie on the circumference, but can lie inside the circle as well, we would like to understand if the mean-square error in the speed and direction estimates has a dependence of the type $\frac{1}{N^pR^q}$, for some $p, q > 0$, where N is the number of cells (not the number of cell pairs), R is the radial extent of the cell locations. Given N cells, we can make many more than N cell pairs. However, since we let the cells be inside the circle as well, it is reasonable to investigate if the decrease in error will be of the form $\frac{1}{N^p}$, versus having a stronger inverse dependence on N.

We investigate into this idea by making log-log-log plots of the RMS error in speed and direction when using the Newton-Raphson and CosCos algorithms, for $v = 714\mu m/sec$. That is, we plot $\log(RMS(error))$ vs. $\log(N)$ and $\log(R)$, and hope

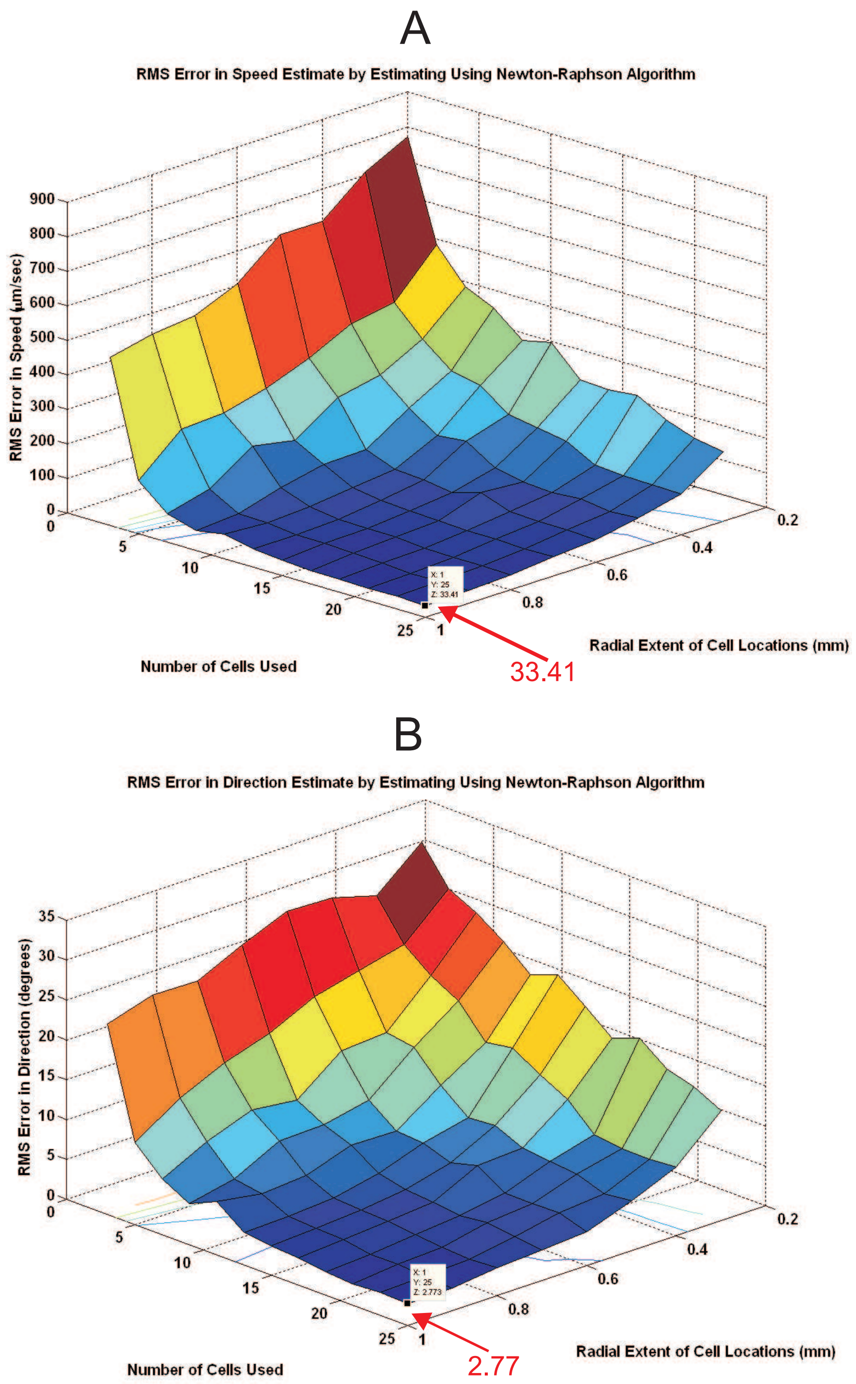

Figure 5-12: **A** and **B** show the errors in estimating speed and direction, respectively, by using the Newton-Raphson algorithm. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

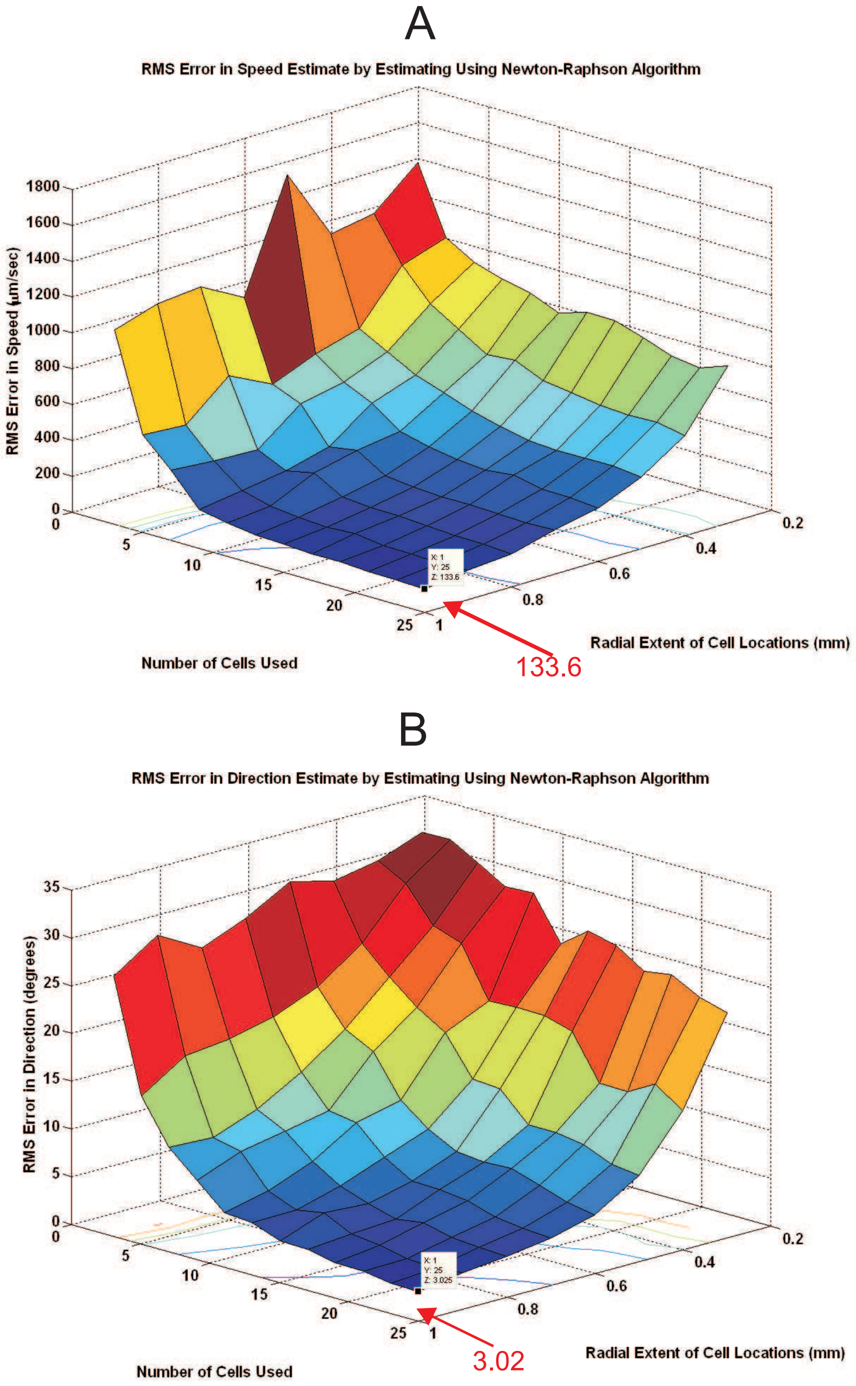

Figure 5-13: **A** and **B** show the errors in estimating speed and direction, respectively, by using the Newton-Raphson algorithm. The speed of the curtain is $v = 1428\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

that it results in a plane.

We notice that the data does appear to lie on a plane when we make these plots. To understand the dependence on N and R, we fit a plane to the data using a total least squares method[1]. Next, we calculate the slope of the plane along the dimensions of N and R to find p and q.

Figures 8-1 and 8-2, in the Appendix, show the planes which are fitted to the data. The values of p and q which we get from these plots are organized in the table below.

Table 5.1: Dependence of Mean-Square Error on N and R.

	CosCos	Newton-Raphson
MS error in speed	$p = 1.86, q = 3.69$	$p = 2.15, q = 2.98$
MS error in direction	$p = 2.09, q = 2.09$	$p = 1.73, q = 2.58$

5.4.6 Comparing Algorithm Performance

In an effort to understand which of the devised algorithms performs best under each scenario, we have prepared plots which denote the algorithm with lowest RMS error in speed and direction estimates for each combination of N and R. The algorithm "Estimating Velocity Vector Directly Without Weights" was not compared to the others due to its poor performance. Figure 5-14 depicts this comparison.

In summary, we notice that for $R = 1$mm and $N = 25$, we are able to estimate speed and direction of a moving curtain with very low error. The resulting RMS errors for the algorithms which do best for these settings of R and N are summarized in Table 5.2.

Table 5.2: Quality of Curtain Speed and Direction Estimates of Best Algorithm.

