# Stereoscopic Vergence Control and Horizontal Tracking Using Biologically Inspired Filters

by

Michael Anthony Schwager



A Thesis Submitted to the Faculty of the

Electrical and Computer Engineering Department

In Partial Fulfillment of the Requirements
For the Degree of

Master of Science

In the Graduate College

The University of Arizona

2 0 0 0

## Statement by Author

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

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

Signed: _____

Michael Anthony Schwager

## Approval by Thesis Director

This thesis has been approved on the date shown below:

_____ _____

Charles M. Higgins
Assistant Professor of Electrical and Computer Engineering

Date

# Acknowledgements

First and foremost I would like to thank my parents John and Maria-Helena and my brother Eric for providing invaluable support, love, and kindness throughout this exercise. Next I would like to thank my advisor Chuck Higgins for his encouragement, resources, opinions, and experience. Thanks must also go to my labmates: Sudhir Korrapati for his enlightening wit and mathematical sophistry, Shaikh Arif Shams for his friendship and company, and late-comers Sam Hill and Steve Beatty for keeping the humor running. I must thank my defense committee, François Cellier and Jeff Rodriguez for their patience and expertise in reading through this work, and of course for their approval. Many thanks also go to the administrative staff of the ECE department, Tami Whelan and Laura Laine, for without them I would never have surpassed the mountain of paperwork required to leave this place. Finally, thanks goes to everyone else who has in some way supported, encouraged, inspired, or entertained me, including Justin, Tina, Dan, JR, Jamie, Carrie, Jennifer, Julie, Jürgen, Leigh, Pablo, Bill, and Slash and Sputnik (my cats), and to anyone else I am forgetting.

Thanks also to the unknown people who helped make this job easier: Tag, for providing endless streaming aural wellness; the authors of wxWindows, the GUI and application framework at the base of this project; the authors of LyX, the graphical front-end to LaTeX, without which this thesis would have been much more difficult to write; and to Microsoft, for making me appreciate Linux more than I could have imagined.

# Table of Contents

TABLE OF CONTENTS—*Continued*

# List of Figures

# List of Tables

# Abstract

One of the requirements of enabling a robot to see in 3D is to move its gaze to match the target. Vergence is the disconjugate horizontal rotation of the cameras to move their gaze over the target. Tracking is the conjugate rotation. The difference in the two images captured by stereoscopic cameras (disparity), is a sufficient measure to accomplish both of these tasks. We reviewed studies of how cat visual cortex measures disparity, combined this disparity-energy model with neurophysiological models of vergence control, and developed a system which also controls horizontal tracking. Experiments confirm the operation of the system with software and inexpensive custom hardware. An architecture is presented for the implementation of this project in analog VLSI hardware, and will show a high degree of parallelism, low power consumption, real-time operation, flexibility and scalability. We discuss how to compare this vision system with others.

**Chapter 1**

# INTRODUCTION

This thesis is concerned with artificial stereoscopic 3D vision and its application to mobile robotics. An analysis of biological vision systems is made and the knowledge obtained therefrom adapted for use in a system which controls the position of two cameras to center a target in both fields of view.

## 1.1 Motivation

When one considers the many senses available to humans and other primates, it is clear that vision is the most useful and thus most important for navigation and identification of objects (food, enemies, mates, obstacles, etc.) in a complex environment. It has been argued that the ability to see and comprehend the environment in its full three dimensions, combined with the dexterity implicit in tree-dwelling species, has led to the current cognitive ability of humans and other primates (Pinker, 1997). It has also been shown that in primates more of the brain is devoted to vision than to any other sensory system (Zigmond *et al.*, 1999, p. 821). Therefore when examining what are the most promising and potentially useful senses with which to equip a robot, vision is one of the first that comes to mind. Not surprisingly, however, it has proven to be one of the most difficult.

It hardly seems necessary to illustrate that a properly functioning vision system is highly complex. Additionally, its completely integrated and transparent nature makes it difficult for us to analyze its various working components. It is therefore appropriate to guide the reader through a preliminary analysis to exemplify the difficulty and depth of the task which evolution has so expertly solved.

The primary purpose of vision is to enable the subject (a.k.a. "agent", "person", etc.) to be aware of the objects around it in a manner suitable for physical survival, navigation, and manipulation. In general this means the vision system should reveal large domain-defining structures, such as floors, walls, mountains, and open space, as well as smaller and potentially more important features such as predators, prey, fallen logs, holes, stairs, and in the case of robots, mission-critical objects such as radioactive waste or rock samples. In addition, the vision system often provides information about the three dimensional structure of the scene, especially in predatory species. The way a biological vision system finds these things is by using various types of static and dynamic information (cues) in the scene itself: lines and edges, brightness, texture, blurriness, binocular disparity (the difference in the two images due to the slightly different vantage points of the two eyes), and motion-parallax (closer objects appear to move more than distant objects). The vision system also uses non-visual cues such as the angle and motion of the body, head, and eyes, accommodation (focus control), and convergence (angle of the eyes toward each other) as inputs to its computational (perceptual) engine (Palmer, 1999). Higher level functions such as object recognition, face identification, pattern recognition, matching, etc., are not addressed here; this study is only concerned with what is known as early vision – that part of the visual computation process that is responsible for "identity-less" processing of visual information (Kandel, 1995).

Natural species which see have evolved varying degrees of three dimensional vision. All vertebrates perceive a visual field from each of two eyes (Sciencenet.org, 2000). The degree with which these fields overlap determines the ability of the species to see three dimensionally. Predatory species with eyes side-by-side in the front of the head tend to have visual fields which overlap more than those of prey species which tend to have eyes on the sides of the head. This shows that predators generally have better three dimensional vision than prey (Sciencenet.org, 2000; Encyclopedia Britannica Online, 2000). Cetaceans (whales and dolphins) and bats, while unable to see three dimensionally either due to the positions of their eyes or the darkness of their environment, use sonar to perceive depth in their environment to find their prey. From this we may conclude that the ability to identify suitable objects for prey or examination, the ability to move in the environment quickly and with agility, the ability to manipulate the prey once caught, and the ability of the predator to avoid becoming prey to a larger and more dangerous animal (or environment), requires the ability to perceive the environment in three dimensions.

## 1.2 Seeing in the Third Dimension

Three dimensional perception is accomplished via a multitude of cues which come from the physical state of the seer (accommodation and vergence) as well as the content of the scene (e.g., disparity,

motion, etc.). Some of these cues are also used as inputs to the vergence control system: *Convergence*, or equivalently *vergence*, is the control of the eyes such that the intersection of each of their optical axes occurs at an appropriate place in front of the eyes, usually at the surface or an edge of the object under consideration. Its control is not only the result of the processing of visual cues, but also an input to the perception of the resulting image depth. *Accommodation* is the effort the eyes make to keep the image focused on the retina by changing the shape of the lens, and results directly in the sharpness of the image. This is also both an output from an early control system and an input to perception. *Binocular disparity* is the difference in the image which is created on the retina due to the different vantage point of each eye. It is the result of the image depth and of the eyes' vergence. In subjects with eyes which are displaced horizontally, this disparity is known as *horizontal disparity,* (as opposed to vertical disparity), and is the most important element for perceiving depth. Disparity alone is also known to provide sufficient information for both depth perception and vergence control (Stevenson *et al.*, 1999; Marefat *et al.*, 1997b; Qian and Zhu, 1997; Ohzawa *et al.*, 1996; Hansen and Sommer, 1996; Mallot *et al.*, 1996; Sanger, 1988). *Proximal vergence* is the set of cognitive ("high level") cues such as perspective, the appearance of parallel lines converging to a single point as their distance from the viewer increases, which give the impression of depth.

These cues – binocular disparity, accommodation, vergence, and proximal vergence – are used as inputs to the vergence system (Palmer, 1999, pp.203-209) (Zigmond *et al.*, 1999, p.1007). In primates, disparity is the primary input to the vergence control system (Mallot *et al.*, 1996; Zigmond *et al.*, 1999; Cova and Galiana, 1994) and is the only input to the control system in this project. The first three cues are highly coupled; changes in one will invoke appropriate changes in the others so that all three cues are congruent (Jiang, 1996). Since these three cues are coupled between the eyes, the failure of one eye will hamper the cues' utility.

There are dynamic cues as well, which are the motion of the scene and the known motion of the subject, and are available from each eye individually (these are known as *monocular* cues). This type of information can be attained by a single eye and is thus valuable for prey species which have non-overlapping fields of view as well as from a fault-tolerance perspective: losing an eye does not significantly affect the ability to perceive motion cues. Visual phenomena such as occlusion (one object hiding behind another object), motion parallax (closer objects appearing to move more than distant objects), and focus-of-expansion and axis-of-rotation (motion vectors in the visual field indicating the motion of the subject) are also considered major cues to motion. In addition to these cues from the early vision system, the computation of three dimensional structure is driven by cues from higher vision processes (Palmer, 1999, Chap. 5, pp. 504-511). Some of the cues in higher vision are object scaling, relative position, and known identity of objects. These cues will not be further addressed.

## 1.3 Background

The approach toward image analysis and artificial vision which has been used for approximately the past 30 years is in some ways similar to how "artificial intelligence" (AI) has been approached: the researcher examines a problem, finds some way to abstract the data and reduce the amount of information required to solve the problem (modeling), and proposes an elegant and mathematically "correct" solution to the problem. Unfortunately for both vision and AI, this historically has tended toward solutions that are "brittle" – that is, their problem-solving ability degrades or fails entirely with even small changes in the environment for which they are not prepared or for which they were not designed.

More recent approaches have attacked the problem from an angle which more readily accepts imperfect sensors and noisy data, and which accepts the impracticality of attempting to model the world in detail (Mead, 1989; Brooks, 1986). Brooks outlines an approach which he calls "the subsumption architecture" wherein a robot's behavior is composed of layers of parallel behaviors, possibly overlapping or redundant, each of which acts in a direct, predictable, and reactive manner to external stimuli or to outputs from other modules. Pinker (1997) corroborates this design methodology by theorizing that various computational "modules" comprise the human brain and result in complex human behavior. Needless to say, we are nowhere near mimicking mammalian (nor even insect) behavior in robots, but it is encouraging to find a system of robotic design, a homologue of which has been found in biology. Brooks' robots rely on the layers of parallel behaviors to produce complex overall behavior rather than being explicitly encoded with a complex global control system. This is attractive because it allows the designer to focus on individual simple behaviors without being bogged down by trying to understand all the complexities of a normal sense→model→plan→execute type of control system and because these architectures are

more robust in unknown or unpredictable environments. The subsumption architecture supports the use of vision as a primary sensory modality for robots, rather than sonar (another popular means of sensing the environment), because of its vastly higher information content. In keeping with this method of robot design, therefore, it seems reasonable to approach robotic vision from a bottom-up perspective. This project was performed with the subsumption architecture in mind and implements two behaviors, vergence control and horizontal tracking, in a parallel and independent manner. The strategy is based on direct sensory input using population decoding to control the position of the cameras. Further discussions in Chapter 7 about its implementation in hardware will support this design style in analog VLSI circuits and encourage its use in real-time mobile robots.

One of the first and most geometrically obvious ways of perceiving the three-dimensional environment involves analyzing two images which come from two different cameras and finding salient points in the images which correspond with each other. By applying geometry and using known constants such as the distance and angle between the two cameras, the cameras' focal length, and the resolution and size of the image sensor, the matched points can be given an X (left-right), Y (up-down), and Z (in-out) value. Thus, a feature in the scene which appears in both cameras is given real-world Cartesian coordinates. These can then be fed to higher-level processing elements, such as to control the robot's motor functions.

The main problem with this approach is that significant processing needs to be done to the images prior to stereo reconstruction to extract the salient features. In an image with any real-world complexity this problem is quite difficult to solve and the type of solution may not apply to the specific image; the types of analyses described, for instance, in Haralick and Shapiro (1992) often assume that some type of function $f(x, y, ...)$ can be mapped to the image, such as lines or 3D contours. This technique in effect requires some degree of image "understanding", interpretation, or assumption prior to doing the stereo computation. In well constrained environments this method is appropriate, since the range of situations in which the agent may find itself is quite limited. A recent example of a well constrained environment may be found in the work of Kolesnik and Baratoff (2000), who use robots to navigate sewer lines in search of cracks in the concrete. The general lack of robustness often found with the image/feature analysis approach is offset by the inflexibility of the environment, and the costs associated with designing such a system are commensurate with the variety of environments in which it can serve. Another possible advantage of direct image analysis techniques is that if the correspondence (point-matching) problem is adequately solved for greatly dissimilar images (images with objects very close to the cameras relative to the cameras' separation distance), any optical or perspective-induced distortion can be overcome, since it is accounted for in the geometrical modeling of the system. This may be used to overcome places in the image where stereofusion is not possible, such as in regions with a high degree of vertical disparity (Woods *et al.*, 1993).

Haralick and Shapiro (1992) describe many low-level mathematical operators for dealing with images prior to higher level analysis. These include (sampling from the text's table of contents): thresholding and segmentation; region analysis with extremal points, spatial moments, and signature analysis; statistical pattern recognition using various rules and with neural networks; binary morphology such as dilation, erosion, opening, closing, and set theory to describe them; neighborhood operators such as region growing and shrinking and convolution and correlation; conditioning and labeling with various types of filters and zero-crossing detectors; facet models with gradient and derivative analyses; texture descriptions; image segmentation; and arc extraction including the Hough transform technique used by Kolesnik and Baratoff (2000). These techniques have a distinct flavor of mathematical formalism with *seemingly* little regard to examining how nature has solved the vision problem, apparently many times over (Pinker, 1997, pp.211–284). (In all fairness, the text does claim a biological motivation for Gaussian noise smoothing and edge detection, however this is in general contrast to the rest of the book.) A recent application of this type of image analysis can be found in the work of Knight and Reid (2000). They use a point-matching technique and construct a depth-map of the scene to calibrate a robot's stereoscopic cameras for further vision-based modeling of the office environment. Their technique is based on several matrix transforms and geometrical theorems.

Another major class of techniques which do not depend on trying to fit the image to a model or function is one which instead uses only local image information for depth perception. These are known as *correspondence-less* algorithms. It is assumed that a 3D scene will project nearly identical images into two stereoscopic cameras if the distance to the scene is large compared to the distance between the two cameras. The two images will differ in horizontal position by a slight amount depending on the distance from the cameras and the horizontal distance between the cameras, resulting in what is known as horizontal disparity or stereoscopic disparity. Whereas the point-matching techniques implicitly or explicitly attempt to find a model to describe the image and to

identify which point in one image matches which point in the other image, the correspondence-less techniques (disparity-measurement techniques specifically) generate only a scalar or set of scalars which indicate the amount of horizontal shift in the two images or parts thereof, with no regard to specific points or features in the images. It is this loss of global information which in general prevents the disparity techniques from detecting large amounts of disparity; disparities from extremely close or extremely far objects can be too great to be useful.

Disparity is therefore most useful for small ranges of depth around the *horopter*, which is the locus of points that project to the same place in both retinas or imaging planes and which therefore produce zero disparity. The horopter is thus a curved surface in front of both cameras. The region in which points in front of and behind the horopter which are interpreted by the vision system as single points, rather than distinct points in the left and right visual fields, is known as *Panum's fusional area*. We use the fact that disparity is only a relative distance between the horopter and the target to verge the cameras. Note that any use of the word disparity here is intended as the horizontal shift between the two images. There is also vertical disparity which is caused by a misalignment in the elevation (up-down angle) of the two eyes or by the perspective-induced differences in the apparent height of an object (Woods *et al.*, 1993). The measurement and use of horizontal disparity has been studied extensively; it is reviewed in more detail in Chapter 2.

## 1.4 Previous Work

The past several decades have seen an enormous amount of research into computer and robotic vision. The following summarizes some work relevant to this thesis.