	$\mathbf{v} = \mathbf{714}\mu m/sec$	$\mathbf{v} = \mathbf{1428}\mu m/sec$
speed RMS error	$33.4\mu m/sec$ (4.6%)	$88.3\mu m/sec$ (6.18%)
direction RMS error	2.32°	2.72°

[1] The total least squares algorithm minimizes the sum of squared perpendicular distances from each of the points being fitted to the fitting plane.

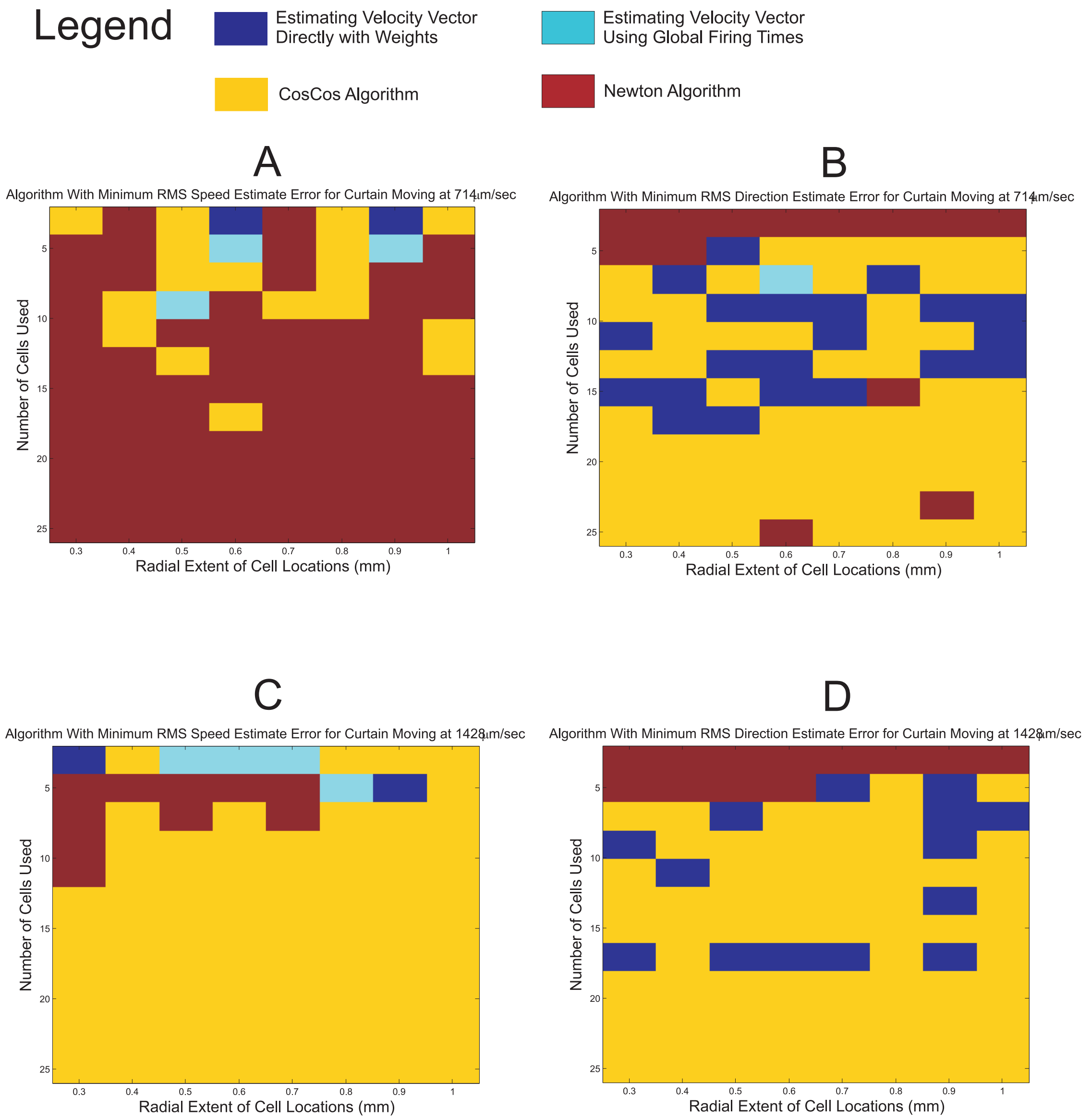

Figure 5-14: Comparing Algorithm Performance in Estimating Speed and Direction. **A** and **B** depict the algorithm which performs best in estimating speed and direction, respectively, for a curtain moving at $v = 714\mu m/sec$. **C** and **D** depict the same for a curtain moving at $v = 1428\mu m/sec$. It is seen that the CosCos algorithm dominates in the region of interest (big R and N) in all except speed estimation for $v = 714\mu m/sec$ (here Newton-Raphson dominates). This plot is based on the data from simulations in the previous subsections.

5.4.7 Comparison of Theoretical Derivations to Simulations

It has been shown in the *Theoretical Developments* chapter that the sensitivities of the speed and direction estimates are, to first order:

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

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

for both the CosCos algorithm and the Newton-Raphson algorithm, which seem to be the leading algorithms in the cases of importance to us.

In this subsection, we compare the simulated results of evenly spacing N pairs of cells on a circumference of radius R to the theoretical results above. We set $v = 714\mu m/sec$, $\sigma_p = 100\mu m$, $\sigma_{\Delta t} = \sqrt{2} \cdot 0.1sec$, and plot [$\sigma_{theory}$ - RMS(error)] of both speed and direction estimates, as a function of R and N, for the Newton-Raphson algorithm and the CosCos algorithm. The comparison is seen in Figures 5-15 and 5-16.

In these plots, it can be seen that the difference between theory and simulation is very small for speed and direction sensitivities when using both algorithms as N and R get big. However, we see that the Newton-Raphson algorithm simulation results are further from the theoretical results for very small N and R. In addition, we see that the CosCos algorithm simulation speed sensitivity results get further from the theoretical results as R gets small.

5.5 Moving Bar Simulations

In this section, we present the results of simulating the motion of a bar at a fixed speed of $v = 714\mu m/sec$. When simulating purely non-DS cells, using the Newton-Raphson algorithm, we calculate the RMS error in the estimates of the bar's speed and direction over 300 simulation runs, where the errors are defined to be $(\hat{v} - v_{true})$ and $(\hat{\theta} - \theta_{true})$, respectively. We make plots of the RMS error in the estimates as D (thickness of the

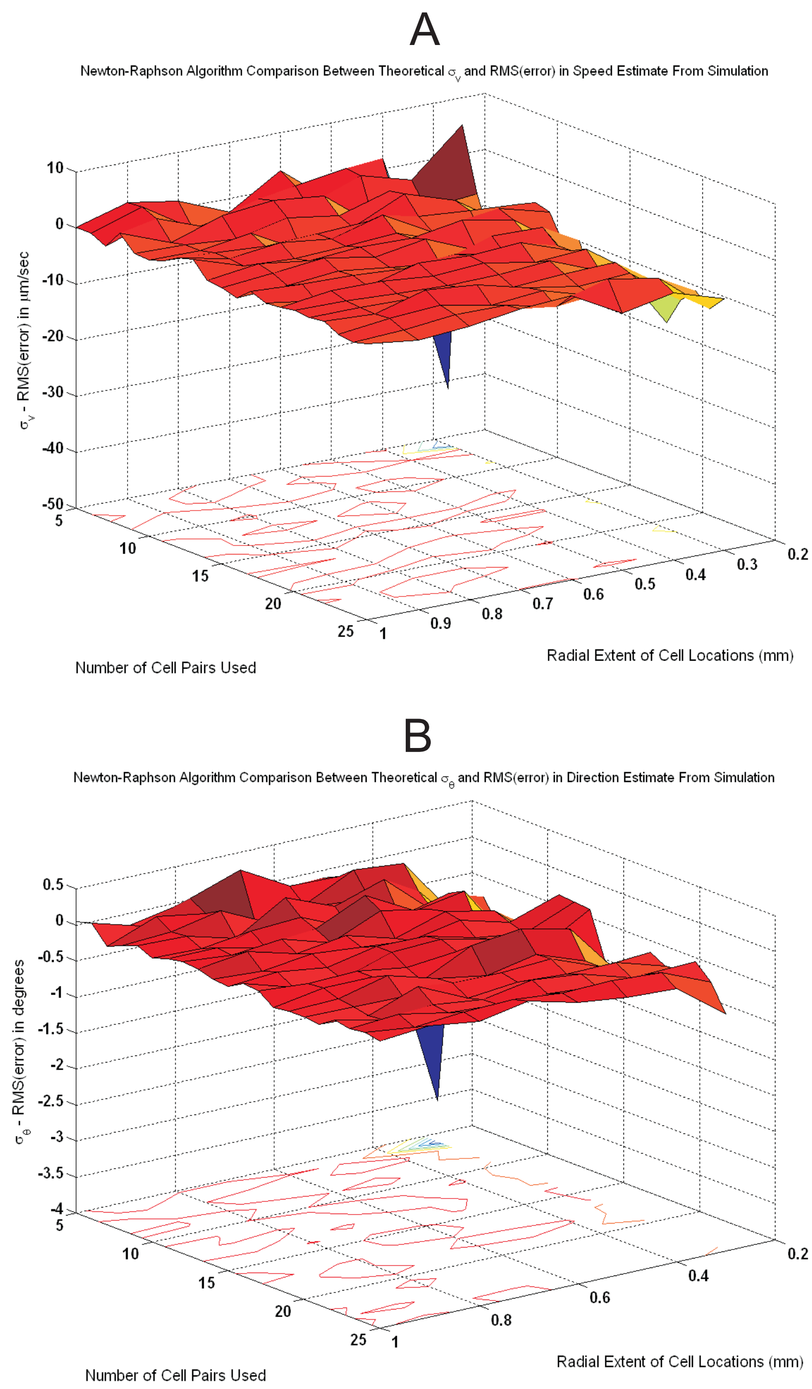

Figure 5-15: Newton-Raphson Algorithm Simulation Compared to Theoretical Results. **A** and **B**, for speed and direction respectively, depict the difference between the theoretical standard deviation of the estimate (by assuming small errors in the parameters, and making first order approximations) and the simulated RMS error in the estimate when using the Newton-Raphson algorithm. The RMS errors were averaged over 300 trials. Negative values in the difference mean that the error in simulation is higher than what we expected theoretically.

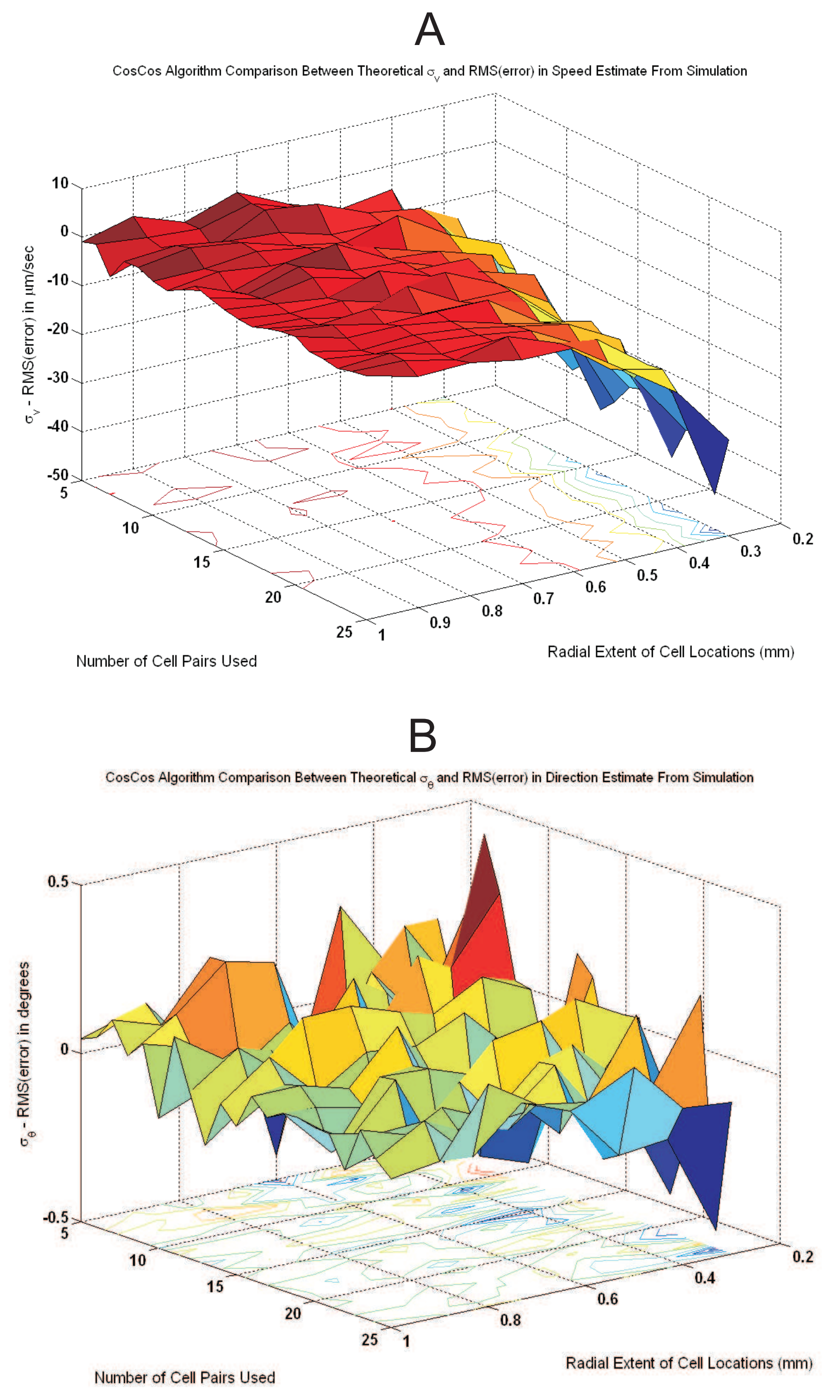

Figure 5-16: CosCos Algorithm Simulation Compared to Theoretical Results. **A** and **B**, for speed and direction respectively, depict the difference between the theoretical standard deviation of the estimate (by assuming small errors in the parameters, and making first order approximations) and the simulated RMS error in the estimate when using the CosCos algorithm. The RMS errors were averaged over 300 trials. Negative values in the difference mean that the error in simulation is higher than what we expected theoretically.

bar) and N (the number of cells used to make the estimates) are varied. The STDs of the noise introduced into the parameters are $\sigma_p = 100\mu m, \sigma_{\Delta t} = \sqrt{2} \cdot 0.1$ sec.

5.5.1 Estimating Speed and Direction Without DS Cells

It is worth noting that the reason why we estimate the bar motion parameters using only the Newton-Raphson minimization algorithm is that is naturally leads us into the next step, minimizing 4.111, which was established in the *Theoretical Developments* chapter. In addition, the estimate fidelity when using this algorithm is not too far away from the CosCos algorithm (shown in simulation and theoretically).

Bars which are 357μm wide, as are the ones we used in experiment, have an effect on cells whose RFs lie (at least partially) in a strip of approximately 500 μm in width. Therefore, we wish to focus on simulated results for bar widths of 500 μm. Regardless of N, we notice in these simulations, that the estimates of speed and direction are very poor compared to the quality of the estimates produced for curtain motion simulations. This was expected because the variety in angles formed by cell pairs is small, since they all lie in the same strip. The results of this simulation can be seen in Figure 5-17.

Although it's interesting to observe how the RMS errors in the speed and direction estimates vary as D and N grow, we are specifically interested in the results for bars of width 500 μm. For this bar width, we note that the estimate fidelity does not depend heavily on the number of cells used. When using 9 cells, a typical scenario we face with experimental data, the RMS error in speed is $355.7\mu m/sec$ (50%) and in direction 30.6°. These estimates are clearly not as good as the estimates we are able to attain for moving curtains. To be more specific, in simulation, the length of the strip which the bar traverses is 2mm. For a strip width of 0.5mm the total area over which the bar moves is then 1(mm)^2. In the case of moving curtains simulations, a radius of $\sqrt{\frac{1}{\pi}} \approx 0.56$mm gives a circular area of 1(mm)^2. In particular, we notice that in the plots of RMS error in speed and direction when using the Newton-Raphson algorithm to make the estimates of curtain motion parameters, we obtain errors of $213.1\mu m/sec$ and 11.8° in speed and direction, respectively.

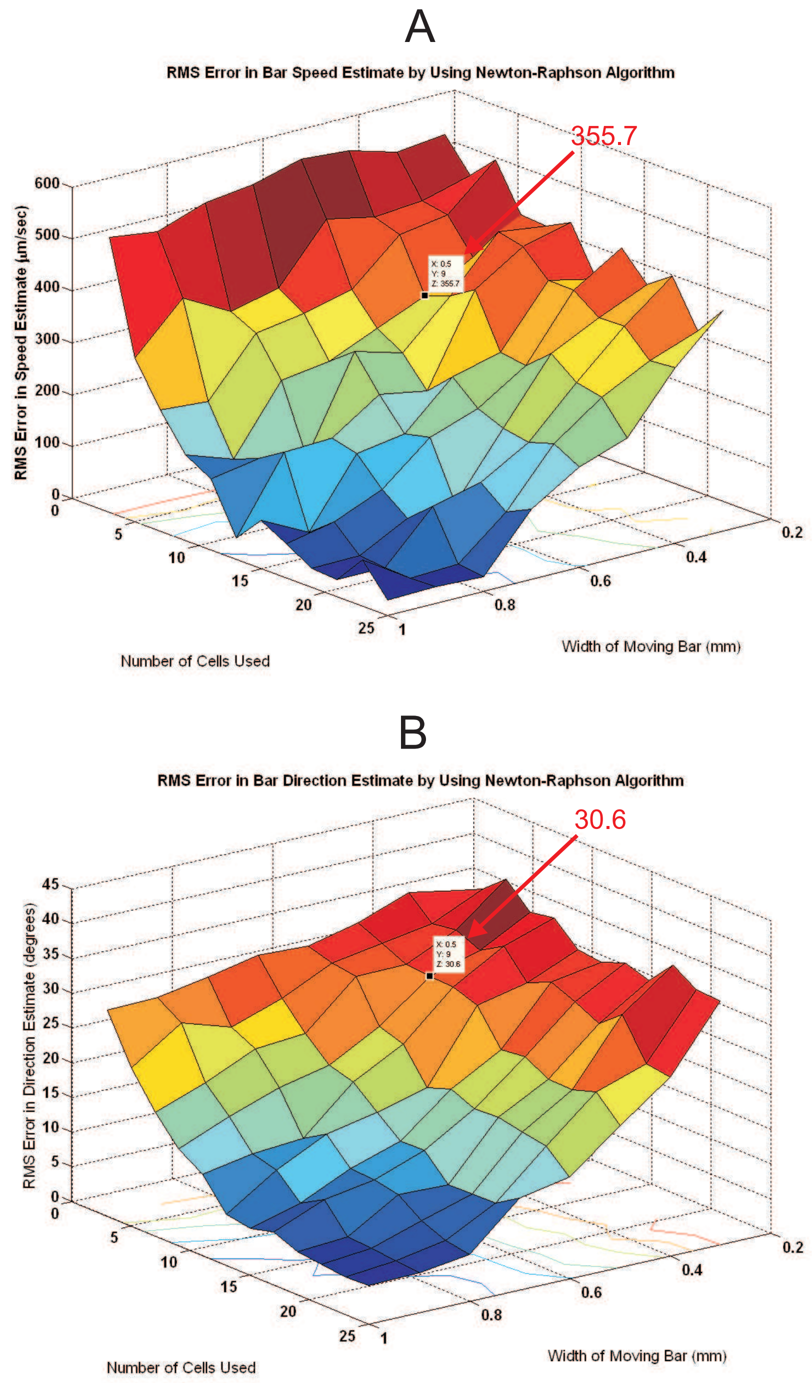

Figure 5-17: **A** and **B** show the errors in estimating speed and direction of a moving bar, respectively, by using the Newton-Raphson algorithm. The speed of the bar is $v = 714\mu m/sec$. The errors which are plotted are RMS errors averaged over 300 trials, all independent of each other (i.e., for each trial the cells are re-placed inside the circle at random and the parameter noise is re-picked, independently from all other trials). We plot the RMS error as we vary the number of cells used along with the width of the moving bar. The STDs of the noise introduced into the simulation are: $\sigma_p = 100\mu m, \sigma_{\Delta t} = 0.141$ sec, which reflect the STDs of the errors we get from real data.