Sanger (1988) and Qian and Zhu (1997) both provide a good mathematical basis for the disparity techniques used in this project. They introduce the notion that complex phase in the frequency domain can be computed at various points in the image with Gabor filters. Complex phase in the frequency domain corresponds directly to disparity in the spatial domain. This directly fits the biological disparity energy model espoused by Ohzawa *et al.* (1996; 1990; 1990; 1986). Sanger and Qian both show depth maps calculated from random-dot stereograms as well as real images using this technique. Chen *et al.* (1994) elaborate on the notion that a system of Gabor filters with a range of spatial widths and tunings can be used to compute depth. Hansen and Sommer (1996) use a constant-size Gabor filter applied iteratively over a subsampled image in a coarse-to-fine manner. They use this information to control the vergence of a pair of cameras and to estimate depth to a target. Cozzi *et al.* (1997) compare the performance of phase-measuring filters introduced by Sanger and Fleet, which are all based on the Gabor filter.

Ohzawa *et al.* have written extensively on the neurological basis for stereoscopic vision in cats. They support the concept of phase-based (as opposed to position-based) disparity detectors in the simple cells of a cat's visual cortex. These cells correspond to various *receptive field* ("RF") widths and are tuned for various disparities; simple cells tuned for stereoscopic disparity respond maximally to a particular disparity at a particular spatial frequency. This is the architectural basis for the disparity model used in this project.

Batista *et al.* (2000; 1997; 1996) and Araujo *et al.* (1996) implement real-time image tracking and vergence control with a multi-degree-of-freedom robotic camera head. By computing optical flow and using cross-correlation (a popular way to measure disparity) to determine image disparity, they are able to control the angles of the head and eyes. Their control scheme consists of an image capture board, various types of "purposive" behaviors, a state machine, computational modules, and a detailed model of the head mechanics, all running on a PC-based system. They use various filters and matrix transforms to predict motion and to calculate the correct angles for the eyes and head. Their approach is very much from a "control-theory" viewpoint, where the motion of the robot is directly based on computing a trajectory and using inverse-kinematics to figure out the angles of the joints.

The above discussion leads us to consider the control of the vergence angle between the two eyes. Other than the depth of the objects in the scene, the vergence angle is the only other parameter which determines image disparity. The vergence angle and any remaining disparities can be used as an estimate of objects' depths in the scene, and thus vergence control is useful and important. This project controls the vergence angle and estimates distance to the target.

Alvarez *et al.* (1999; 1998) and Semmlow *et al.* (1998) have studied the dynamics of human ocular vergence and have supported the theory that vergence is a two-step process: the first step is triggered by an onset of image disparity and produces one or two saccade-like (fast) motions to minimize the disparity. The maximum speed of the response is determined by the amount of disparity. If the first movement does not bring the vergence to within approximately 80% of its final value then another fast movement is generated. During these movements visual feedback is not used, hence its controller most resembles an open-loop controller. After the initial one or two

movements a second-order visual-feedback control system is invoked which adapts the vergence to small (slow) changes in the disparity, and is responsible for the extremely accurate positioning of the eyes to achieve image convergence (minimization of disparity). They have also shown that the control of horizontal tracking (smooth pursuit) is separate from that of the vergence control, the former being more precise for slowly moving targets. Hung *et al.* (1997) have shown that convergence movements are faster than divergence movements for the same (symmetric) stimuli. Howard *et al.* (1997) discuss the amplitude and phase (dynamics) of the vertical vergence response to varying stimuli.

Popple *et al.* (1998) have shown that horizontal vergence response in humans is affected by the area of the image where the disparity exists. They show that the vergence response does not necessarily react to the disparity of the entire visual field, nor to a target presented in that field, nor to disparities simply within the fovea, but instead integrates over a region of at least 6° of visual field to determine the vergence response. Stevenson *et al.* (1999) have shown that foveal disparity targets are given preference, however, in inducing a vergence response, compared to targets farther out in the visual field, and have indicated that the vergence response is a result of a weighted integration over the visual field, which supports Popple *et al.* They also confirm that a larger disparity target is more effective in driving the vergence response than a smaller target. Mallot *et al.* (1996) have shown that the vergence response is initiated by the correlation of the left and right images, influenced by the density of the image (they used random-dot stereograms), and that the disparity-detection system averages multiple disparities in a visual location so that only one depth is perceived.

The dual-mode vergence behavior has motivated some researchers to a develop a dual-mode model. Cova and Galiana (1994; 1995) have developed a neural model to account for accommodation (focus) and vergence control, with a neurophysiological basis. Their model incorporates both the fast and slow responses by eliminating a negative-feedback connection during the fast response. Patel *et al.* (1997) also provide a physiologically-based neural model of vergence control, but instead of using a switching mechanism to elicit the slow and fast responses, they rely on inherent nonlinearities in the neural elements to provide the behavior. Hung (1998) provides a Matlab-based model for fast and slow vergence response, with apparently little attempt to mimic neurobiology. All three of the above models have been simulated and appear to provide responses similar to the biological case. In the first and third case, the inputs are simply the desired vergence angle. In the second case the authors provide a block labeled "disparity detectors" which then similarly provides a desired vergence angle as input to the rest of the system. None of the models provide a means of measuring disparity or computing the desired vergence angle, but Patel *et al.* suggest than an array of disparity-tuned cells provide input to the vergence controller (the "disparity detectors" block). This part of the model has been adopted to the current project.

Non-biological vergence control models have also been developed. Olson and Potter (1989) approach the problem from a signal-processing perspective and use a cepstral filter to calculate image disparity. The results of the filter are then used to control the vergence angle in real-time. Yim and Bovik (1994) subsample the images and use a hierarchical coarse-to-fine Laplacian Pyramid scheme with zero-crossing-detectors and sign-correlation to measure disparity and control the vergence angle. Bernardino and Santos-Victor (1996) use a log-polar function to create a foveal area in their image (higher resolution in the center than at the edges) to reduce the required computation and show that using a log-polar representation yields better results for vergence control than a Cartesian representation, in addition to being faster due to the reduced image size. Marefat *et al.* (1997a; 1997b) describe their use of disparity to control the vergence of stereoscopic cameras. They perform thresholding on the image and use windowing to divide it up for use in the disparity computation. Piater *et al.* (1999) use a Cartesian logarithmic subsampling method to create a foveal region in the center of the image. They then use column-wise stereomatching and parameter correlation as inputs to a reinforcement learning algorithm. The algorithm determines the best parameters to use in controlling the vergence of their stereo cameras, and they have shown it to be superior to human-determined values.

In general the biological models differ from the non-biological models in the extent to which the computations carried out are possible and feasible in a neural system. The biological control methods and input selection are typically a first or second order controller with population-encoded inputs, and may use some nonlinearity, such as thresholding. The non-biological models may use various types of signals-and-systems approaches that have been developed over the years for other control systems, and often involve filters and matrix operations that are typically not associated with biological neural processing. The disparity computation has been shown rather conclusively in Ohzawa *et al.* (1997) to be represented as an energy in a cat visual cortex, rather than being

decoded trigonometrically through some type of "arctangent" cell, as has been used in Hansen and Sommer (1996) and Marefat *et al.* (1997a). Also, the non-biological models process the images in ways that have not been shown to occur in neurobiology. Specifically, segmentation and binarization of an image in Marefat *et al.* do not appear to have counterparts in neurobiology. Conversely, the biological models tend to use the entire field of view, which may be an inefficient use of resources.

So far the previous work that has been covered has used software to compute disparity and control the vergence angle or to do tracking. There have been efforts to perform these tasks in hardware as well. We review some of them here. Mahowald and Delbrück (1989) developed a chip capable of detecting disparity via two one-dimensional detectors arranged on a single chip. They developed a static and a dynamic version of their chip. Erten and Goodman (1996) present an analog VLSI chip which performs correlation between two images of a stereo pair and outputs a disparity map. Asai *et al.* (1999) present a biologically-inspired analog MOS circuit which tracks objects and compensates for head/eye movement. Etienne-Cummings *et al.* (1999) describe a mixed-signal VLSI imaging chip which can perform arbitrary filtering with 3x3, 5x5, 7x7, 9x9, and 11x11 kernels. While this chip is not designed specifically for stereoscopic imaging, it is reasonable to extend the concept they present to the stereo domain. Shi et *al.* (2000; 1999) introduce a VLSI architecture for implementing Gabor filters in real-time hardware and show how these can be used for tracking and vergence control.

## 1.5 Project Goals

This project aims to combine the research on biological disparity computation with a model for vergence control and to add a novel disparity-energy driven horizontal tracking mechanism. *Horizontal tracking* is the ability of the eyes to move horizontally such that the object in question remains centered in both the left and right image. The system presented here is driven only by visual information, without knowledge or higher-level symbolic description of the scene. In general, this is different from the approaches that have classically been taken in computer vision; it is our conviction that by mimicking the architecture and computational approach found in the mammalian visual cortex for early-vision, a system that is as robust, flexible, and low powered will be possible in the long term. It appears the disparity-energy techniques discussed by Ohzawa *et al.* (1997) and Qian and Zhu (1997) have not been directly input to a vergence control system, nor does it appear that the vergence control models here have been driven by any type of disparity-energy measurement system. This project therefore aims to combine the two. This project concerns itself with using the disparity-energy from two images from a pair of stereoscopically-mounted cameras ("eyes") and producing the appropriate eye movements to achieve both convergence and horizontal tracking. Additionally, in this project horizontal tracking has been implemented in a way that does not necessarily appear supported by biological research, but which offers a simple way to track objects of interest. By combining these three elements - disparity measurement, vergence control, and horizontal tracking, it is hoped that a useful biologically-inspired vision system can be created.

It is difficult to compare this research with other vision-related research, so at this point we cannot argue whether this particular implementation is better in any measurable degree than other systems (Hansen and Sommer, 1996; Marefat *et al.*, 1997b), but we believe it to be a step in the right direction. In keeping with a biological motivation for vergence control, this system avoids standard computer vision techniques such as modeling its environment, generating trajectories, computing Hough transforms, performing statistical analyses, or attempting point matching (Haralick and Shapiro, 1992; Knight and Reid, 2000; Kolesnik and Baratoff, 2000). Rather, the system emulates current neurobiological models by detecting and exploiting image disparity and employing a high degree of parallelism among locally connected simple mathematical operators, including summations, center-surround receptive fields ($RF$s), and winner-take-all networks with nonlinear activation functions (thresholding) (Lande, 1998; Mead, 1989; Kandel *et al.*, 1995). Although this project is software based, it is developed with the hope that it will be expanded to a full hardware-based implementation, thereby achieving the goal of a robust fully parallel real-time low-powered vision component for use in a mobile robot.

From an algorithmic perspective, this research does not set out to improve upon any specific algorithm or means of computation. Therefore a quantitative comparison of accuracy, repeatability, etc., is not appropriate. From a performance perspective, while the algorithms and methods presented here are carried out in software, the ultimate goal of this research is to create hardware that accomplishes the same task. The software therefore is not optimized for a serial computer, but rather it simulates the actions of a highly parallel hardware system, and so it does not compare easily or fairly with other software-based real-time vision systems (Batista *et al.* (2000; 1997; 1996)). Similarly, the accuracy and error measurements shown in Chapter 6 are reflective of the

low-cost hardware platform we use here, whereas Batista *et al.*, for example, use an admittedly very expensive and precise mechanical setup. Probably the most appropriate way to compare this work is against other hardware implementations of the same task, however this author has not found any other purely-hardware implementations of vergence and tracking.

Other hardware-based disparity-measurement systems have not been based on disparity-energy (Higgins and Koch, 2000) or are limited to a single dimension and have not set out to control camera movement (Mahowald and Delbrück, 1989). Some are two dimensional (Erten and Goodman, 1996) and are used to control camera movement (Lu and Shi, 2000), but still perform critical computation off-chip in a serial computer (Lu and Shi), and still do not use disparity energy as their metric. As far as we know this is the first proposal of a multichip, layered, disparity-energy-based, analog, continuous-time vergence and tracking controller.

Some other points of comparison which may be considered against other systems that ultimately perform the same task (Hansen and Sommer (1996), Batista *et al.* (2000; 1997; 1996)) are those of power consumption, scalability, and robustness. The power consumption of a multichip system has been shown by Higgins and Koch (2000) to be vastly superior (9.2mW peak) to that of a software-based system; the power consumption of the processor cooling fan alone for an Intel Pentium III (Intel Corp, 2000, p. 92) processor is shown to be approximately 1.2W. The layered nature of the system presented in Chapter 7 allows for computation to be added or rearranged as needed, such as by adding more spatial filtering, more disparity tuning, motion computation, different controllers, etc., with only linear increases in complexity and power consumption, and no noticeable decrease in system speed. Increasing the computational ability of the software-based systems would either slow down the system or require more processors and much more complex software to keep the computation in constant time. The proposed hardware system will also not be as susceptible to things such as software crashes, power glitches, etc. The ability to add more computation also increases the system's ability to perceive depth in arbitrary images, rather than the toy images used in this software project and by others such as Shi.

**Chapter 2**

# DISPARITY MEASUREMENT

Stereoscopic disparity is the difference in the two images of a scene caused by the spatial separation of two horizontally spaced cameras. It is assumed that for normal viewing conditions the difference is simply a horizontal shift, representable as a phase offset in the frequency domain, and true differences in perspective caused by the different camera vantage points are ignored or considered negligible. This section describes the use of quadrature-phase Gabor filters for disparity measurement, as proposed by Ohzawa *et al.* (1997); Qian and Zhu (1997); Sanger (1988). *Cell* will be used interchangeably with *filter* where the context is clear.

## 2.1 Gabor Filters and Disparity-Tuned Complex Cells

Let us begin by considering two one-dimensional images and the following two Gabor functions:

$$f_{l,1}(x) = e^{\frac{-x^2}{2\sigma^2}} \cos(\omega_0 x + \phi_l) \tag{2.1}$$

$$f_{r,1}(x) = e^{\frac{-x^2}{2\sigma^2}} \cos(\omega_0 x + \phi_r) \tag{2.2}$$

where $\sigma$ is the width of a Gaussian envelope and $\omega_0$ and $\phi$ are the spatial frequency in cycles/pixel and phase, respectively, of a cosine, for $l$eft and $r$ight filters. If the functions are centered at some point $x_0$ in their respective images, then a convolution of the image with the function will indicate the amount of image energy at the phase-shifted location of the sinusoid, at the frequency $\omega_0$, within the Gaussian. In other words, the output of the convolution is the value at one location along the frequency axis of the even (real) part of a mini-Fourier transform, performed at the filter location, where the transform is most sensitive to $\phi$-shifted cosines of frequency $\omega_0$. Since the phases for the left and right cosines are different, each filter can respond specifically to a particular shift in the image. By using filters with a phase difference of $\Delta\phi = \phi_l - \phi_r$ and combining their outputs, the presence of a certain disparity between left and right images can be checked for. Thus, the response of a *simple cell* is

$$r_{simple,1}(x_0) = \int_{-\infty}^{+\infty} [I_l(x) \cdot f_{l,1}(x_0 - x) + I_r(x) \cdot f_{r,1}(x_0 - x)]dx \tag{2.3}$$

where $I(x)$ is the intensity at location $x$ (Qian and Zhu, 1997).

A *complex cell* is a cell which combines the squared outputs of two pairs of simple cells in quadrature phase. By using filters in quadrature phase, the output becomes independent of stimulus contrast and mostly independent of absolute phase, yielding a *disparity energy*. This comes from the trigonometric identity $\sin^2 x + \cos^2 x = 1$. That is, the second cell of the pair uses a sine function instead of a cosine to compute its output. The sine function in effect is the odd (imaginary) part of a Fourier transform at the filter location:

$$f_{l,2}(x) = e^{\frac{-x^2}{2\sigma^2}} \sin(\omega_0 x + \phi_l) \tag{2.4}$$

$$f_{r,2}(x) = e^{\frac{-x^2}{2\sigma^2}} \sin(\omega_0 x + \phi_r) \tag{2.5}$$

The response of complex cells in a cat's visual cortex has been found by Ohzawa *et al.* (1997) to follow this mathematical description quite well. We can define $r_{simple,2}$ similarly to $r_{simple,1}$ by using $f_{l,2}$ and $f_{r,2}$. A major function of a complex cell is that it responds positively to stimuli in which both the left and right eye have the same contrast sign, but has a negative (antagonistic) response to stimuli in which the left and right eyes are receiving oppositely-contrasted stimuli.