5.5.2 Estimating Speed and Direction With DS Cells

Due to the poor quality of the estimates, we search for refuge in DS cell information and find it necessary to simulate DS cell responses. We do this as described in Section 5.3 by introducing additive Gaussian noise of STD equal to 30% of the real number of spikes the cell would have fired according to its elliptical firing profile. The estimates are made as described in 4.5. Giving the DS cells very high significance in the minimization ($K_g = 10^9$), a trend appears showing that as more DS cells are used to estimate the speed and direction of the bar, both estimates get better. In particular, the direction estimate RMS error decreases by almost 50% as the information from 4 and 5 DS cells is introduced. The results can be seen in Figure 5-18.

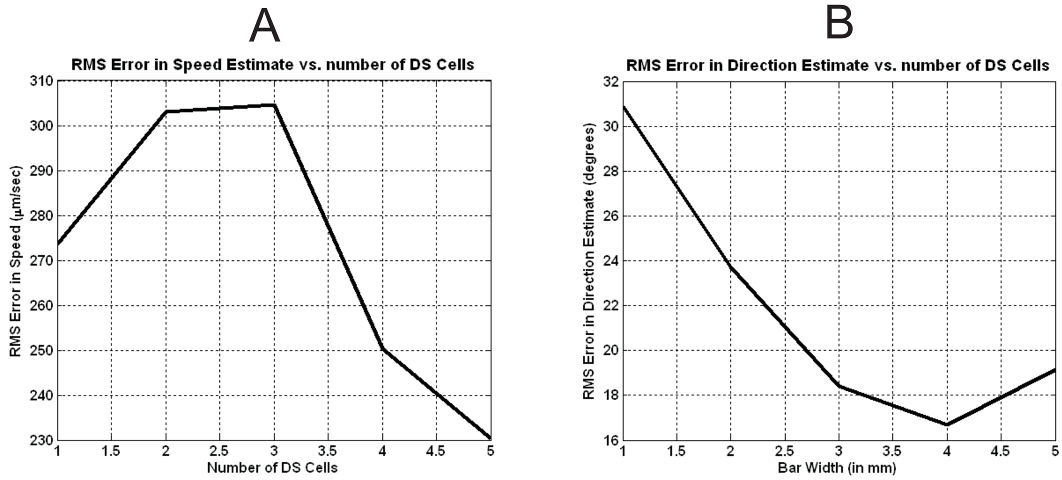

Figure 5-18: Estimating Bar Speed and Direction Using the Newton-Raphson Algorithm to Minimize $q(v, \theta)$, i.e., Using DS Cell Information. $v = 714\mu m/sec$. **A** and **B** show a decreasing trend in RMS error (averaged over 300 simulated trials) of speed and direction estimates, respectively, as the number of DS cells is increased. The decrease of the direction estimate RMS error is more apparent, decreasing by almost 50% as 4 and 5 DS cells are introduced into the minimization algorithm. For this minimization, DS cells are given very high significance, i.e., $K_g = 10^9$.

Chapter 6

Data Processing Methods

Here we present the methods which were used to transform the data from the post spike sorting state into numbers which are inputs to the estimation algorithms. In essence, this chapter is a continuation of the chapter *Experimental Procedures*. It was put further back in order for the reader to understand the reasons why the data was processed in the manner to be described. In the aforementioned chapter, we described the procedures which were followed to measure the retina's response to visual stimuli and to convert the raw measured data into a spike time ensemble (times at which each cell whose activity we measured fired). The following is a description of the data analysis required to transform the spike time ensemble, relative to the visual stimulus, into: 1) A selection of cells which are useful to estimate speed and direction of a moving curtain and bar, 2) Estimates of the coordinates of each cell's RF center location, 3) For each pair, the time between the moments at which each cell in the pair fired, 4) A classification of cells as DS, 5) Polar firing plots representative of DS cell firing strengths for multiple directions of motion.

6.1 Processing ON, OFF and ON-OFF Non-DS Cell Spike Times

In order to estimate the speed and direction of a moving curtain or moving bar, we select the cells whose response will be used in the estimation algorithm; we find estimates of each of their RF center locations; and, for each cell pair, we produce estimates of the time between the moments at which each cell in the pair fired.

6.1.1 Selecting Non-DS Cells

We select cells that have transient responses to curtain motion, as in Figure 1-3. The reason for this is that bursts of spikes provide better time resolution of the moment at which the curtain/bar passed over the receptive field. The second requirement to select a cell was that it responded robustly to motion on at least two different axes. This requirement is enforced in order to have enough data to estimate the RF center location for each selected cell. The cell selection judgement was made on a case-to-case basis.

6.1.2 Finding the Locations of Non-DS Cell RF Centers

To find a cell's RF center we study its response to motion of a *known* set of curtains moving at $v = 714\mu m/sec$ in as many directions as were run on the experimental day. For every axis of motion in which the cell responded robustly, we do the following: 1) Calculate the mean firing position (over all trials) for one of the directions of motion on the axis and throw away spikes which are 1.5 standard deviations away from the mean on the later side of the burst[1], 2) Calculate the mean firing position for motion in the opposite direction, 3) Denote the mean firing position on the axis as the mean of the positions found in 1) and 2).

By this method, each axis of motion in which the cell responds robustly suggests a

[1] The reason why we throw away spike which are at the end of the burst is to make up for the fact that some cells are somewhat less transient and fire even after the edge has left their receptive fields.

line on which the cell's RF center potentially lies[2]. For this reason, a robust response to a *single* axis of motion is not sufficient to produce an estimate of the cell's RF center. We then find cell k's RF center (x_k, y_k) to be the point which minimizes the sum of squared perpendicular distances from itself to each line (one for every axis of motion with a robust response).

Lastly, we substantiate the estimated location of each cell's RF center by comparing it to the location of the electrode which measured its signals. Because it is rare for two electrodes to sense activity from the same cell, the electrode location provides a rough approximation of where the cell is located.

6.1.3 Finding the Time Between Firing for Pairs of Non-DS Cells

We select a subset (out of the set of cells whose positions we found) of cells which respond robustly to each curtain/bar (speed and direction unknown). The selected cells are paired in all possible ways, and the time between the moments at which each cell in a pair fired is calculated as follows: 1) For every trial of motion of the particular curtain/bar, we smooth each cell's spike train by convolving it with a Gaussian kernel of standard deviation 30msec, 2) The smoothed spike trains are cross-correlated and the peak of the cross-correlation is found, 3) The firing lag between the cell pair k, Δt_k, is chosen to be the time location of this peak, 4) Each number found in 3) (one for each trial) is averaged, the result of this represents the average firing lag between the cells in the pair.

The reasons for finding the average firing lag between cells in each pair, versus finding the lag for each trial are: 1) Due to spike sorting limitations and imperfect action potential threshold settings during recording, we can not correctly identify every spike produced by every cell. For this reason, a cell's firing during certain trials is not representative of what actually occurred in response to the stimulus. Therefore, averaging over trials reduces the noise in the overall lag estimate; 2) We have access to

[2] The line passes through the point calculated in 3) and is perpendicular to the axis of motion.

more cells in the estimation process if we use the overall (over all trials) response lag, as opposed to the individual trial response lags, because for a particular trial a subset of cells will not respond robustly (to our knowledge, based on threshold settings and spike sorting).

An analogous individual trial analysis could be carried through, i.e., a firing lag estimate could be produced based on single trials, rather than averaging over trials.

6.2 Processing DS Cell Spike Times

In order to use the directional information that DS cells provide about a moving bar, we first identify the cells which have a directional selective response. Next, we find their positions in the plane and construct directional firing profiles, as in Figure 1-5. Lastly, we select a set of bars (unknown speed and direction) whose speed and direction will be estimated. For each of these bars, we quantify the response of every DS cell along its path.

6.2.1 Classifying Cells as DS

As described in the *Experimental Procedures* chapter, bars are moved back and forth along every axis of motion at different positions along the visual space, effectively covering it. ON-OFF DS cells are selected based on the response to this stimulus type. A cell is classified as ON-OFF DS if for a particular axis of motion it responds robustly to the leading and trailing edges of the bar in one direction and responds lightly or does not respond at all for motion in the opposite direction. For example, the cell in Figure 6-1 is clearly classified as ON-OFF DS.

6.2.2 Finding the Positions of DS Cells

To estimate the position of ON-OFF RF centers, we are not able to use their response to curtain motion because ON-OFF DS cells do not respond to curtain motion robustly due to RF surround inhibitory effects. In general we can not use their response

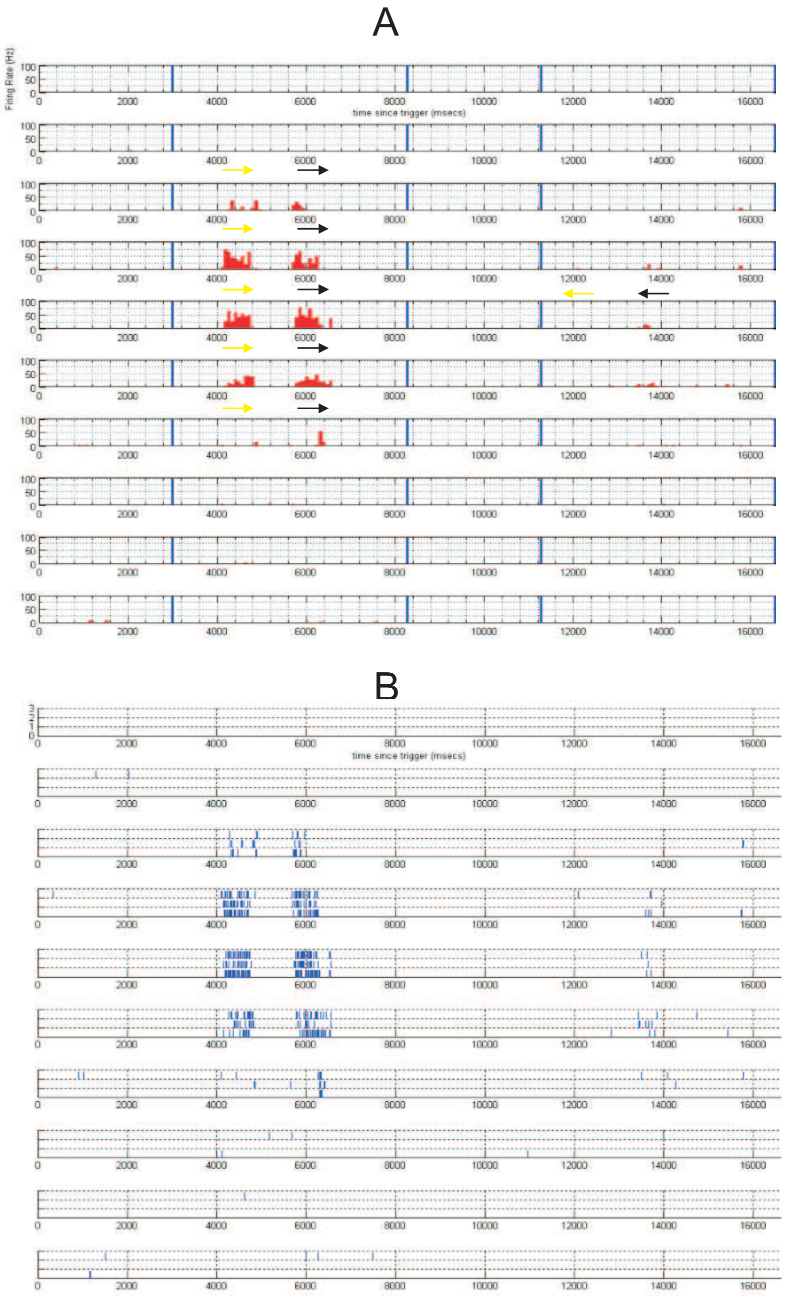

Figure 6-1: ON-OFF Directional Selective Response. **A)** PSTH (over 3 trials) of the response of an ON-OFF DS cell to a bar moving back and forth the visual space. Each plot, in the set of 10, represents the motion of a bar along a different part of the visual space (the 10 parts more than cover the visual space). For a given plot, the response contained between the first pair of blue lines is to motion close to the cell's preferred direction. The response (or lack thereof) contained between the second pair of blue lines is to motion in the opposite direction. The yellow arrows signal the ON effect caused by the leading edge, whereas the black arrows signal the OFF effect caused by the trailing edge of the bar. **B)** Corresponding 3 trial raster plot. Each trial was run within 15 minutes of the previous one to assure that the retina was under similar conditions over different trials.

to bars moving in opposite directions (as done for curtains) because by definition they have directional selectivity. To estimate the position of their centers we observe each cell's responses to the motion of the two bars which make it fire most robustly and are on different axes of motion. Next, we locate the lines in the visual plane along which each of the two bars moved[3]. The intersection point of these two lines is taken to be the cell's RF center[4].

6.2.3 Making DS Cell Polar Firing Plots

A DS cell's polar firing plot represents the robustness with which the cell fires for motion in each possible direction. We construct these plots by representing the cell's firing strength by the average number of spikes (including ON and OFF responses) that the cell fires for motion in each direction. For each particular direction, the average number of spikes is found by averaging over all trials, and over the response of the three bars which make the cell fire most[5]. We can possibly calculate these firing strengths only for motion in the directions at which we ran the bars during experimentation. Each strength is plotted as a point which is a distance — average number of spikes — from the origin in the direction of the particular bar's motion. Then, in order to produce a polar plot which represents the cell's firing strength in a continuum of all possible directions, we fit the available data with an ellipse. The expected number of spikes for the motion of a bar in a direction θ is then $h_k(\theta)$ as mentioned in section 4.5 and is defined by the shortest distance from the origin to the ellipse's trace in the direction θ. All bars which are used to make these plots move at a speed of $v = 714\mu m/sec$.

[3] These imaginary lines cut the strip defined by the motion of the bar into two equal halves.

[4] An intersection point will exist because the chosen bars move on different axes.

[5] Normally, the three bars are contiguous ones. For example, in the case of the cell whose response is in Figure 6-1, these three bars are # 4, 5, and 6 (counting from top to bottom).

6.2.4 Finding the Number of Spikes Fired by a DS Cell to a Moving Bar

For each moving bar, the number of spikes fired by a DS cell to a moving bar is found by averaging the number of spikes that the cell fired in response to the bar over all trials. This number, S_k (defined in section 4.5 to be the number of spikes fired by DS cell k in response to the motion of a particular bar whose speed and direction we are estimating), is handed to the estimation algorithm.

Chapter 7

Experimental Results

In this chapter, we present the results of estimating the speed and direction of moving curtains and bars purely from the response of RGCs of a rabbit retinal patch. Here we focus on data from a single experimental day.

Initially, we present the details regarding the types of cells we use to make the curtain estimates, the number of cells with useful transient responses, and the number of curtains (each one in a different direction) we use to calculate the average RMS error in curtain speed and direction estimates[1]. Subsequently, we present the results regarding the dependence of RMS error on the number of cell pairs used and on the maximum allowed separation between cells[2]. We compare the experimental results to what we found in simulation. Next, we present the results of estimating the speed and direction of bars moving at $v = 714\mu m/sec$ and $v = 1428\mu m/sec$. Lastly, we show how the estimates of direction in fact get better as DS cell information is used at the small price of a decay in speed estimate fidelity (contrary to simulation results).

7.1 Cell Findings

We will present results based on data acquired on the experimental day of 04/06/07. On the former experimental day we ran 5 trials of curtains moving at a single speed

[1] The average is over the number of curtain directions.

[2] The maximum allowed separation is analogous to the diameter of the circle in the *Simulations* chapter.

($714\mu m/sec$) in 8 distinct directions. In addition, we ran 5 trials of bars in 8 directions and at three distinct speeds ($714\mu m/sec$, $1428\mu m/sec$, and $2142\mu m/sec$) as described in protocol #2 of section 3.3.3, in the *Experimental Methods* chapter. However, we present the results of the estimations only for the speed of $714\mu m/sec$ for the sake of brevity and clarity.

As transient OFF responses were more abundant than transient ON responses, we chose to use the OFF responses of OFF and ON-OFF cells to estimate the speed and direction of moving curtains. On the experimental date of 04/06/07 we selected 16 cells that had transient OFF responses which appeared to be useful for our purposes. For this experiment, the variance in the cells' RF center position estimates and the variance in the Δt_k's appeared to be a bit smaller than the ones used in simulation[3]. Figure 7-1 depicts the RF center location estimates of these cells, which were estimated as described in the *Data Processing Methods* chapter.

To facilitate the task of finding the time delays between cell firing times for bar motion, we chose to use the responses of cells which had ON *and* OFF bursts in response to the leading and trailing edges of a bar. A total of 27 such cells were found. However, a single bar typically made 4-6 of these cells respond.

A total of 18 cells were identified as ON-OFF DS in response to all three speeds of motion. Each bar whose speed and direction we estimated activated 2-6 DS cells.

7.2 Estimating Speed and Direction of a Moving Curtain

In this section, we present the results of estimating the speed and direction of a moving curtain the speed of which is $714\mu m/sec$. The estimates are made using the CosCos algorithm which proved to be superior to the Newton-Raphson algorithm in most simulated cases (Newton-Raphson dominated when estimating speed for curtain motion with $v = 714\mu m/sec$). The average RMS errors (averaged over 7 curtains,

[3] These are variances from the means which were computed over all trials to make the estimates of position and Δt.

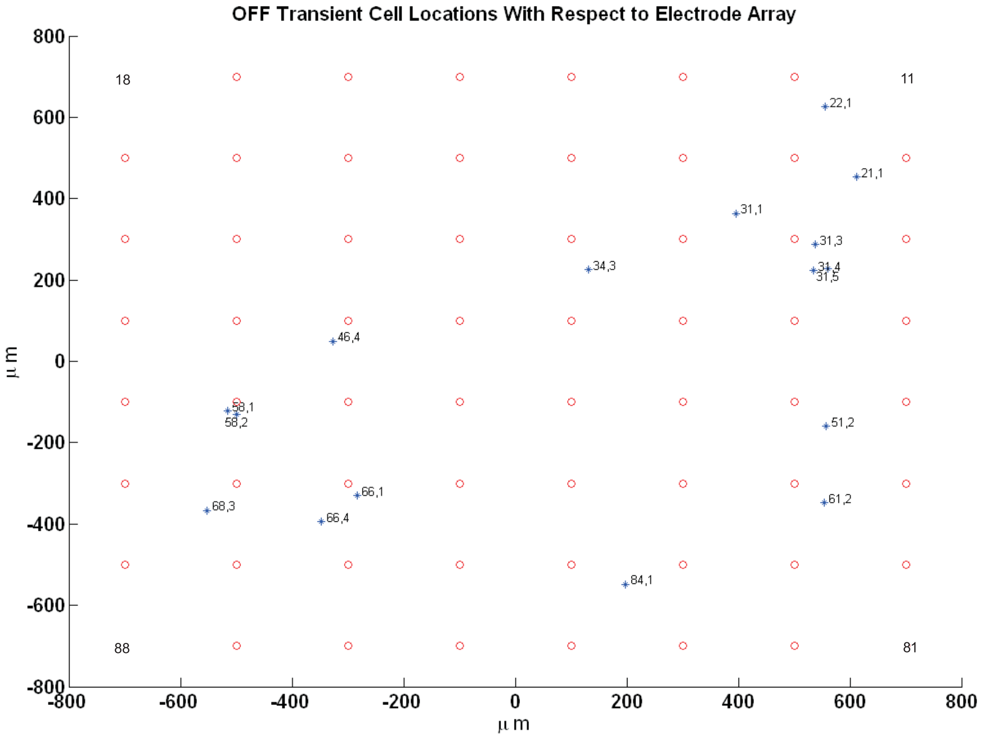

Figure 7-1: OFF Transient Cell Locations With Respect to Electrode Array. Electrode positions are seen in red and are numbered XY, where X is the row number counting from top to bottom, and Y is the column number counting from right to left. Cell RF center position estimates are denoted by blue asterisks. The electrode and unit number appear separated by a comma next to each cell's RF center location estimate.

each moving in a distinct direction at $v = 714\mu m/sec$) of the speed and direction estimates are plotted as a function of the number of cell pairs used N, and the maximum allowed distance ($2R$) between cells in each pair (analogous to the circle diameter, in simulation) is used to make the estimates. For each particular N, the RMS error is calculated over 1000 trials in which the N cell pairs are picked at random from all possible pairs available. For each particular value of $2R$, the RMS error is calculated over 1000 trials in which 15 cell pairs are picked at random from all cell pairs whose d_k satisfy the condition $d_k \leq 2R$.