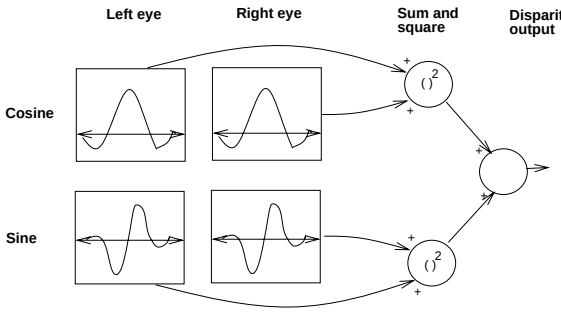



FIGURE 2.1. Block diagram of complex cell tuned for zero disparity($\phi_l = \phi_r = 0$). Blocks on the left represent simple cell Gabor filters, with a nonlinear squaring function.

Although this project does not make use of the following approximation explicitly, the response of a complex cell (the *disparity energy*) is:

$$r_{complex} = (r_{simple,1})^2 + (r_{simple,2})^2 \tag{2.6}$$

$$\approx c^2 \left|\widetilde{I}(\omega_0)\right|^2 \cos^2\left(\frac{\Delta\phi}{2} - \frac{\omega_0 D}{2}\right) \tag{2.7}$$

where $c \equiv 4\int_0^\infty d\omega \left|\widetilde{f}_l(\omega)\right|$ is a constant and $\left|\widetilde{I}(\omega_0)\right|^2$ is the Fourier power of the image under the RF (receptive field) at the preferred frequency $\omega_0$ (Qian and Zhu, 1997). Fig. 2.1 diagrams how the complex cell computes its response.

The preferred disparity of the complex cell is assumed to be small compared to the width of the RF, and is defined as the difference in phase of its constituent left and right simple cells, divided by their frequency:

$$D_{pref} \approx \frac{\Delta\phi}{\omega_0} \tag{2.8}$$

which carries units of linear distance (in this project, the units are pixels). A simple cell, and therefore a complex cell, cannot detect nor be tuned for any disparity outside the range $\left[-\frac{\pi}{\omega_0}, \frac{\pi}{\omega_0}\right]$, although Cozzi *et al.* (1997) tell us the useful range is actually approximately $\left[-\frac{2\pi}{3\omega_0}, \frac{2\pi}{3\omega_0}\right]$. These ranges can perhaps be better interpreted if expressed as $\left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right]$ and $\left[-\frac{\lambda}{3}, \frac{\lambda}{3}\right]$, respectively, where $\lambda = \frac{2\pi}{\omega_0}$ is the wavelength of the sinusoids in pixels. If a complex cell is tuned for disparities outside the available range, then its maximum output will occur $\pi$ radians away from the correct location. In other words, if the stimuli fall outside the middle cycle of the sinusoid under the RF, at locations less than $-\frac{\lambda}{2}$ or greater than $\frac{\lambda}{2}$, then a problem known as *phase-aliasing* occurs, since the periodic nature of the sinusoid makes disparity of the stimuli ambiguous. Phase aliasing is the phenomenon caused by trying to map an infinite range of disparity inputs to a finite range of output energies. Since each output of the cell can be interpreted as being caused by a legal disparity, when a disparity is presented that is beyond the limits of the cell, the output energy must be ambiguous with a legal disparity.

How are the filter parameters, $\omega_0$, $\sigma$, and $\phi$ determined? For simplicity, we choose parameters which will allow the same filter shape for all spatial scales. First we choose $\sigma$ to be proportional to the entire width of the filter, typically anywhere between 7 and 50 pixels:

$$\sigma = \frac{kW}{2\pi} \tag{2.9}$$

where $k$ is a constant and $W$ is the entire width of the filter. A value of $k = 1$ allows for a "nice" Gaussian curve over the width of the filter with wavelength $\lambda$, so that the number of cycles of the

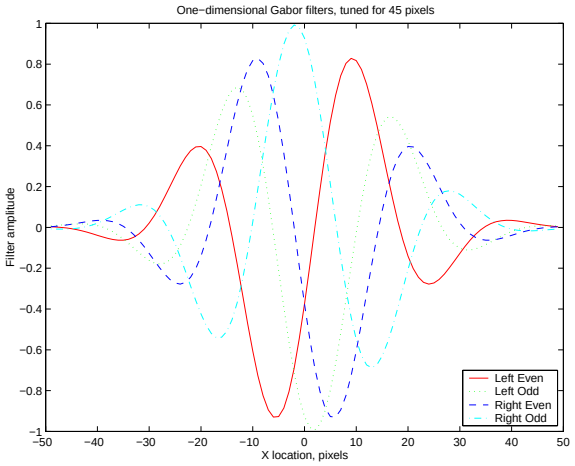



FIGURE 2.2. Even and odd Gabor filters with $k = 1$, $t = \frac{1}{3}$, $W = 99$, $D_{pref} = 45$ pixels, yielding $\Delta\phi = 8.65$ rad, $\lambda = 32.67$ pixels/cycle, $\omega_0 = 0.19$ cycles/pixel, and $\sigma = 15.76$.

sinusoids will always be the same for any given Gaussian. The pixels referred to here are the image pixels after appropriate data-reduction (subsampling) of the original CCD image. In the hardware architecture which will be discussed in Chapter 7, they refer to the adaptive photocell's output.

Next we choose the wavelength $\lambda$ to be proportional to $\sigma$ by a factor of $t$, and thus also proportional to $W$:

$$\lambda = t\sigma = \frac{tkW}{2\pi} \tag{2.10}$$

so that

$$\omega_0 \quad = \frac{2\pi}{tkW} \tag{2.11}$$

With $k = 1$ and $t = \frac{1}{3}$, the two Gabor filters in Fig. 2.2 are generated.

Fig. 2.3 shows five different complex cells, each tuned for a different disparity. Notice that the complex cells respond sharply to differences in disparity (the purpose of the cell), but they respond rather weakly to the overall position of the stimuli, i.e., their absolute horizontal position (phase) within the RF. Qian (1997) shows how this is a characteristic and desirable feature of a complex cell, which is not possible using only disparity-tuned simple cells. Ohzawa *et al.* (1997) confirm this behavior in the cat visual cortex.

## 2.2 Disparity-Tuned Filter Bank

Our first attempt at using disparity-tuned cells to measure disparity was with a single cell tuned for zero disparity, located at each spatial position. The vergence controller simply tried to maximize the energy reported by the cells using a hill-climbing algorithm. Chapter 4 talks about this in greater depth; the introduction to disparity already presented provides enough background for this type of vergence control.

After the zero-disparity-tuned cell method was tried and found inadequate, a more "biologically plausible" architecture was developed using a bank of disparity-tuned cells covering a range of spatial widths and disparity-tunings, to be used at each spatial location, as suggested by Ohzawa *et al.* (1996) and Qian (1994). We will now discuss several issues pertaining to this implementation.

We determined that the filter bank at each location should provide sufficient information about the unique depth at that location to drive the vergence controller. This means that either the bank should report one single disparity to the controller, or that the bank should provide some type of average response across the various widths and tunings. This is known as "population decoding", since the information required is "encoded" in the population of cells ("neurons"). Qian (1994) was the first to show that this method can be used to compute a depth map. Fig. 2.4 shows a map

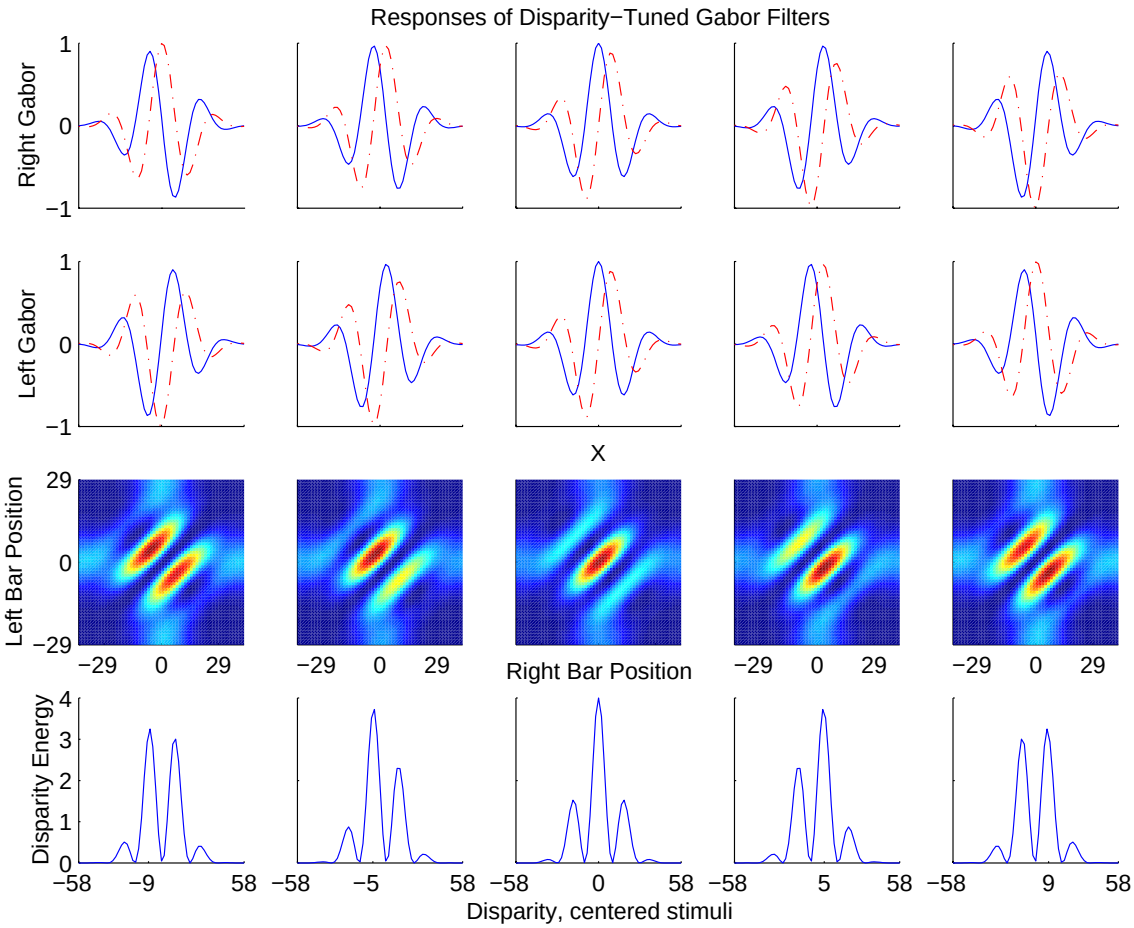



FIGURE 2.3. Simple and complex cell responses for a range of disparity tunings. The disparities for which the filters were tuned are -9, -5, 0, 5, and 9 pixels, from left to right column. The top two rows show the simple cells' Gabor filters. The third row shows the complex cells' responses to every stimulus position under the cells' RFs. The cell's sensitivity to disparity (upper-left-to-lower-right diagonal) and their insensitivity to stimulus average horizontal position (lower-left-to-upper-right diagonal) is seen clearly in the intensity of the plot. The fourth row shows the response profile of each cell to a range of disparities with the average position of the two stimuli centered under the RF of each cell, which is the same as the upper-left-to-lower-right diagonal of each corresponding third row plot. In both the third and the fourth rows the location and magnitude of the profile peak changes with tuning, as do the sidelobes. At the -9 and +9 tunings the sidelobes are almost as large as the peak and make the location of the peak ambiguous. This illustrates the range of tunings for a cell of a particular width and $t$ value. The stimuli were each a 1-unit high, 1-unit wide impulse. The parameters for all the filters were as follows: $k = 1$, $t = \frac{1}{3}$, $W = 59$, $\Delta\phi = -2.92, -1.63, 0, 1.63, 2.92$ radians, yielding $\lambda = 19.33$ pixels/cycle, $\omega_0 = 0.33$ radians/pixel, and $\sigma = 9.39$. In this case the limit of disparity tuning $\pm\frac{\pi}{\omega_0} = \pm 9.7$ which is truncated to $\pm 9$.

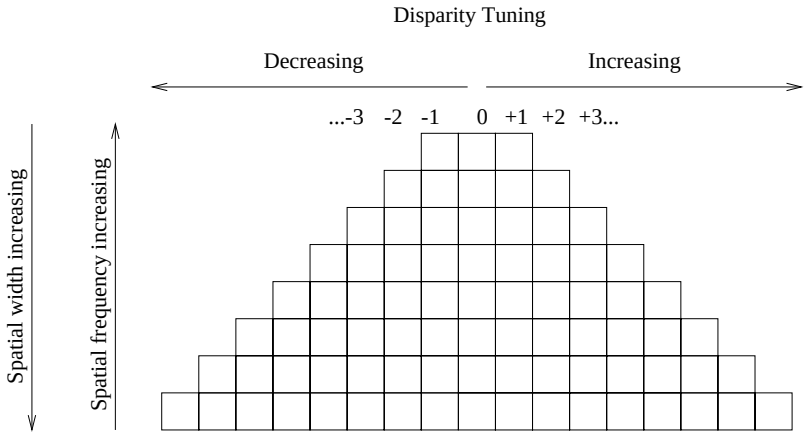



FIGURE 2.4. Disparity-tuned filter bank: Each square represents one complex cell with a particular spatial width and disparity tuning. As the spatial width increases the number of possible integral tunings increases, giving the pyramidal shape. Each pixel under the RF for a particular cell width "consumes" less phase relative to the width of the RF; the narrower cells at the top of the figure "consume" more phase for each pixel of tuning. The narrower cells therefore can represent less "real" disparity, measured in pixels, than the wider cells. The tradeoff is that the smaller cells respond better to higher spatial frequency (smaller image features). Ideally, each filter bank exists at each spatial location.

of the disparity-tuned filter bank. Each square represents one complex cell in the spatial-width vs. disparity-tuning space ("RF-Tuning" space) seen in Fig. 2.4. The vertical axis represents spatial scale: the filter width, RF size, or $\frac{1}{\omega_0}$ of the filter. The horizontal axis represents the disparity tuning of the cells. Thus, the cells within a row all have the same spatial width and cover a range of disparity-tunings allowable for that width, and the cells within a column all have the same disparity tuning (measured in pixels, not phase), covering a range of spatial widths. Since the cells toward the top of the map have a smaller RF than those toward the bottom, they also have a smaller range of allowable disparities, giving the pyramid shape. It turns out that using multiple cells with a range of RFs is identical to subsampling the image and using cells with a constant-sized RF, as in Hansen and Sommer (1996). Although using larger RFs in parallel on the same sized image rather than using single-sized RFs on a progressively smaller (subsampled) image is computationally slower, this approach is closer to how to biological vision systems effect the same task. Any insight gained through this slower architecture will hopefully be applicable to the hardware implementation ideas presented in Chapter 7.