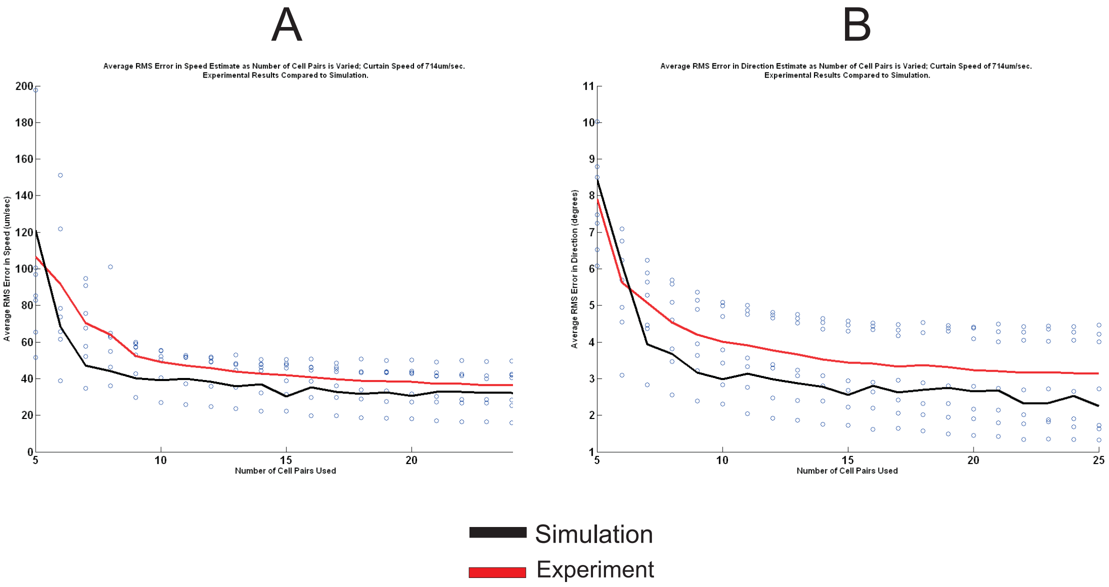

Figure 7-2: **A** and **B** depict the RMS error in speed and direction estimates, respectively, vs. number of cell pairs used. Blue dots and red line represent experimental data. Each blue dot corresponds to a single value of N and a single direction of curtain motion. Each dot represents the RMS error in the estimate, averaged over 1000 trials in which N cell pairs are picked at random from the set of all pairs. The red line represents the average RMS error (averaged over curtains in 7 distinct directions) in the estimate as N is varied. The black line represents the RMS error from 300 simulations. In each simulation, cells are redistributed at random inside a circle of radius 1mm (roughly the size of the 1.4mm × 1.4mm square electrode array), N cell pairs are chosen at random, new noise values are assigned to cell positions and firing times, and one least-squares determination of speed and angle is made. The STDs of the noise used in simulation are $\sigma_p = 50\mu m$ and $\sigma_{\Delta t} = 0.071 sec$.

It is apparent in Figure 7-2 that the experimental results follow the simulated ones very well. Figure 7-3 shows a dissimilarity between experiment and simulation

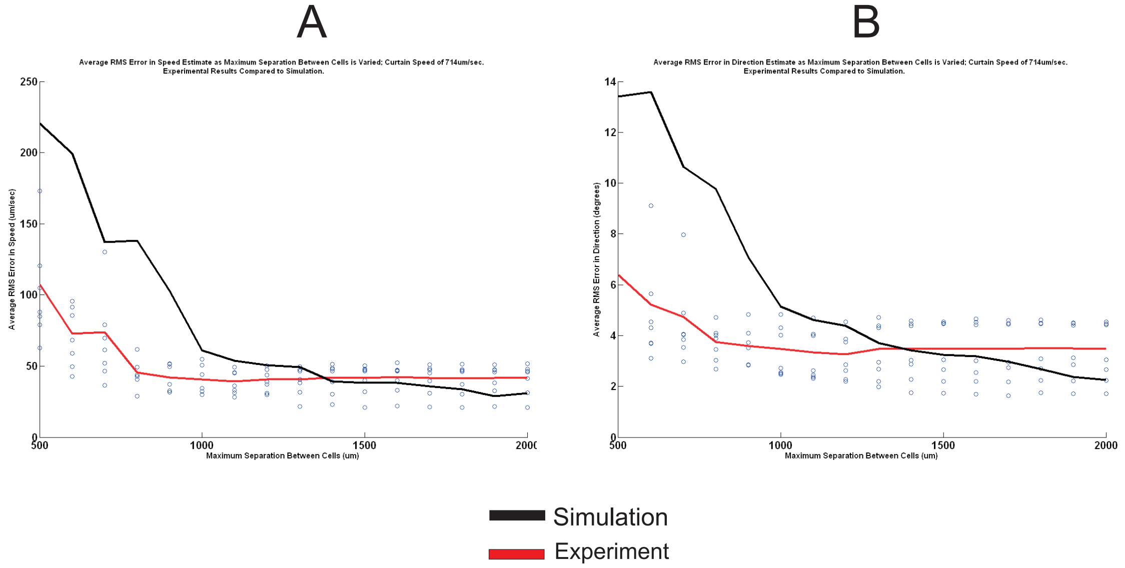

Figure 7-3: **A** and **B** depict the RMS error in speed and direction estimates, respectively, vs. maximum distance separating cells used. Blue dots and red line represent experimental data. Each blue dot corresponds to a single value of N and a single direction of curtain motion. Each dot represents the RMS error in the estimate, averaged over 1000 trials in which 15 cell pairs are picked at random from the set of all pairs whose d_k satisfy $d_k \leq 2R^5$. The red line represents the average RMS error (averaged over curtains in 7 distinct directions) in the estimate as $2R$ is varied. The black line represents the RMS error from 300 simulations. In each simulation, cells are redistributed at random inside a circle of radius R, 15 cell pairs are chosen at random, new noise values are assigned to cell positions and firing times, and one least-squares determination of speed and angle is made. The STDs of the noise used in simulation are $\sigma_p = 50\mu m$ and $\sigma_{\Delta t} = 0.071sec$.

for small values of R, but shows near convergence for larger values of R. However, the results suggest that direction estimates are good even for small values of R. Most importantly, we notice that the experimental estimate fidelity is good ($\sim$7% error in speed, and 4° in direction). Therefore, we establish that for this experimental day we were able to estimate speed and direction of a moving curtain fairly precisely.

7.3 Estimating Speed and Direction of a Moving Bar

In this section, we present the results of estimating the speed and direction of moving bars which move at speeds of $714\mu m/sec$ and $1428\mu m/sec$. The estimates are made using the Newton-Raphson algorithm because this algorithm is also used to make the estimates using DS cell information. In the first subsection we show the results of making the estimates for each of the speeds above using only non-DS ON-OFF cells. In the second subsection we show that when ON-OFF DS cells are given very high significance (i.e., $K_g = 10^9$) the RMS error in the direction estimate decreases as more DS cells are used. However, the RMS error in the speed estimate increases a bit, contrary to simulation.

7.3.1 Estimation Using ON-OFF Non-DS Cells

The average RMS errors in speed and direction estimates were found by averaging over the RMS error of estimating the motion of 11 bars (some moving in the same direction on different parts of the visual plane, in a total of 4 directions) using the non-directional ON-OFF responses of RGCs. The bars the speed and direction of which were estimated were chosen based on the condition that ON-OFF DS cells (the locations of which had been previously found) being along their trajectory. Table 7.1 below summarizes the results of the estimations. It is clear that the estimates' fidelities are far worse than the estimates achieved for moving curtains.

Table 7.1: Average RMS Error in Estimates of Speed and Direction of Moving Bars

	v =714$\mu m/sec$	**v =1428**$\mu m/sec$
Av. Speed Estimate RMS Error	$339\mu m/sec$ (47.5%)	$777\mu m/sec$ (54.5%)
Av. Direction Estimate RMS Error	73.1°	62.3°

7.3.2 Estimation Using ON-OFF Non-DS Cells and DS Cells

The average RMS errors in speed and direction estimates were found by averaging over the RMS error of estimating the motion of the same 11 bars mentioned in the previous subsection, for $v = 714\mu m/sec$. We make the estimates using all available non-directional ON-OFF responses of RGCs (including the non-directional timing information which is extracted from DS cell firing) and the directional ON-OFF responses of ON-OFF DS cells. Given a particular number of DS cells to be used, we choose this number of DS cells which lie along the trajectory of the bar at random, and use the information they provide as described in *Data Processing Methods*.

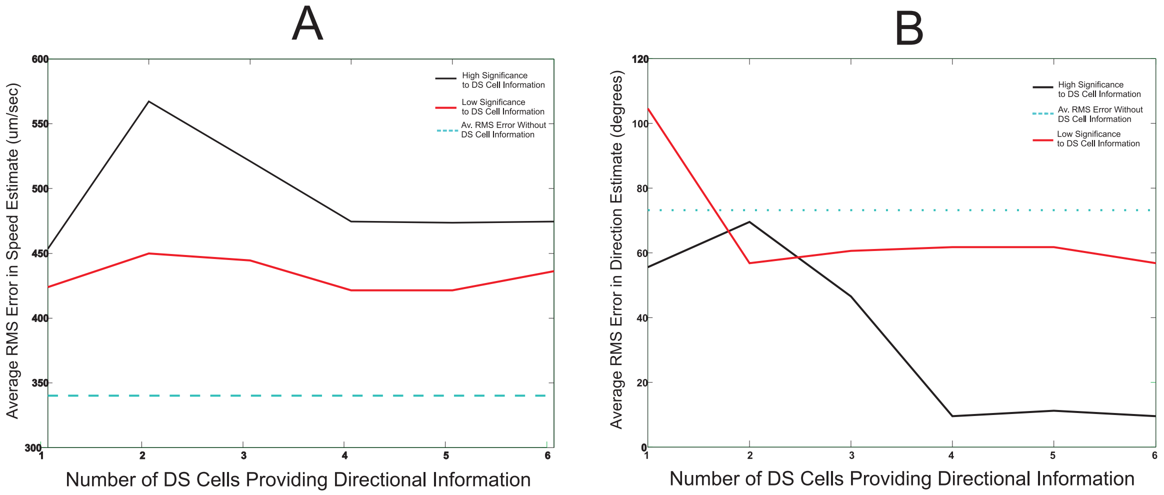

Figure 7-4: **A** and **B** depict the RMS error in speed and direction estimates, respectively, vs. number of DS cells used in conjunction with ON-OFF non-directional information (provided by both non-DS and DS cells). Each plot has a reference level (turquoise dotted line) which represents the average RMS error in the estimates found without using directional information from DS cells, i.e., the values in Table 7.1. The red lines correspond to setting $K_g = 10^6$, and the black line corresponds to setting $K_g = 10^9$.