It should also be apparent from Fig. 2.4 that there is much redundancy in the larger cells if their tunings are spaced similarly to those of the smaller cells. That is, the relative phase differences between adjacent cells in the wider RF rows are smaller than those in smaller RF rows, even though they represent the same number of pixels in disparity. Less cells can be used by spacing them further apart both in width and in tuning. By choosing a constant phase, or equivalently by choosing a certain number of tunings per row, we can reduce the amount of computation required for a given range of RFs. Using a constant phase to space them apart rather than a constant pixel distance would allow smaller cells to show small disparities while allowing the larger cells to represent more tunings spaced further apart, while using the same computational resources as the smaller cells. Unfortunately, for low-spatial-frequency images, this spacing leads to a coarser control of the cameras since it takes a greater disparity (away from zero) to move the cameras, since they can only be moved when a non-zero cell responds maximally. This results in the cameras possibly stabilizing on a vergence which does not reduce the disparity to zero (visible by the experimenter in the software's GUI, described in Chapter 3), but which is still smaller than the disparity of the first non-zero-tuned cell in either the positive or negative direction. The overall depth estimate, however, should not be affected too much because the average disparity of the image should still be available from the responses of the cells which *are* available. If the resulting spacing is too coarse about the zero tuned cells for a particular row, a decaying log function can be used to space the

cells an increasing distance from the zero tuning so small near-zero disparities are accounted for, but computation resources are saved for large disparities where the finer resolution is not required. It would not be surprising if similar spacing were found in biological systems.

## 2.3 Phase Aliasing: Description

Problems arise when the disparity presented to a range of cells is too large for their spatial width. Fig. 2.5 shows the outputs of a filterbank to several stimuli disparities. For each row where $W$ was the minimum required for that range of integral disparity tunings. Each row of subplots is associated with a particular stimulus width (1,5, and 10 pixels), and each column is associated with a particular input disparity (0, 2,...,10 pixels). Each subplot represents the same bank of complex cells (as in Fig. 2.4) ranging in both spatial width and disparity tuning. The value of each grayscale dot is the value of the RF profiles seen earlier for its respective cell. The grayscale contrast in each row of six subplots was normalized against zero and the maximum energy for *that row* of subplots, so subplots for a given stimulus width can be compared. The white dots at the right and bottom of each subplot mark the [row, column] with the overall maximum response, determined as the maximum of a sum-of-[columns, rows] (not shown) in each [row, column].

In general, the rows with the smaller RF respond less to the wider stimulus, and the wider RF rows respond more, exhibiting the spatial-frequency-tuning nature of the Gabor filters. As the disparity increases, the cells tuned for that disparity are the ones with the maximum response. The top row of plots gets darker as the disparity increases because the stimuli are approaching the end of the Gaussian envelope and so the response from the small cells is not strong. The response from the large cells is weak too, however it is because of the high frequency content of the stimulus rather than because the stimulus is out of spatial range of the cells.

As the stimuli widths increase (5 and 10 pixels), at large (+8) disparity, although the cells tuned to +8 in the bottom few rows are responding strongly, the cells at the opposite end of the range (negative large) within that spatial width are also responding strongly, and rows of narrower-RF are responding strongly as well, even though those narrower cells cannot possibly represent the disparities indicated by the larger RF cells. These rogue cells at the sides of the wide-RF rows and all over the narrow-RF rows represent the side lobes in the RF profiles of Fig. 2.3. The response of the small-width rows eventually tapers off as the stimulus exceeds the bounds of the RF. When the stimulus has a large low-frequency content, the maximum detectable disparity increases since the cells which exhibited aliasing with a higher-frequency content are not responding as much. Thus, the filter-bank's report of overall disparity is more accurate when the larger RF cells are favored by a wide stimulus. The maximum disparity input of 10 pixels, however, is still too large for the widest RF's limit of 9, and the reported disparity is therefore wrong even for the wide stimulus, as indicated in the bottom-right plot of Fig. 2.5.

The sum-of-columns (which produces the white dot at the bottom of each subplot) has a bias toward disparities which are shared by all the rows, since its sum contains the most elements, so the disparity indicated by the bottom-most row jumps around based not only on the true disparity but also on the spatial width of the stimuli. Similarly, the maximum point in the main part of the subplots do not necessarily match up with the sum-of-[rows, columns] maximum. The mismatch between the sum-of-columns maximum and the true disparity illustrates the phase-aliasing problem: as the disparity increases, the "perceived" disparity follows up to a limit, and then wraps around. The limit is increased as the width of the stimuli, since only the wider cells can represent those large disparities and they need the wide stimuli to guarantee their dominance.

Therefore, taking the maximally responding cell with this scheme, as Qian et *al.* (1997; 1994) do, whether the maximum comes from the sum-of-columns or from the middle of the map itself, may not always yield the correct sign of stimulus disparity, much less represent its actual value. Although narrow-RF rows may alias heavily, the preponderance of correctly-valued and correctly-signed wide-RF cell responses will in many cases make the sum-of-columns report a correct sign of disparity, and for reasonably small values of alias-inducing disparity, the sum-of-columns will still show the correct cell index for the input disparity. At very large stimulus disparities, however, the sum-of-columns response may exhibit somewhat chaotic behavior, as does the location of the maximum point in the map. With very large disparities, the effects of this aliasing are somewhat offset by the fact that the narrow cells respond rather weakly due to the RF tapering off toward the edge. Clearly some type of parameter-tweaking or architectural modification must be made to allow unambiguous disparity responses.

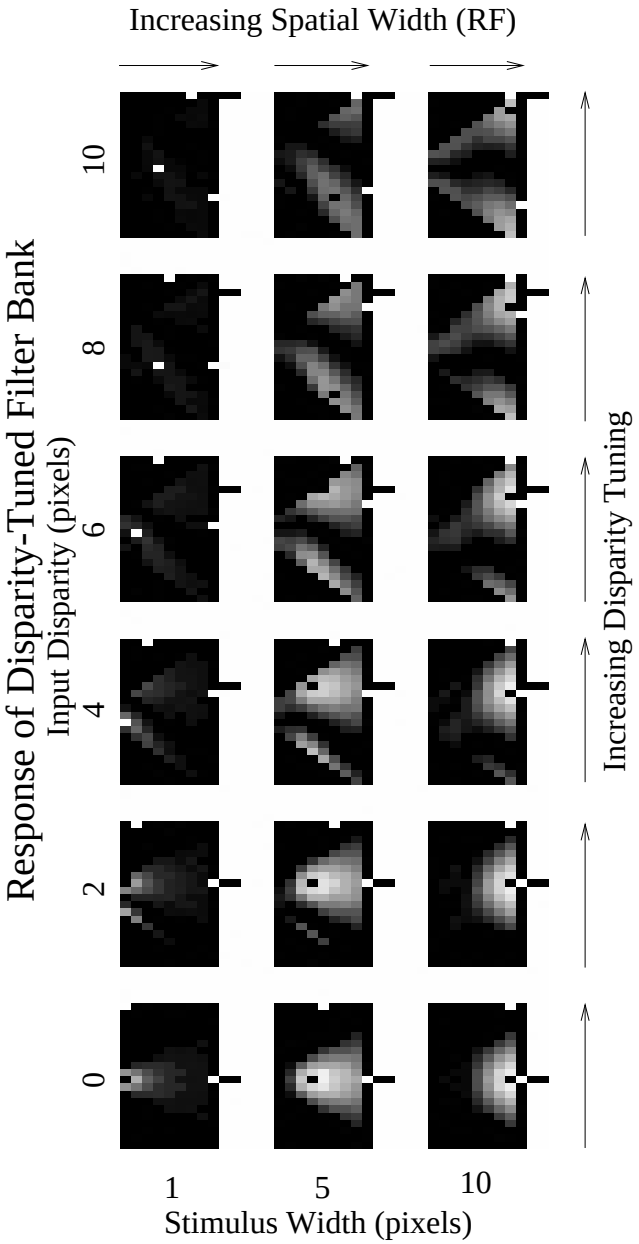



FIGURE 2.5. Response of disparity-tuned filter bank to various disparities and stimulus widths. A filter bank located at one spatial point was presented with centered ideal vertical bar stimuli. Filter parameters were: $t = \frac{1}{3}$, $k = 1$, with the ranges $W = 15$, $\omega_0 = 1.26$, $\lambda = 5$, $\sigma = 2.39$, and $D_{pref} = \{-2, -1, ..., +2\}$ pixels to $W = 59$, $\omega_0 = 0.32$, $\lambda = 19.67$, $\sigma = 9.39$, and $D_{pref} = \{-9, -8, ..., +9\}$ pixels. The stimuli disparity was $\{0, +2, ..., +10\}$ and their width was $\{1, 5, 10\}$ pixels. The mark in the bottom-most row of each subplot indicates the filter bank's consensus on what the true disparity is and is compared to the true disparity shown by a black mark under the subplot. The input disparity of 10 is actually beyond the range of the filterbank, but is shown where it would appear. The maximum point in the gray portion of each subplot is marked with an opposite-contrast dot.

## 2.4 Phase Aliasing: Analysis

The source for this aliasing comes from the fact that as the disparity is increased over the RF of the cells, the cell with maximum response does not always represent the disparity, and the index of the maximum cell cycles through all the cells. When the response profiles of the cells are superimposed, the profile with the maximum value at any particular disparity gives the index that is reported in this scheme, similar to that of Qian and Zhu (1997). For disparities greater than the maximum or minimum allowable disparity, the index of the maximally responding cell cannot possibly be of a cell that represents the illegal disparity, so as the disparity is increased beyond the legal limits, the index of the maximally responding cell cycles around to the opposite cell's index. By changing the frequency tuning of the cell and limiting the number of cycles of sinusoid which fall under the RF, the index-cycling behavior can be minimized or eliminated altogether. The major problem with this scheme, however, is that the cells are no longer as selective for a particular disparity, and lose disparity selectivity completely when the phase cycling is eliminated.

Figs. 2.6 and 2.7 illustrate the phase-aliasing behavior and the results of modifying filter parameters in an attempt to remove it. The top row of Fig. 2.6 shows that as $t$ is increased from $\frac{1}{3}$ to $\frac{1}{0.5}$ the disparity tuning becomes less sharp and the sidelobes disappear. The bottom row shows the energy output of a range of cells tuned to fill the maximum allowable range for their width ($W = 59$) and their output energy versus a centered input disparity. The range of the disparity tuning is shown in the x-axis and the range of the disparity input is shown in the y-axis; note that the tuning range increases as $t$ increases and the sidelobes appear as diagonal bands that do not reach across the entire subplot. As the cells' tunings increase from the negative to the positive limit, the response maximum tends to follow the input disparity, giving the major diagonal response. A horizontal cross section of the subplots on the bottom row yields the response of the range of cells for a particular disparity, from which can be extracted the index of the maximally-responding cell.

This maximum cycles from full negative to full positive, as shown in Fig. 2.7. By increasing $t$ and reducing or eliminating the sidelobes, the number of times the index cycles can be reduced. Within the range of $\left[-\frac{\lambda}{2}, +\frac{\lambda}{2}\right]$ (identical to [-90, +90] degrees), the ordering is sequential (and so the plot is monotonic), as expected. Outside the range, however, the ordering starts again. The index-cycling occurs when $t < 2$. The frequency along the disparity axis of the response profile is governed by $(\sin\omega_0 + \cos\omega_0)^2 = (1 + \sin(2\omega_0))$ (the frequency doubling is important) where $\omega_0$ is such that if $t > \frac{1}{0.5}$ or equivalently if $t > 2$ then one half-cycle of the sinusoid covers the width of the filter (according to how $t$ controls $\omega_0$). Outside $\pm\frac{\pi}{\omega_0}$, which are the legal boundaries of the filter, the sinusoids start again, and so the index of the maximally-responding filter also resets.

The major tradeoff with increasing $t$ as required to increase monotonicity in the maximally responding cell index is that it reduces the cell's selectivity to disparity, thereby degrading the cell's performance and ability to discriminate among various disparities. Sanger (1988) shows that a quadrature pair of Gabor functions, used with arctangents rather than energy to calculate complex phase directly, exhibits an error approximately the same as the ratio of spatial-widths to cycles. For example, a pair of filters with $t = \frac{1}{3}$ produce a disparity estimate within 33% of the actual disparity. A pair of cells with $t = \frac{1}{0.5}$ would report a disparity within 200% of the true disparity, which would mean both the true and the reported disparity are anywhere within the RF of the filter. This effect can be seen in the upper-rightmost subplot of Fig. 2.6 which exhibits no disparity selectivity compared to its width. It appears that for ideal stimuli such as in Figs 2.6 and 2.7, the disparity estimate is perfect and no problem appears. However, Sanger also shows that the actual error (within 200% of the true disparity in this case) is associated with the frequency content of the image, so for real images with unpredictable frequency content, estimating the disparity based on the maximally responding cell is not appropriate. For a given RF and range of tunings, any cell could respond to any given disparity, since they are all within the error estimate. In other words, the distinction from one maximally responding cell to another is not sharp, and so for an arbitrary image, the true disparity is ambiguous. It is not clear if this estimate error is directly applicable to energy-based disparity estimates as used here, but the qualitative similarities are apparent. Finally, Sanger indicates an upper limit of 1 on $t$, although the reasons are not clear. Therefore from a semantic viewpoint we may not use $t = \frac{1}{0.5}$, in addition to the performance reasons discussed here.

Fig. 2.7 also shows in the second row two different weighted energy plots for each value of $t$. These were calculated based on the product of the energy output of the cells and their index. The solid line follows:

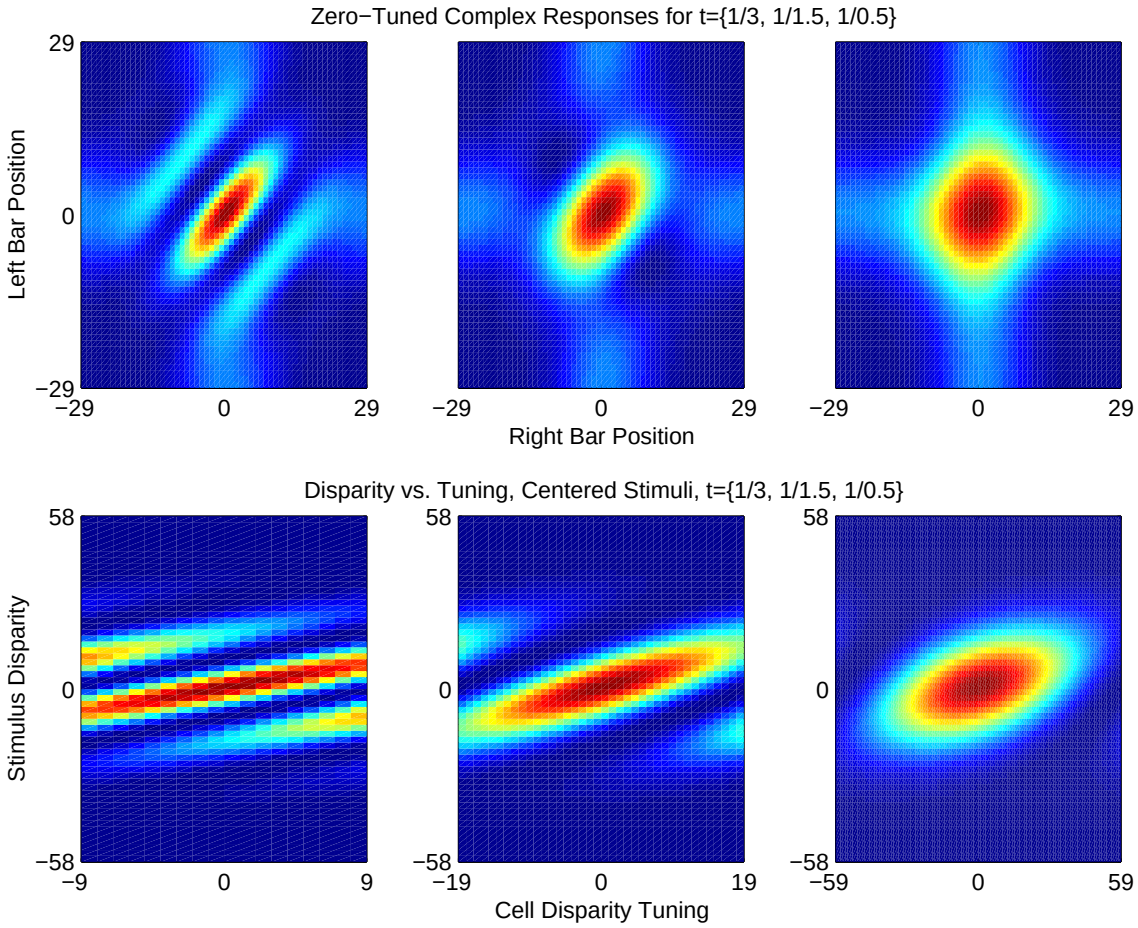



FIGURE 2.6. Complex cell responses for $t = \{\frac{1}{3}, \frac{1}{1.5}, \frac{1}{0.5}\}$. The top row illustrates the reduction in disparity selectivity for zero-tuned complex cells as $t$ increases. The bottom row shows the response of a full range of tuned cells (x-axis) to a full range of centered stimulus disparities (y-axis). The index of the maximally responding cell in the subplots of the bottom row sweeps from left (negative tuning) to right (positive tuning) as disparity increases, and then repeats, the number of repetitions depending on $t$.

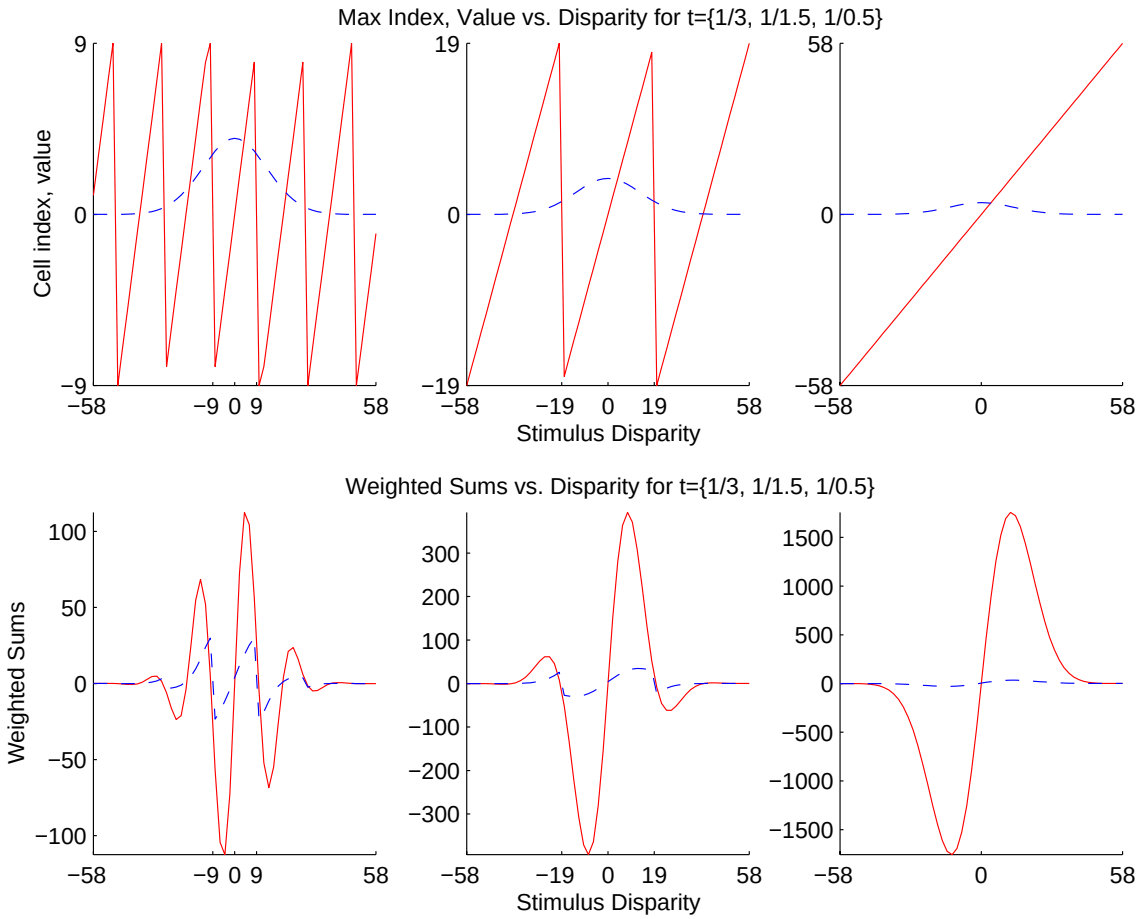



FIGURE 2.7. Index and energy of maximally-responding cell and total weighted disparity energy is shown. The top row (solid line) shows the index of the maximally responding cell as the disparity increases. The top row (dashed line) also shows the energy output of the maximally responding cell. Note that the maximum-index line is more monotonic as $t$ increases (left column to right column), but that benefit is traded-off with the reduced disparity sensitivity. The bottom row (solid line) shows for each disparity the sum of the products of each cell's energy with its tuning (index), i.e., the vector of dot-products, governed by $f_1(d) = \sum_{i \in D} E_i(d) \cdot i$, where $f_1(d)$ is the value of the solid line at disparity $d$, $D$ is the set of filter tunings, and $E_i(d)$ is the energy of filter $i$ for a particular disparity. The bottom row (dashed line) also shows for each disparity the product of the maximum-index and its energy value, governed by $f_2(d) = \max(D, d) \cdot E_{\max(D,d)}$, where $f_2(d)$ is the value of the dashed line at disparity $d$, $\max(D, d)$ is the disparity tuning of the maximally responding cell for disparity $d$ and $E_{\max(D,d)}$ is the energy output of that cell. These values may be used to determine how worthy the maximum-index is, or may be used directly as a measure of maximum index and its strength. The subplots indicate the legal disparity range on the x axis. Note that the maximum-index is monotonic within this region. This region covers $\pm\frac{\pi}{\omega_0} = \pm\frac{\lambda}{2}$.

$$f_1(d) = \sum_{i \in D} E_i(d) \cdot i \tag{2.12}$$

where $f_1(d)$ is the value of the solid line at disparity $d$, $D$ is the set of filter tunings, and $E_i(d)$ is the energy of filter $i$ for a particular disparity. The dashed line follows:

$$f_2(d) = \max(D, d) \cdot E_{\max(D,d)} \tag{2.13}$$

where $f_2(d)$ is the value of the dashed line at disparity $d$, $\max(D, d)$ is the disparity tuning of the maximally responding cell for disparity $d$ and $E_{\max(D,d)}$ is the energy output of that cell. Both lines follow the same qualitative trends, although their quantitative values are different.

## 2.5 Phase Aliasing: Solution

We consider two possible biologically-inspired solutions to the phase-aliasing (sidelobe) problem, the first of which is implemented in this project and is described in Chapter 4. The second solution is based on either lateral excitation or lateral inhibition and has not been implemented due to time constraints. The second solution is described here, however, because it resides entirely within the filter bank, whereas the first solution uses the output of the filter bank and some other components.

This second solution involves the large cells in the filter bank allowing (exciting) or disallowing (inhibiting) the smaller cells. When there is a large input disparity, all cells initially respond, even if only weakly. If the largest cells can unambiguously accommodate all input disparities, then the cells which represent very wide disparities can inhibit smaller cells so that when the disparity is very large, the smaller cells do not respond at all and hence do not alias. In this case the simplest architecture is as follows. For any row of cells in Fig. 2.8, there are inhibitory outputs from the outermost cells which disparities are not shared by any smaller row. The more these cells are activated, the more they prevent the unshared cells in all the smaller rows from responding, i.e., there is an inhibitory connection from the outer cell on each side of the triangle to all the cells in all narrower rows. Thus, the cells in the smallest rows have the most inhibitory connections. In addition, for any row, the next-to-outermost cell has an inhibitory connection to the cell with the same tuning in the next smaller row. The weights must be adjusted so that only when the outer cells in the wide row are firing maximally do they completely inhibit the cells in the smaller rows, and so that for rows that are not completely inhibited, their maximally-firing cells are all conjunctive (in agreement and not aliasing). A complementary solution would be to use the inner cells to excite the cells of smaller rows, rather than having outer cells inhibit. These two solutions have to be explored further.

## 2.6 Monoscopic Response

Another problem in addition to phase aliasing occurs in a more realistic usage of the disparity-tuned filter bank. Rather than measuring the disparity at only one point in the image as we so far have done, there should be a filter bank at each point in the image so a disparity estimate can be made everywhere. This other problem also occurs when a large disparity is presented to the system. The stereoscopic stimuli can be interpreted in one of two ways: the first way is the intended way, in which the stimuli are "fused" stereoscopically, i.e., both left and right stimuli fall within the RF of the underlying simple cells, and the cells between the stimuli respond as if the stimuli represent the same point in the image. This appears to be the reason for Panum's fusional area. The second "interpretation" occurs monoscopically, in which a small-RF cell is only wide enough to see one of the two stimuli. As the stimulus is passed over it, the cells of that RF and spatial point respond, albeit weakly, and give an incorrect reading of the disparity. They should not respond at all when there is only one input. Normalizing for frequency content and ignoring the Gaussian envelope for the moment, the response of a complex cell to only a single left or right point stimulus at the center of the RF is $\cos^2 + \sin^2 = 1$, regardless of the disparity tuning of the cell. Thus, the cell's actual response is exactly its Gaussian envelope times the cell's frequency-sensitive response; there is no nonlinear stereoscopic element. Ohzawa *et al.* (1997) show that a complex cell's response can be decomposed into two monocular elements (the ones which give the erroneous reading) and one binocular element:

$$R_c(X_L, X_R) = e^{-2kX_L^2} + e^{-2kX_R^2} + 2e^{-k\left(X_L^2 + X_R^2\right)} \cos\left[2\pi f\left(X_L - X_R\right) - \psi\right] \tag{2.14}$$

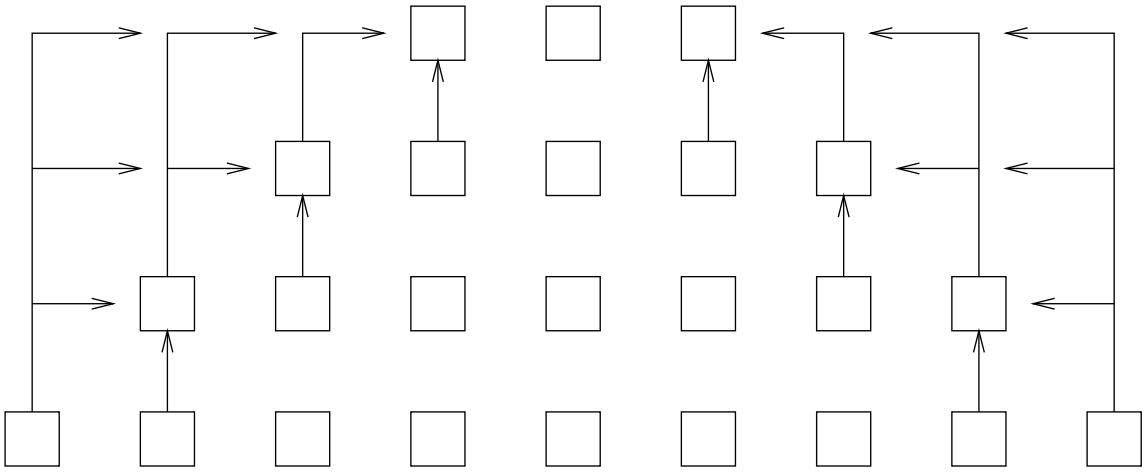


FIGURE 2.8. One possible solution to the sidelobe problem. Arrows represent inhibitory connections. Each row is inhibited by an outer cell of all the rows larger than it. Each outermost cell is further inhibited by the cell immediately larger than it with the same disparity tuning. The smallest cells therefore have the most inhibitory inputs.

where $k$ is the factor that determines the width of the cell, $X_L$ and $X_R$ are the left-eye and right-eye stimulus positions, $f$ is the spatial frequency, and $\psi$ is the disparity tuning phase. By thresholding the output of the cell to limit the majority of the monocular elements, we can prevent the cell from responding to an input appearing only in one eye. The threshold may not be a constant, however, because it must change with the frequency content of the image; hence, it may be called an "adaptive threshold". An image which matches the frequency tuning of the cell will have a much greater response than one that does not exactly match, and so the threshold needs to be greater for the matching case than for the nonmatching case. It has already been shown that the response of a complex cell to a purely monocular spike input is unity at the center of the RF and less toward the edges. We should threshold the output at some value greater than unity, but less than four, which is the maximum response of the cell to a binocular input. The sums of squares of the simple cells give us a usable threshold (using the terminology from Eqs. 2.1, 2.2, 2.4, and 2.5):

$$T = f_{l,1}^2 + f_{l,2}^2 + f_{r,1}^2 + f_{l,2}^2 \tag{2.15}$$

where $T$ is the threshold. If there is no input in the right, for instance, then $T$ will be 1 since the sum of the square of a sin and a cosine is one, regardless of the phase of the input (ignoring for the moment the tuning of the cell). The output of the complex cell is also one at this point, so the threshold should be multiplied by some small factor $k_t > 1$ to force a *greater-than* relationship and to suppress the output of the complex cell. When there is input to both eyes the threshold is $2k_t$ , but the output of the complex cell is 4, well above $2k_t$. Fig. 2.9 shows what a zero-tuned complex cell's response looks like before and after thresholding. The plots are slightly different from similar previous plots because the RF of the cell has been made smaller to emphasize the monocular response as the left and right bars are allowed to be the sole stimulus in the RF.

## 2.7 Practical Issues with Two Dimensional Images

So far the discussion has been concerning one-dimensional Gabor cells with one-dimensional images. The filters can be easily expanded to two dimensions by adding an unmodulated Gaussian component in the y direction to yield the following Gabor function in two dimensions:

$$f(x,y) = e^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}} \sin(\omega_0 x + \phi) \tag{2.16}$$

where all the variables take on their obvious definitions. This allows the filters to respond to vertical lines in a real two dimensional image, rather than single points in the one-dimensional world with

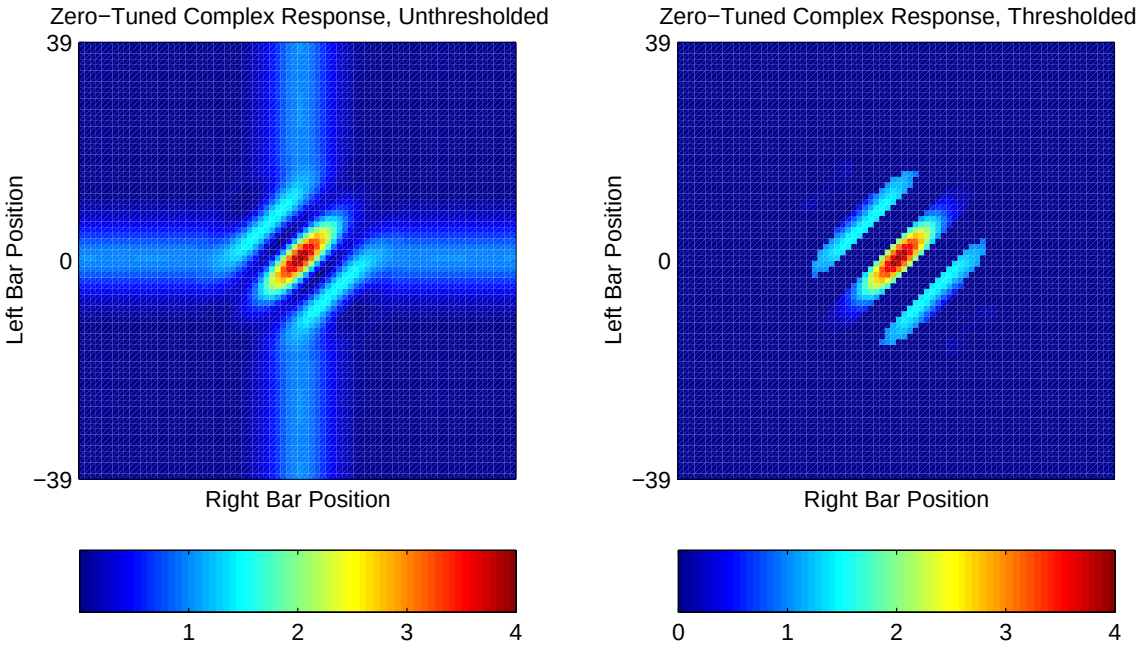



FIGURE 2.9. Complex cell tuning before and after thresholding. The wide vertical and horizontal bars in the left plot are the monocular responses which are removed in the right plot by thresholding the output of the complex cell.

which we have so far been dealing. The location of the filters must occupy positions in both the x and y dimensions of the image. All the experiments with real images carried out in this project used 2D Gabor filters but they were placed only along a horizontal line. This is because only a (real) vertical bar stimulus was used for most experiments and so while measurements in the y dimension for any one filter is performed, the overall disparity is measured only along the horizontal axis. The width-to-height ratio (of the sigmas) of the 2D Gabor functions was 2.