It is apparent in Figure 7-4 that the direction estimate RMS error becomes very small ($\sim 9°$, compared to 73.1° when no DS cells are used) as DS cell information is granted high significance. However, in the case of high DS cell information significance, we notice an increase in speed estimate RMS error from $339 \mu m/sec$ to $470 \mu m/sec$. In the Future Work portion of the *Conclusions and Future Work* chapter, we suggest a method of extracting speed information from DS cell responses based on preliminary results suggested by a 1-dimensional motion ON-OFF DS cell model we constructed. In the case that we extracted speed information by the methods to be described, we believe that the RMS error in the speed estimate would decrease to below the RMS error value attained without using DS information at all.

Chapter 8

Conclusions and Further Work

In this thesis, we have proposed an instance of a simple Inverse Problem: Estimating the speed and direction of curtains and bars moving at fixed speeds and directions on the photoreceptor layer of a rabbit retinal patch. In order to delve into this problem, using an MEA, we recorded the action potentials of dozens of retinal ganglion cells extracellularly in response to these moving edges of light. After assigning each action potential to a specific unit, we were equipped with a spike time ensemble in response to each stimulus we presented on the particular experimental day.

The solution we propose is based on the intuitive picture that the edge motion information is likely contained in the relative response times of a subset of the cells in the ensemble. First, we estimate the location of every cell which we use to make the estimates of speed and direction. To estimate the speed and direction of a moving curtain, we focus on the response of cells which have transient responses. Because these responses are concentrated in time, they provide us with good time resolution of the moment at which the edge passed over the cell's RF center. To estimate the speed of a moving bar we focus on the relative timing between the responses of ON-OFF cells.

We conclude the chapter with a list of problems which could be addressed as a continuation of the work in this thesis.

8.1 Estimating Speed and Direction in Simulation and Experiment

In order to generate the estimates of speed and direction, we propose five algorithms each of which estimate speed and direction simultaneously. The first algorithm we present produces the overall estimate by first generating many estimates from each available two-pairing of cells, and then averaging. This algorithm is improved to form a second algorithm; this one computes a weighted average based on the size of the variance of each single estimate. The third algorithm, suggested by Prof. Berthold Horn, computes the estimates based on the cell positions and concentrates on the firing times as a whole. The fourth algorithm, proposed by the author and Stavros Valavanis, makes use of pairwise relative firing times and positions and solves a linear least squares problem to produce the estimates. The fifth algorithm, suggested by Prof. Wyatt minimizes a sum of squared residuals which are also based on pairwise information.

In the *Theoretical Developments* chapter, we perform a noise sensitivity analysis (in regards to a moving curtain scenario) which provides us with a grasp of how the variances of the estimate errors vary as a function of the variances in position and firing time estimates, the number of cell pairs used, and the distance separating the cells in each pair. This analysis, assumes that the parameter errors have a finite variance without regard to the parameter error distribution form. The sensitivity analysis was performed for the third, fourth and fifth algorithms. We do not compare the sensitivity results of the third algorithm to those of the fourth (CosCos) and fifth (Newton-Raphson). However, we notice after simple manipulations, that the CosCos and Newton-Raphson algorithms have the same sensitivity for small errors in the measured parameters, and when the cell pairs are evenly placed on a circumference.

Despite the fact that we don't compare each algorithms' noise sensitivities to each other analytically, we run simulations of the response of a cell ensemble to curtain and bar motion. With the help of these simulations, we describe the fidelity of the estimates produced by each algorithm as a function of the number of cells used, and the size of the circle (in the case of moving curtains) *or* the width of the strip containing them (in the case of moving bars). When we compare the performance of each of the algorithms in estimating the speed and direction of moving curtains (at speeds $v = 714\mu m/sec$ and $v = 1428\mu m/sec$, the speeds which were used in experiments), we conclude that the CosCos algorithm dominates in most cases, followed close behind by the Newton-Raphson algorithm. We conclude that in simulation, we are able to estimate the speed and direction of a moving curtain fairly accurately ($\sim$ 5% RMS error in speed estimates, and 3° RMS error in direction estimates) when introducing noise levels which are very close to those seen in experimental settings.

We also simulated cell responses to moving bars, and noticed that given the same noise levels mentioned above, we can not estimate speed and direction as accurately, which we intuitively expected (since cells lie within a strip). However, when the response of DS cells to moving bars were modeled and simulated, we showed that the information they provide reduced the direction estimate RMS error by a large amount, and somewhat reduced the speed estimate RMS error, even subject to large noise (the standard deviation of the zero-mean Gaussian noise added is 30% of the real value defined by the elliptical firing profile).

Next, we presented the results of making estimates of speed and direction based on data acquired from rabbit retinal ganglion cells. Regarding moving curtains, these results suggest that indeed, on average, we are able to produce estimates which follow the predictions of the simulations fairly closely. That is, the RMS error in the speed and direction estimates decrease when using cells which are further apart from each other and when using more cell pairs. Thus, we are satisfied with the estimates we attain with the cells which are available. Regarding moving bars, the estimates produced without using DS cell information are somewhat less accurate than what we predicted based on simulations. However, heavily weighing DS cell information increases the fidelity of the bar direction estimates (RMS error of $\sim 9°$), whereas the speed estimates become somewhat worse than they were when omitting DS cell information. This is contrary to simulation, in which the speed estimate gets better when DS information is incorporated.

In simulation, when we set $K_g = 10^9$ we are effectively minimizing $g(\theta)$ first, and then fixing the optimal value of θ in the optimization of $q(v, \theta)$, which becomes a problem of minimizing $f(v, \theta)$, where θ is fixed. Since in simulation the speed estimates get better as we use DS cell information, we can state that fixing θ to a value close to the true one helps improve the estimate of v when minimizing $f(v, \theta)$. However, we don't see this effect when performing the estimation on real data.

Once again, we *do not* make claims that the brain performs estimation of speed and direction in the ways we propose. The purpose of this thesis is to demonstrate that we are able to estimate speed and direction very well for curtains moving at speeds within a reasonable range. In addition, we show that using the same methods which grant us estimates of high fidelity for curtains, the fidelity decreases when estimating the speed and direction of moving bars. This opens up a great role for DS cells which seem to be meant to at least provide directional information to the brain. In this thesis we show that we are able to extract directional information effectively.

The following section provides a brief outline of what could be done to strengthen the evidence presented in this thesis even more and to further investigate about what information is available in the cell spike times.

8.2 Suggested Further Work

The following topics (in no particular order) are left to explore and could provide significant insight into the questions posed in this thesis:

1. Investigate if the estimate fidelity changes when using individual trials (as opposed to averaging over trials) to estimate the Δt's and calculate the S_k's which are passed as inputs to the estimating algorithms.

2. Devise ways to use the cells' spike times in a more sophisticated manner than simply cross-correlating the responses of a pair of cells.

3. Quantify the time lag inherent in each cell's burst in response to the motion of a curtain or bar. This time lag could be used to reduce the noise in the Δt's.

A possible way of doing this is based on the response of the cell to curtains moving at different speeds.

4. Evidence has been presented which leads us to believe that DS cells which have similar preferred directions fire synchronously with each other. It would be interesting to verify that this is true and to find a way to incorporate this information into refining the direction estimates.

5. Although the work is not included in this thesis, we have quantitatively modeled the firing characteristics of ON-OFF DS cells in response to 1-dimensional motion of binary images on the cell's preferred-null axis. The model has been compared to the results of experiments published in [2] and [11] to learn that the model agrees closely with real DS cell data. Simulations of this model suggest that the response of a DS cell to a single edge of a moving bar (of a width similar to the ones we used in our experiments) has the following property: The slope of the line segment which connects the beginning of the cell's smoothed (e.g. with a Gaussian filter) response to the peak of it, is proportional to the speed of the bar. This suggests that if the constant of proportionality is established during a training phase, the response of a DS cell could provide information about the speed of the moving bar. The information could be incorporated in a way similar to how the g_k residuals were constructed to aid in estimating direction using DS cells.

6. Simulate the responses of DS cells by including the fact that ON-OFF DS cell preferred directions lie roughly in one of 4 possible directions. Changing this in the simulation environment could make our simulated results more close to what is seen experimentally. Does making these changes cause the direction estimates for motion in some directions to be better than others?

7. Once one is able to accurately estimate speed and direction of moving curtains and bars, the next natural step is to estimate these parameters for two objects moving simultaneously in the image plane. Another interesting problem is to

track the motion of an object which changes its direction of motion at random but maintains a constant speed.

8. Compare the sensitivities of the v and θ estimates when using the algorithm suggested by Prof. Horn with the sensitivities of the Newton-Raphson and CosCos algorithms. Note that the sensitivities of the latter two algorithms are equal, to first order.

9. To improve the estimates of moving bar direction it seems plausible to notice which cells in the ensemble fired and which did not when the bar was moved.

10. In Figures 7-2 and 7-3 we notice that the blue circles follow a trajectory as N and R are increased. This is most likely because in experiment the estimates for curtains moving in some directions are better than for curtains moving in others. It might be worthwhile to plot the estimate errors for each curtain direction separately. It would be interesting to show that these discrepancies are due to the particular positions of the cells which we use to make the estimates.

11. Explore more carefully why the speed error (and not the direction error) is bigger using real data than in simulations.

12. Compare the distribution of cell pair distances in simulation and in real data. This could be the reason for the discrepancy mentioned in 11. Another option is to perform the simulations using actual cell locations from experiment.

13. Find an intuitive way to understand the sensitivities of v and θ given in Equations 4.37 and 4.41.

14. Minimize the sum of squared errors which the Newton-Raphson minimization solves by using Lagrange multipliers to perform a constrained minimization.

15. There are big velocity errors with bars. Should we weigh the data in such a way that two quite distant cells stimulated by a bar get a high weight? The best method is not clear, e.g. should we give more significance to pairs with high $\frac{d_k}{\Delta t_k}$?

16. Determine if we obtain better θ estimates with the CosCos algorithm if we estimate θ by finding the least squares solution of $\sin(\theta) = \frac{\beta}{\sqrt{\alpha^2+\beta^2}}$ and $\cos(\theta) = \frac{\alpha}{\sqrt{\alpha^2+\beta^2}}$.

17. Determine for what purposes we should estimate (u, w) versus v and θ.

18. Does Berthold's method outperform others when it is handed the exact value of T, so that it only solves for two unknowns, as the others do?

19. Extend the variance of v and θ to a disc consisting of concentric circumferences, at a spacing in R, and with a number of cells at each R that approximate a uniform distribution. In this case, should we form pairs of cells from different circumferences?

Appendix

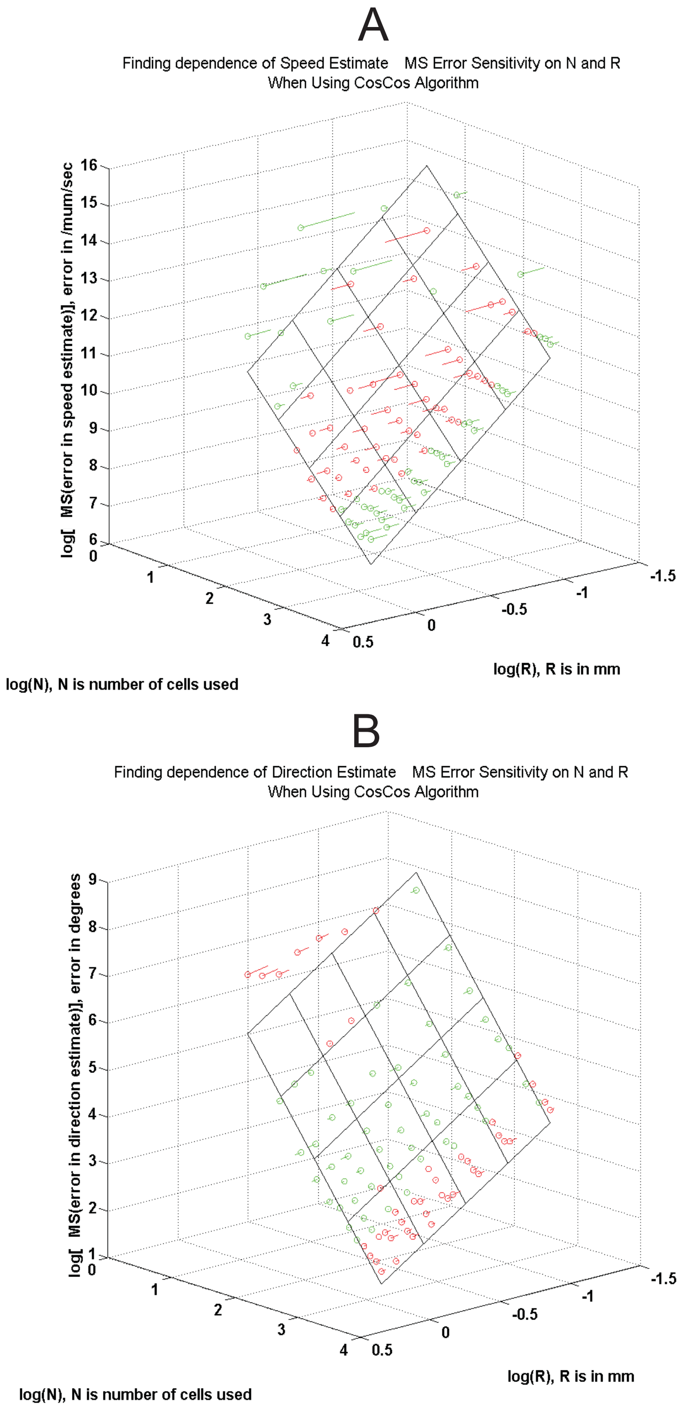

Figure 8-1: Log-Log-Log Plot of MS Error in Speed and Direction When Using the CosCos Algorithm for $v = 714\mu m/sec$. We see in **A** and **B** that log(MS error) in speed and direction, respectively, approximately lies on a plane when plotted vs. $\log(N)$ and $\log(R)$. This implies that the Mean-square error in speed and direction estimates decrease approximately as $\frac{1}{N^p R^q}$ for some $p, q > 0$. The p and q which were found appear in Table 5.1 in the *Simulations* chapter.

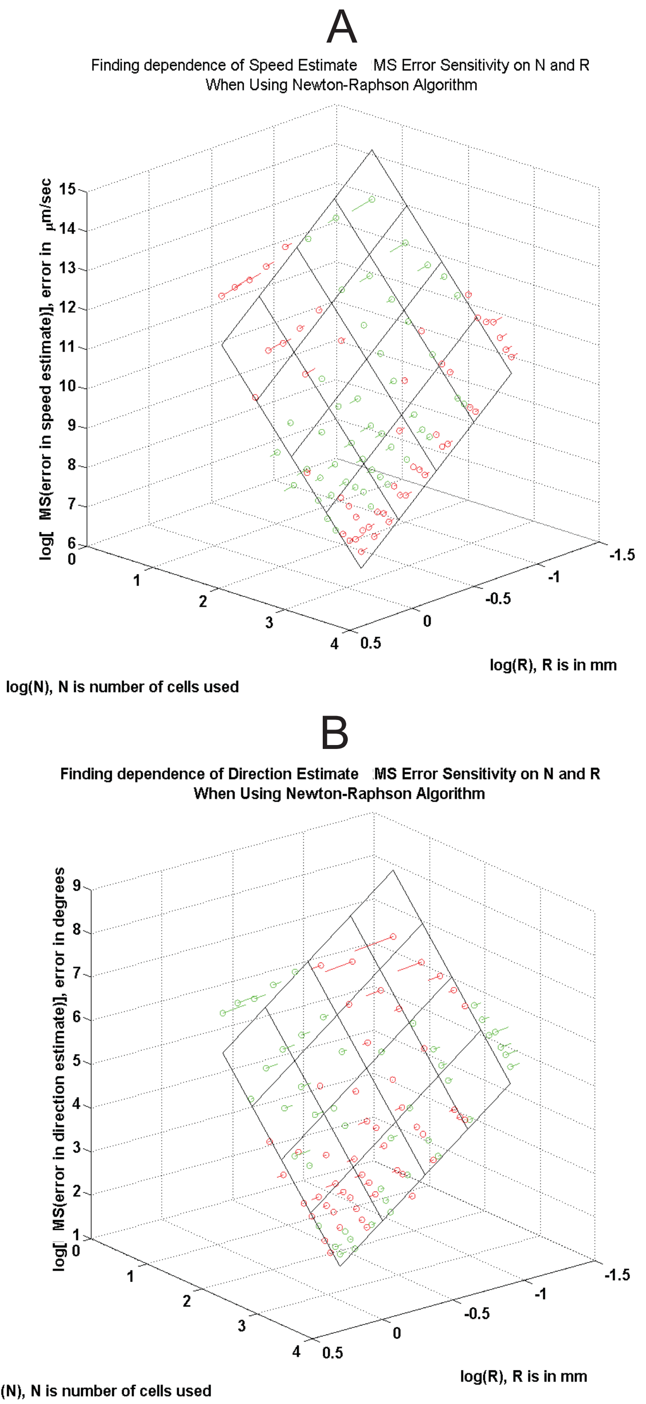

Figure 8-2: Log-Log-Log Plot of MS Error in Speed and Direction When Using the Newton-Raphson Algorithm for $v = 714\mu m/sec$. We see in **A** and **B** that log(MS error) in speed and direction, respectively, approximately lies on a plane when plotted vs. $\log(N)$ and $\log(R)$. This implies that the Mean-square error in speed and direction estimates decrease approximately as $\frac{1}{N^p R^q}$ for some $p, q > 0$. The p and q we found appear in Table 5.1 in the *Simulations* chapter.

Bibliography

[1] 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.

[2] Shelley I. Fried, Thomas A. Munch, and Frank S. Werblin. Directional selectivity is formed at multiple levels by laterally offset inhibition in the rabbit retina. *Neuron*, 46:117–127, 2005.

[3] 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.

[4] A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[5] J.Y. Lettvin, W.S. Maturana, W.S. McCulloch, and W.H. Pitts. What the frog's eye tells the frog's brain. *Proc. IRE*, 47:1940–1951, 1959.

[6] M.S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4):53–78, 1998.

[7] 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, 772.

[8] F. Rieke, R.D.R.V. Steveninck, D. Warland, and W. Bialek. *Spikes:: Exploring the Neural Code.* Bradford Book, 1997.

[9] 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.

[10] Steven F. Stasheff. Emergence of sustained spontaneous hyperactivity and temporary preservation of off responses in ganglion cells of the retinal degeneration (rd1) mouse. *Submitted for Publication.*

[11] Steven F. Stasheff and Richard H. Masland. Functional inhibition in direction-selective retinal ganglion cells: Spatiotemporal extent and intralaminar interactions. *J. Neurophysiology*, 88:1026–1039, 2002.

[12] David K. Warland, Pamela Reinagel, and Markus Meister. Decoding visual information from a population of retinal ganglion cells. *J. Neurophysiology*, 78:2336–2350, 1997.

[13] HJ Wyatt and NW Daw. Directionally sensitive ganglion cells in the rabbit retina: specificity for stimulus direction, size, and speed. *Journal of Neurophysiology*, 38(3):613–626, 1975.