Another issue involves determining what portions of the image should be used for extracting the disparity measurement. In humans this area occupies approximately 6° in the field of view (Popple *et al.*, 1998). In this project the disparity-measurement area is limited by each spatial width; the range of pixels that any particular RF covered was limited so that the RF did not overstep the bounds of the image region. An earlier version of the software allowed for the center of the filters to extend to the edge of the image, allowing RF overlap and preserving of the output scaling, but that capability was removed to improve performance and to simplify the design. This is another reasons only a single horizontal line of cells was used for the majority of the experiments: if multiple disparities exist along the vertical axis then how should the system choose? In this case, it turns out that if the disparity-measurement of a 2D area of the image were carried out, the system would simply average them together.

By limiting the disparity cells to a horizontal line, it is easier to visualize the resulting surface plots across only an X and a Z dimension. A Y dimension would result in a 3D density plot, which is hard to visualize and serves no useful purpose in demonstrating how the system works. Perhaps if the system is used in a real robot with faster hardware, then a more realistic disparity-integration area can be created.

Finally, Marr and Poggio (1976) specify uniqueness as a desirable feature of a disparity-detecting system. That is, any point in the image should be assigned one disparity. It is known that humans have the ability to perceive multiple disparities at one location (Palmer, 1999; Qian, 1997; Grigo and Lappe, 1998), and although the system presented here does not explicitly prevent a multiple-disparity-representation from occurring at the filter-bank level, the vergence controller reduces this to a single disparity. In other words, there is no "intelligent" or neuromorphic scheme here that would allow the system to choose the disparity to which the eyes should converge. It has been shown that humans are sensitive to combinations of disparity and motion, wherein a

complex cell responds to a stimulus exhibiting a particular disparity at a particular spatial-temporal frequency (motion).

**Chapter 3**

# EXPERIMENTAL APPARATUS

Because the algorithms developed for vergence control and tracking depend on the physical properties of the experimental apparatus, we now present a description of the apparatus.

The physical oculomotor system used in this project has several components: the camera frame, the computer, the XY positioning table, and motor controllers. The frame which holds the two cameras and servos is a custom-designed-and-built gymbal-type aluminum structure (Fig. 3.1a). The frame holds each camera on a vertical axis with the servo motor adjacent. Thus, each camera has independent control of its azimuth (horizontal angle), but they share their elevation (vertical angle). Although an outer frame was built to control the cameras' elevation, only the inner frame was used throughout the project; control of the cameras' elevation was not addressed and therefore the outer frame was not used.

The two cameras (Fig. 3.1b) are MB-1250HRP color CCD board cameras from *Polaris Industries* (http://polarisusa.com/mb-1250.htm). They produce 470 lines of color NTSC video from a 1/4" (actually 3.6 x 2.8mm) interline transfer CCD at the normal 60 interlaced frames per second. The pinhole lens has a focal length of 5mm, resulting in a field of view of 35 degrees. The cameras were chosen based on their size, weight, resolution, and price. It was desired to have cameras with a small size and low enough mass so a large frame would not be required and so that an inexpensive servo motor could move it easily and quickly. The resolution and the fact that they produce color output were actually beyond the requirements of the system, but the price was low enough (approximately $200) that the extra flexibility was worth the cost over cameras with lower resolution and/or grayscale output.

The servo motors used to move the cameras are *Hobbico* Command CS-11 Micro Servos. Their maximum torque output is 30 oz·in (0.21 N·m) and their maximum unloaded speed is 400°/sec. The position of the servo is controlled by the duty cycle of a PWM signal.

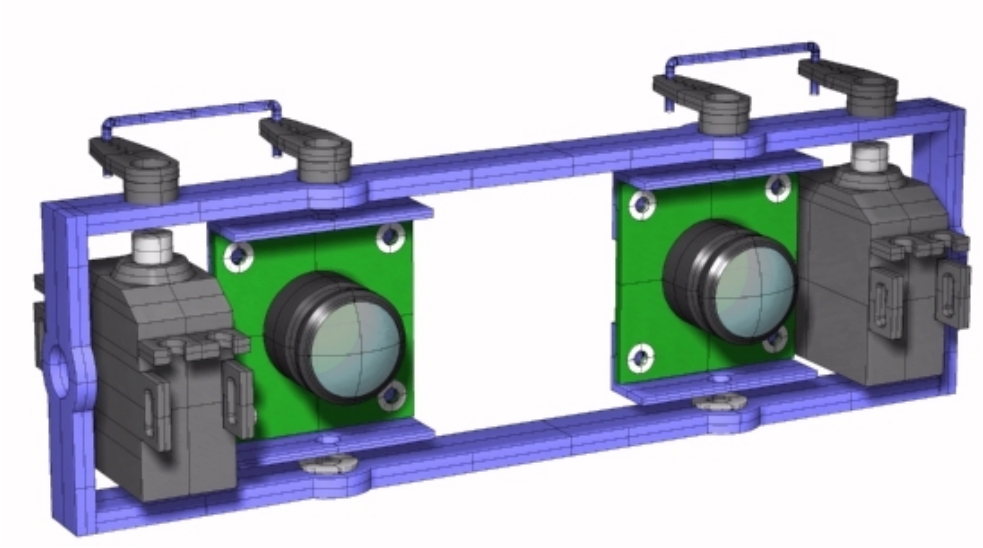
The computer generates two of these signals through a *Seetron* Mini-SSC (http://www.seetron.com/ssc.htm) servo controller (Fig. 3.1c). The computer runs all the image processing and control software. It is a Gateway Intel PIII-550MHz PC running the Microsoft Windows 98 operating system, equipped with 128MB of RAM and two *Imagenation* PXC200 video capture cards. The cards are capable of 640x480 24-bit color capture at 30 frames per second, with real-time display to the screen via a DirectDraw library function.

The XY table used for stimulus positioning is an *Arrick Robotics* model XY-18 (http://www.robotics.com), capable of 18 inches of range in both dimensions (Fig. 3.1d). It is driven by two generic size #23 stepper motors, which in turn are driven by a *Stepper Control* A-200 Stepper Motor Controller (Fig.3.1e) (http://www.steppercontrol.com). Figs. 3.2 and 3.3 show photographs of different views of the hardware setup.

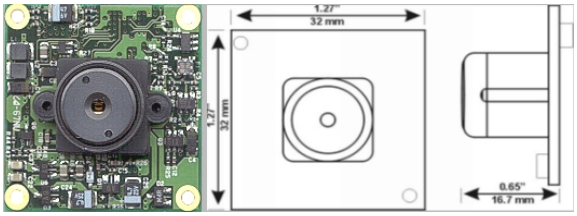
The control software was written in C++ with the Microsoft Developer Studio 6.0 environment. All the software elements reside in classes, arranged hierarchically both in encapsulation as well as in inheritance. This makes it relatively easy to instantiate multiple filters and cells to cover the range of tuning frequencies, disparities, and spatial locations. There are also software hooks into Matlab for ease of data manipulation and visualization. The stepper motor and servo controllers each provide dynamically loadable libraries for automated stimulus and camera positioning.

A GUI (Fig. 3.4) allows for run-time manual control of the camera positions and ideal-stimulus width, position, and disparity. The GUI does not have direct XY table control since the stepper motor controller has its own interface for manually adjusting the table. (In actuality, though, during development the power to the stepper motors was usually turned off and the table position moved manually, since this was more convenient and was much faster than the motors could move the table; stimulus position accuracy was not critical at this point. During the data-measuring phase, however, the software did control the table for repeatability and accuracy.) The GUI also shows the experimenter the two camera images in real-time and their superposition, which is useful to gauge subjectively the amount of disparity in the image. The software also has the capability to show the intermediate results of each filter if necessary, although this feature was included toward the beginning of the project for easy runtime verification of the system and is not used anymore because it consumes tremendous amounts of memory and slows the system down greatly.

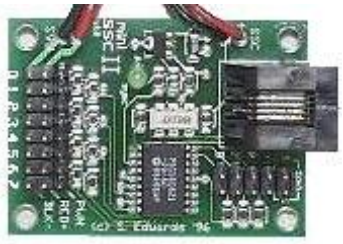
The entire application was written with wxWindows, an application framework which wraps the



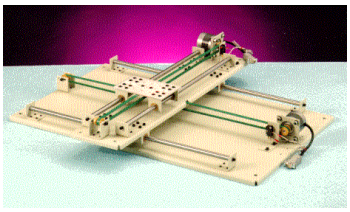
(a) Camera frame CAD model. Some changes were made between the design and construction, so there are discrepancies between this model and the photos.



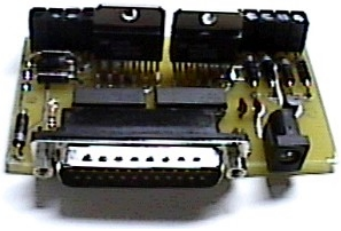

(b) MB-1250HRP high resolution color CCD camera.



(c) Seetron Mini-SSC serial servo controller.

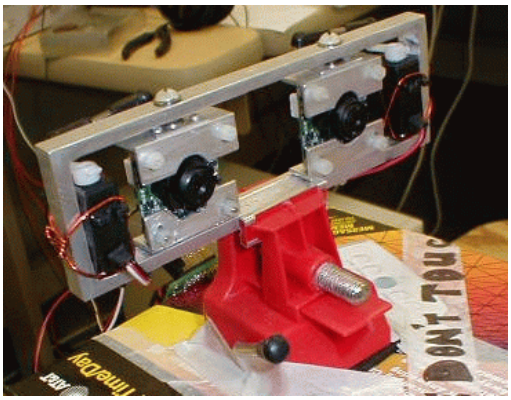


(d) Arrick Robotics XY-18 positioning table.

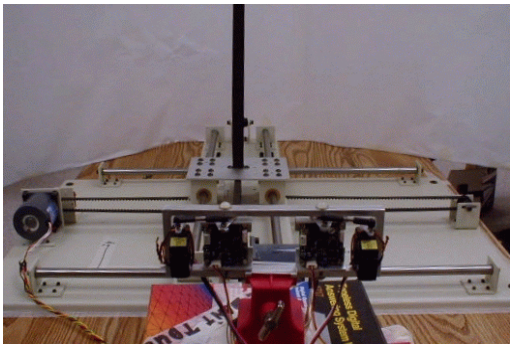


(e) Steppcontrol A-200 stepper motor controller.

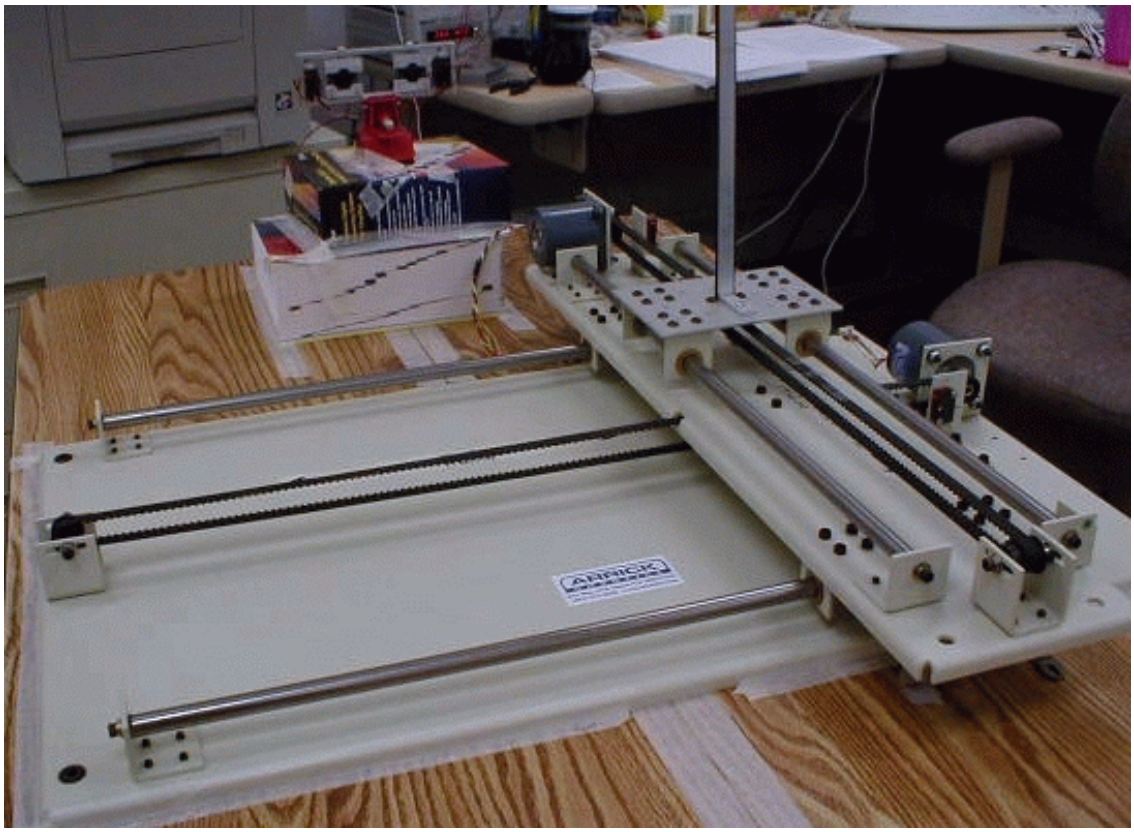
FIGURE 3.1. Small hardware: camera frame, camera, servo motor controller, XY table, and stepper motor controller.



(a) Closeup of camera frame.



(b) Centerline view of setup.



(c) Front view of setup.

FIGURE 3.2. Photographs of entire setup showing camera frame and its parts, XY table, stepper motors, servo motors, test stimulus, and background.

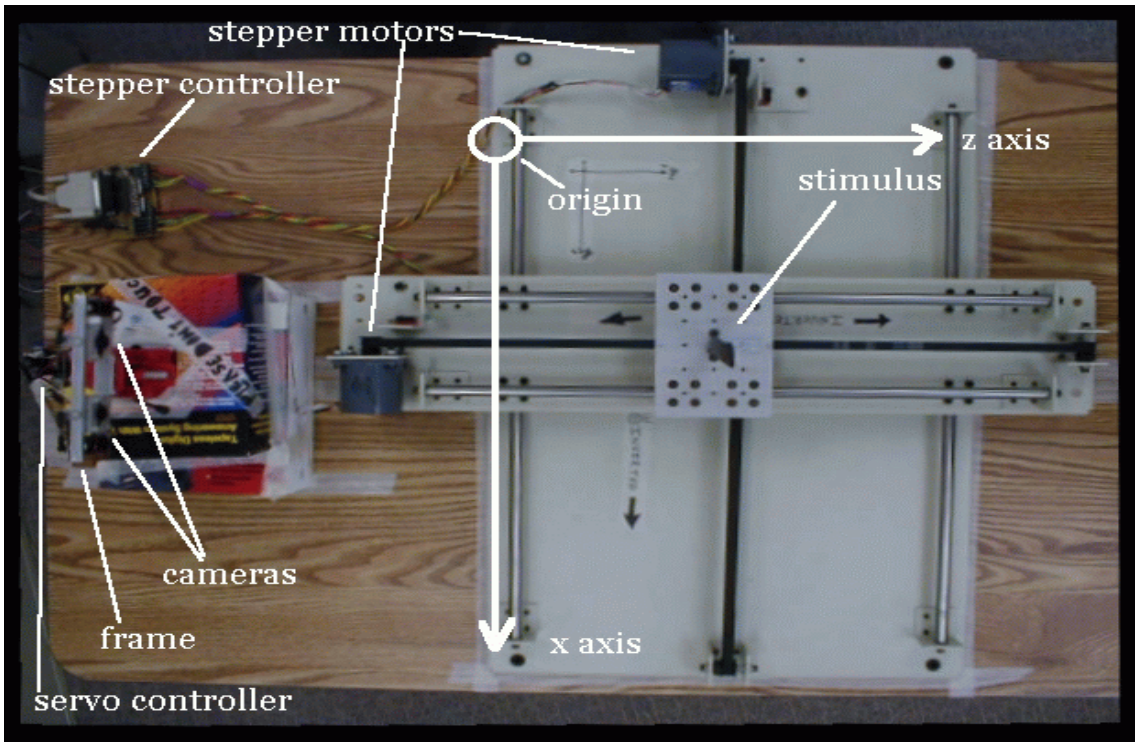



FIGURE 3.3. Photograph of setup, top view. Important components are indicated.

Microsoft Windows API into more manageable classes, similarly to MFC, but which also provides the potential for cross-platform compatibility. This provided easy creation of windows, sliders, callback functions, etc. Originally it was thought that this project might get ported to Linux, but it soon became apparent this would not be so; enough of the software is extremely platform dependent, such as the frame grabber, servo, and motor controller libraries, that the effort required to port the application is too great for any perceived benefit. In the unlikely event this project is ported to Linux, drivers/libraries are available for the video capture boards and the stepper motor controller, and the servo controller only requires a few bytes of serial data for each move. More detail of how the software works is not very useful at this point, so a separate document describes the software in detail for the interested reader and includes the full source code.

There are several sources of error in the apparatus. There are approximately 2-4 PWM units of slack when changing servo direction, which corresponds to $4 \cdot 90°/254 = 1.42$ degrees of error in the azimuth (horizontal angle), since there are 254 positions available over 90° of arc. There are also 1-2 degrees (estimated) of slack between the servo itself and the cameras and another degree or two of angular offset between the two cameras, due to unequal pushrod length and flexibility and loose connections in the servo control horns. In addition, the handmade camera control horns which convert pseudo-linear pushrod motion to angular camera motion are unequal, resulting in a few degrees/degree of cyclopean angular gain difference between the servos and the cameras. Combining all the known errors yields approximately 5° of error around the center cyclopean angle and a few more degrees at large cyclopean angles. These errors will have an effect on the estimation of absolute stimulus location. Finally, there is a small difference (perhaps a fraction of a degree, or one pixel at a scene distance of about 3-4 meters) in the elevation (vertical angle) of the two cameras, which does not affect this project since only horizontal disparity is considered.

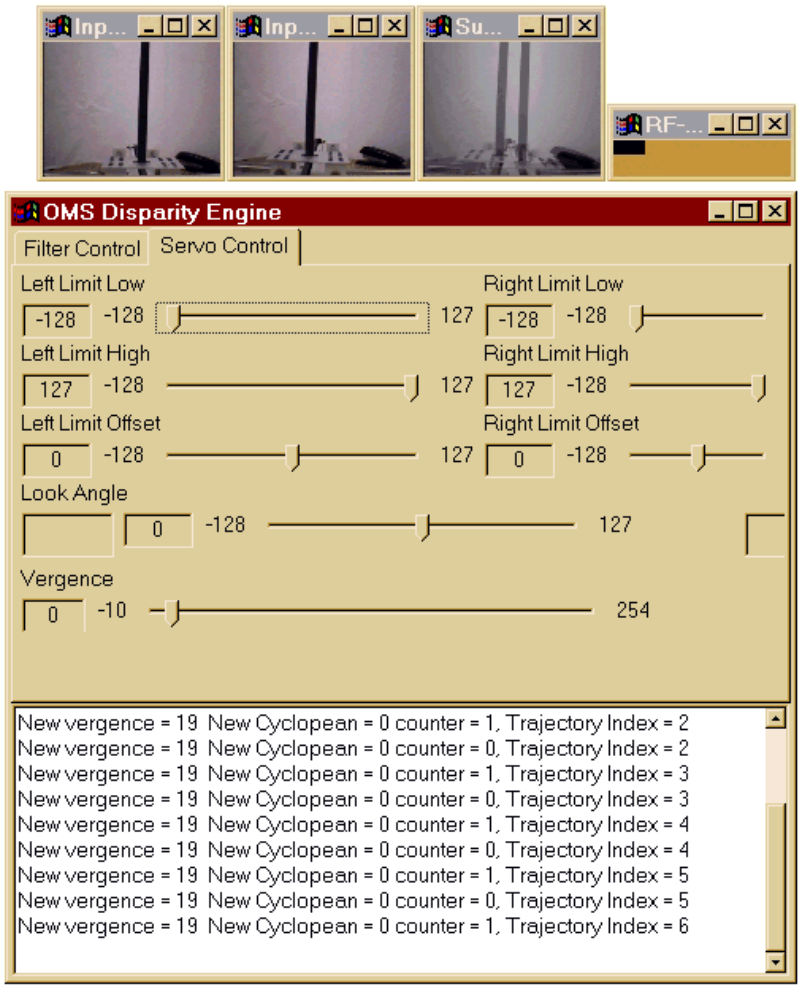



FIGURE 3.4. GUI screen capture. The top row of windows shows the images from the left and right cameras and their superposition. The cameras are *not* verged on the vertical bar stimulus. The small window in the top row shows the disparity-tuned filterbank. The main control panel shows sliders for controlling manually the position of the servos and their upper and lower limits and offsets (which were not used). The "Filter Control" tab was created during the beginning of the project when there was only one cell and it was feasible to experiment with its parameters interactively via the GUI

**Chapter 4**

# VERGENCE

*Vergence* is the disconjugate rotation of the eyes toward each other in order to center a region of interest on the retina (Fig. 4.1). This project rests on the notion that vergence control is possible through measures of disparity, which has been shown to be sufficient for biological systems (Mallot *et al.*, 1996; Stevenson *et al.*, 1999). Table 4.1 summarizes the population-decoding methods used here. Others have used disparity for vergence control (Marefat *et al.*, 1997b), but they measured the disparity directly (geometrically) by taking the arctangent of the ratio of the odd and even simple cell responses; they did not incorporate a complex cell to measure disparity energy. The vergence control algorithms here use three different population decoding methods (PDMs) available from the cells as described in the previous chapter. Since this project did not set out to *model* a primate visual system, but only to use a model of such as starting point for a robotic system, the control scheme presented here does not attempt to model any of the dynamics of a biological system, including the two-stage behavior or any of the control s-domain dynamics, etc. Since the servo motors used here are position-based, with their own built-in feedback and positioning control, it is nontrivial to measure things like angular velocity or position from, or force being applied to, the cameras. All moves from one angular position to another are performed at full speed, and there is no reason to model the motion of the eyes other than to claim a better understanding of how the natural mechanism works. Since the angular positions sent to the servos are discrete, it is impossible to impose a continuous type control loop on the system.

Previous vergence control models which have been studied for this project assume a disparity input, although they do not always provide a clear definition of what that input is or how it is derived. In Cova and Galiana (1995; 1994) they claim their control system is essentially a differential amplifier, with a common-mode (fast response) and a differential mode (slow response), with the inputs coming from a left and a right desired vergence angle. Hung (1998) also expects a vergence angle as the input to his controller. Patel *et al.* (1997) made a reference to disparity-tuned cells providing a winner-takes-all type of input to the vergence controller, but did not elaborate on how this was done. They do not assume a vergence-angle as the input, but instead rely on which disparity cell was giving the maximum response to determine the speed and direction of the vergence change. This scheme fits best with the system presented in this project, and is thus the starting point for some of the control methods presented below.

## 4.1 Using A Single Cell Hill Climbing Method For Vergence Control

The first use of disparity for vergence control was developed simply to illustrate that vergence control is possible using disparity energy from a complex cell as suggested by Ohzawa. This method consisted of using a single complex cell tuned for zero disparity at each spatial location, and the use of an 8-state finite state machine to find the maximum energy point. Fig. 4.2 shows a bubble diagram. The algorithm is an ad-hoc local-hill-climbing algorithm. The state machine scans the

| | Population Decoding Method (PDM) | Equation |
|---|---|---|
| 1 | Maximum | $\max(S(d))$ |
| 2 | Average index in sum-of-columns, weighted with energy | $\frac{\Sigma_{d\in D} S(d)\cdot d}{\Sigma_{d\in D} S(d)}$ |
| 3 | Sum of energy in sum-of-columns, weighted with index | $\Sigma_{d\in D} S(d)\cdot d$ |

TABLE 4.1. Summary of population decoding methods, or PDM[1-3]). $S(d)$ is the energy output of a disparity tuning $d$ along the sum-of-columns of a filter bank, $D$ is the set of filter tunings present at the sum-of-columns. PDM1 chooses the maximally-responding cell in the sum-of-columns of a filterbank. PDM2 finds the average index of the sum-of-columns, weighted by energy output. PDM3 reports total energy summed across the sum-of-columns; since the cell indices range from negative to positive, the sum of weighted energies could be zero, even though all the energies themselves may be greater than zero. These PDMs are applicable to both the global and local disparity estimates.

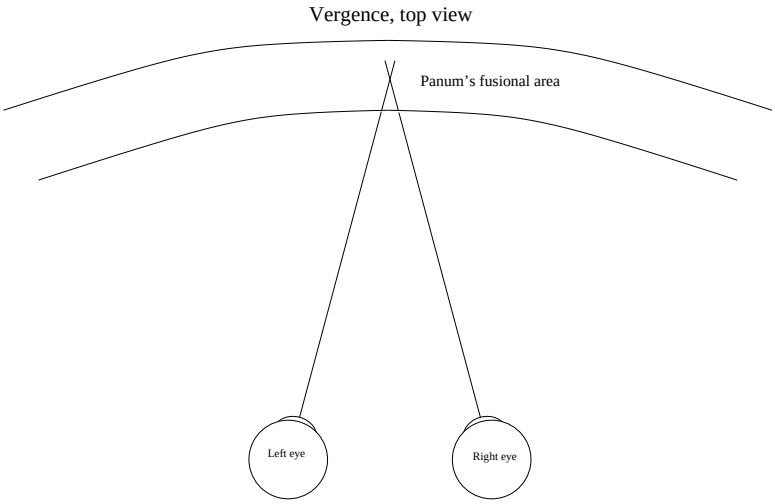



FIGURE 4.1. Vergence, top view. The eyes rotate until the target is centered in both retinas. Any remaining disparity can be used to perceive depth within Panum's fusional area.

range of vergence values and records the sum of the cells' output energies for each value of vergence. At each point in this global vergence scan, it compares the energy to the maximum known energy and marks the vergence location of the maximum energy so far. Then the vergence makes a discontinuous jump to the last known maximum location and performs a small local scan to resolve more finely the maximum point. If the energy then reduces past some threshold, the algorithm assumes the depth of the object has moved forward or back, and another local maximum search is performed, again using several thresholds to mark its place along the hill. If the object moves too far for the sample-and-update period (temporal aliasing) then the algorithm may get stuck on the side lobes since those present a local maximum. If the disparity then changes slowly, the system will continue to lock on to the sidelobe instead of the global maximum. The result is a 1D disparity tuning profile, as shown in Fig. 6.1 in Chapter 6. The pseudo-code representation of the state machine may be found in Appendix A of this document; Fig. 4.2 presents a bubble-diagram.

Part of the difficulty in developing the hill-climbing algorithm was that there is considerable hysteresis in the servo motors as explained in Chapter 3; a jump back to some vergence value is likely to undershoot the desired location, thus a guarantee of the distance from the hill maximum is limited by the hysteresis of the servos. If the thresholds in the algorithm are made too "tight", the algorithm will never converge, or will converge in a long time, because the vergence will always jump back and forth around the hill maximum and never reach it, or reach it only by a chance misstep someplace else which places the jump to the correct location far enough away that a single jump or two will actually reach it. The algorithm also does not always converge to the proper point after the initial sweep, and it certainly may get "lost" somewhere on the disparity energy profile if it makes an erroneous jump. Another problem is that the hill maximum recorded in a "scanning" state, such as states 4 and 5, may have been a noisy value and that value does not exist when the controller goes back to find it if the threshold is set too close to the measured value. Low pass filtering relieves this issue, but slows the system down. The height of the hysteretic windows used for this algorithm were therefore determined experimentally, and were generally found to be useful at 1/25th the height of the energy value they were surrounding. Another drawback to this method is that the energy from a single cell appears very sensitive to shadows in the image. It is sensitive to the point that the faint (unnoticeable by the experimenter) shadow of a person walking a few feet away from the apparatus may reduce the energy enough for the system to interpret that as a change in target object depth and to initiate the local hill climbing scan. Finally, another problem is that when the target object moves in depth and the output energy decreases, the system does not inherently know which direction the target has moved, either closer or farther. Thus, the hill climbing algorithm must pick a direction to test the energy slope and immediately change direction if it is incorrect. In this implementation the initial search direction used for a local-maximum scan is the direction of the last movement.

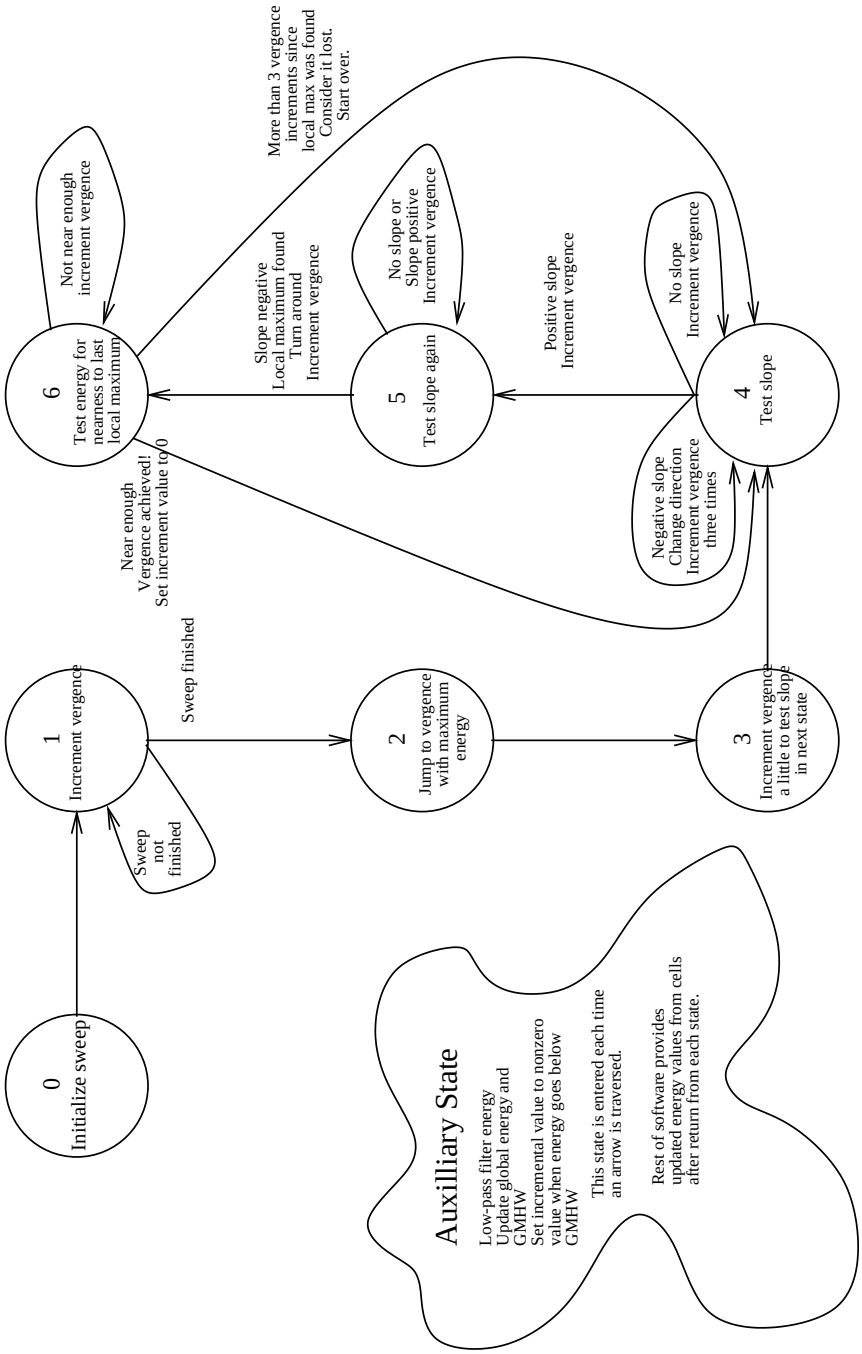



FIGURE 4.2. Bubble diagram showing state machine for primitive hill-climbing vergence control algorithm. The cameras were "swept" over their entire vergence range while total disparity energy was recorded. The cameras were then returned to the position of maximum energy and made to find the local maximum, which was assumed to be the global maximum.

## 4.2 Using A Disparity-Tuned Filter Bank for Vergence Control Via Global Disparity Estimate

The second use of disparity for vergence control uses the filter bank as a measure of global disparity. In Chapter 2, a filter bank was described as existing for each point in the image, thereby giving an RF-tuning estimate for each point in the image. By summing the results of the individual location-specific maps we can derive the global disparity estimate. The result of the sums is an RF-tuning map that returns the frequency content and disparities of the image averaged over space. This map can be used with the three PDMs to extract a single value used for vergence control.

PDM1 simply gives us which cell in the sum-of-columns is firing the most. PDM2 gives us the centroid of all the cells, depending on how much they are firing. PDM3 gives us the product of energy and the indices from where it came; e.g., a lot of energy near zero disparity or little energy anywhere will return a low value, whereas only a lot of energy at a distant non-zero disparity will result in a significant return value. PDM3 was developed so that a scene with little or noisy information would result in little response, rather than *always* giving *some* response, regardless of scene content, driving the controller to some erroneous value. The RF-tuning map can also return the energy sum of all the cells.

Since this use of disparity information removes any spatial specificity of the source(s) of the disparities, the system does not know where the stimulus is in its field. This method can be considered "cell-centric" in that each cell in the resulting global RF-tune map is the sum of the same homologous cells in all other spatial locations, thus removing spatial specificity. The particular software implementation of the cells made this the next easiest interpretation to implement after the first use of disparity with the state machine. Since the spatial location of the stimulus is removed, the disparity energy cannot be used to keep the target object centered in the field (horizontal tracking), one of the stated goals of this project.

## 4.3 Using A Disparity-Tuned Filter Bank for Vergence Control Via Local Disparity Estimates

The third use of disparity for vergence control is similar to the second, but it preserves spatial specificity of the source of the disparity. This third method can be considered "pixel-centric" because all energies from all RF widths for a particular disparity are summed for each spatial point. The easiest way to visualize this is in the x-d (x location vs. disparity) space, which is the model assumed for the following discussion. Fig. 4.3 shows a visualization of x-d space with a real vertical bar stimulus. For each coordinate in the x-d space, the energies from all cell widths are summed. This way, the disparity energy at each location is measured. If there are multiple disparities for any one location, this method should theoretically be able to represent that situation.

The x-d space can be expanded to x-y-d space if required, simply by adding cells along the y axis of the field, however this causes difficulty in visualization since it would require a true 3D display, and it would increase the complexity of the system to include some way of interpreting disparity values along the y-axis in a reasonable manner. See Marr and Poggio (1976) for a set of requirements a disparity measurement system should have, but also see Marshall *et al.* (1996) for a nicely working system that violates those requirements.

The three PDM outputs and the total energy can be derived from the x-d space similarly to the cell-centric method. It turns out that using the maximum cell (PDM1) in x-d space combined with the total energy of all the cells in a *complex controller* (below) produces the best vergence control, and using the average (centroid) of the cells (PDM2) in x-d space resolves the remaining disparity and horizontal position into real Cartesian coordinates.

The vergence signal obtained via cell-centric or pixel-centric methods is then fed to either a "simple" or a "complex" vergence controller described below. Recall that the disparity estimate is only a *relative* difference between the horopter and the target object; the disparity estimate cannot give an absolute angle for vergence. However, it is also desirable that the vergence and horizontal controllers return the eyes to some neutral position in the absence of any valid data, not only to mimic biological behavior, but also to effect a "search" for a valid vergence angle if the controller gets stuck in a sidelobe.

## 4.4 Simple Controller

The simple controller shown in Fig. 4.4A is basically an integrator. It takes as input a disparity estimate which it converts to a change in the vergence angle by some scaling factor determined experimentally. It then adds this delta angle to the current vergence angle and moves the cameras. This allows the vergence angle to stabilize when the disparity estimate is zero, and to accumulate

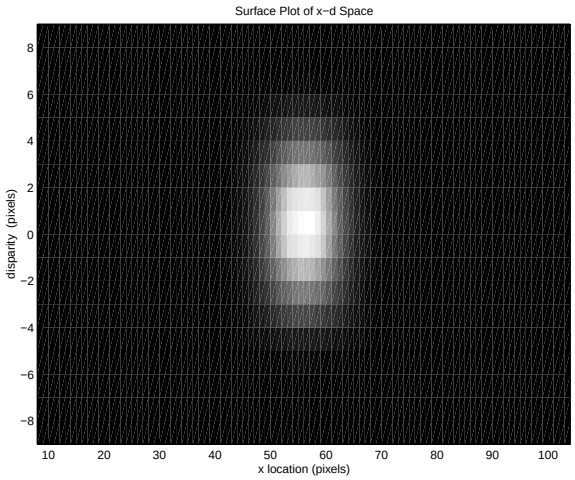



FIGURE 4.3. Example of x-d space. The aggregation of energy shows the location of the stimulus, both in horizontal (x) space and in depth (disparity). This plot was taken from a real vertical bar stimulus.

small integral disparity amounts which would individually not cause much change due to slack in the servo motors, etc. At one point it was thought that the PDM3 value could be used to "pull" the vergence away from some neutral position, opposed by a "spring", so that when there was little energy, the system would return to its neutral position. Unfortunately, when the disparity did get corrected to zero by a vergence move, the cameras jumped back to their neutral position since the energy with zero disparity is zero, and the system would oscillate. This could be interpreted as there being too much gain for the sampling rate, or as a large gain error in a difference-amplifier (due to too low a gain, if the system were continuous). It was also determined that using the energy this way in effect removed the assumption that the disparity was only a relative difference between the horopter and the target, and instead wound up interpreting the disparity as an absolute angular position. As this was discovered fairly late in the project, the PDM3 value remains in the block diagram, although its use is deprecated.

Another concern is that the PDM2 value would often yield results between 0 and 1, but the software servo driver only accepts integral values. (The integral positions of the servos themselves combine to serve as the integration variable). This means that small nonzero disparity values could not be used to turn the cameras, as their values would not accumulate over time (with each cycle the error would get truncated to zero), leaving a small but constant error. More importantly, however, a small nonzero disparity value could also be obtained by a disparity estimate with a lot of side-lobe activity, i.e., a large disparity (and hence large sidelobe) can result in an average disparity close to zero. If the sidelobes are approximately the same size as the main lobe, then the average can easily be near zero, but still less than 1, thus exacerbating the detrimental effect of sidelobe ambiguity by failing to pull the cameras in the correct direction. The effect is that the system is more likely to stabilize ("lock") in between the main lobe and the sidelobe, which is a much more serious problem than a small error near zero disparity which cannot be accumulated. It was decided that only the maximally-firing cell (PDM1) should be used as input to the vergence controller, since that does not compromise the range of disparities as much as the average does. The average (PDM2) can be used, however, once the vergence has stabilized, to estimate the true disparity and therefore the depth around the horopter, since at this point the sidelobes are assumed to be minimal or nonexistent.

## 4.5 Complex Controller

The major problem with the simple controller (Fig. 4.4A) is that unless the disparity estimate is zero, regardless of the amount of energy in the system, it always wants to move the cameras *somewhere*. In the case of a scene with a low signal-to-noise ratio, the maximum point of disparity could conceivably jump around, causing the eyes to move discontinuously with no apparent motive, and if the eyes get stuck in some sidelobe (still possible, though the chances are reduced with the use of maximally-firing cells rather than average cells), then attempts to correct them by moving

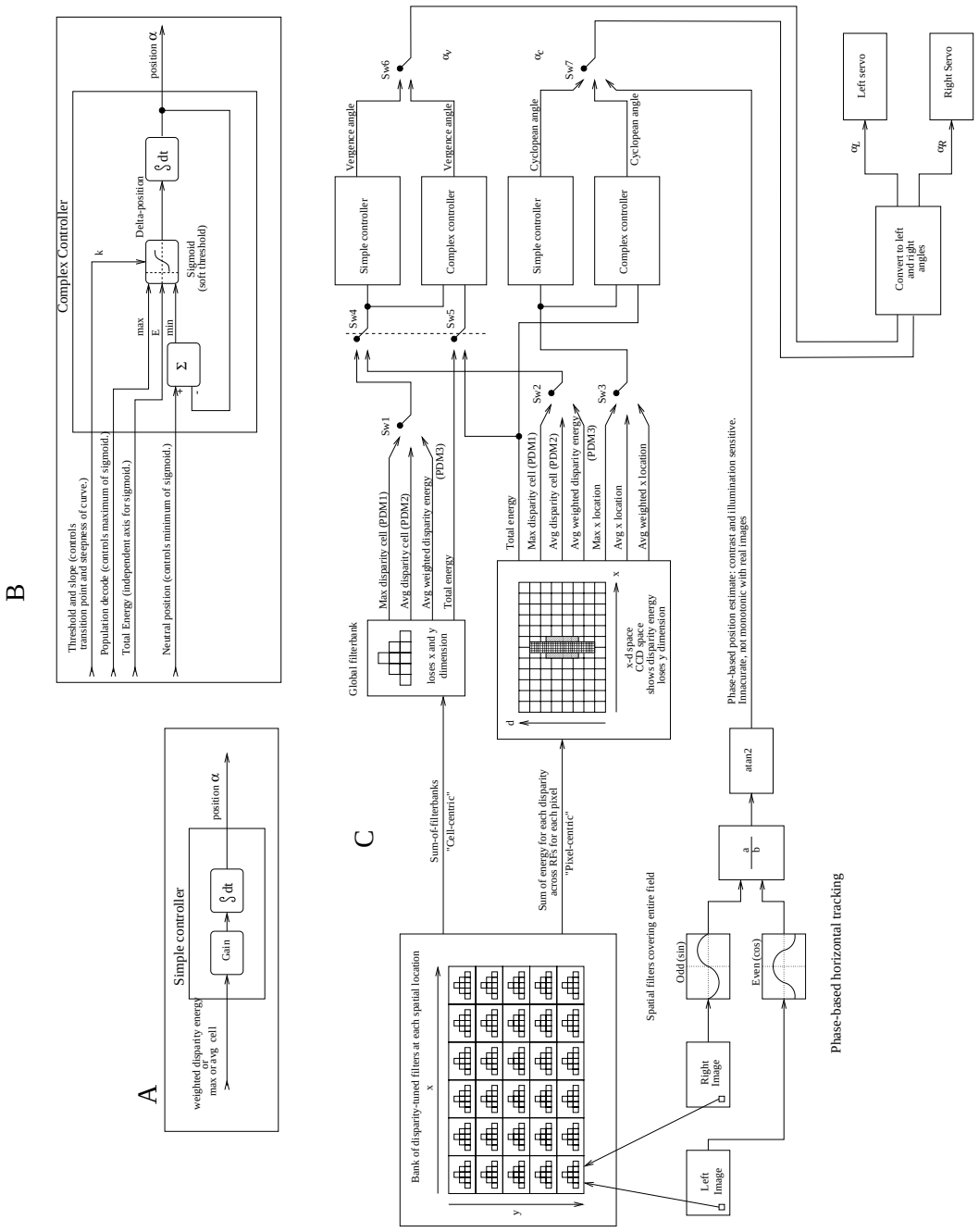



FIGURE 4.4. System-level block diagram showing the two controllers for the three vergence and two tracking methods. Region A Shows the simple controller. Region B Shows the complex controller. Region C shows how the disparity-tuned filterbanks cover the visual field and are used as inputs to the simple or complex controllers. It also shows towards the bottom a phase-based horizontal tracking mechanism. The outputs of the system are angles for the left and right servo motors.

the stimulus to an extreme position can cause the error to grow, causing a significant (if not stupid-looking) amount of cross-eyes or its opposite, "goat-eyes".

Similarly for the tracking, if the control relies only on relative maximum locations, the perceived target could jump around, or the shading gradient of the background could be interpreted as data, causing the eyes to move to the extreme left or right. There ought to be some way of qualifying disparity with the overall amount of energy in the system, which allows for the eyes to move back to a neutral position when there is not enough energy, and to move to the appropriate vergence value when there is enough energy. The value from PDM3 (total energy weighted by disparity) was a first attempt at solving this problem, but it has been shown in the last subsection why it does not work.

Biology has demonstrated its nonlinear nature, for example, in the sigmoidal soft-threshold function of a generic neuron. By using a soft threshold function on the total energy of all the cells and combining it with the index of the maximally-firing cell and the desired neutral position, a delta-vergence can be given to the servos so that strong zero-disparity response results in the eyes remaining where they are, but a weak zero disparity (or weak any-disparity for that matter) response will result in the eyes moving back to the neutral position. This move to a neutral position is quite effective in recapturing the disparity-lock when the eyes are verged near and the target is moved away suddenly beyond the point of maximum disparity so that big sidelobes are created. Because this new stimulus creates less overall energy, the thresholding function does not fire, and the controller goes back to its neutral position. We set the neutral position and the sigmoid function to return the eyes to a reasonable location where the stimulus might be. In this case, a vergence value of 20 (servo units) causes the eyes to verge to a location approximately 24 inches away, which is the edge of the X-Y table on which the stimulus is placed (Chapter 3), and which allows a large range of remaining disparities to be resolved by the largest-RF cells in the filter bank.

The following is the sigmoid function used in the complex controller:

$$f(E) = \frac{max - min}{1 + e^{-k(E-E_0)}} + min \tag{4.1}$$

where $max$ is the index of the maximally firing cell, $min$ is the difference between the current vergence angle and the neutral vergence angle, k is the sharpness or steepness of the threshold, $E$ is the total energy of the cells, and $E_0$ is the center-point of the slope, i.e., the threshold itself. $k$ is set to yield a "smooth" curve which almost reaches the extremes over the domain of the function. $E_0$ and therefore $k$ are determined experimentally, and vary with the number of cells used. These variables are represented in the block diagram (Fig. 4.4B) by expanded signal names. The numerator determines the range of the function, as the function asymptotically approaches horizontal with +/- range. The term added on the end recenters the function vertically so that when the energy is great, the index of the maximally firing cell is used, and when the energy is weak, the difference between where it is now and where it wants to go (neutral position) is given to the controller. Fig. 4.5 shows the sigmoid function.

## 4.6 System Integration

Figs. 4.4A and 4.4B show the system as described so far. The simple and complex controllers are shown at the top. The simple controller is a straightforward combination of a gain element and an integrator, which shows the cumulative nature of the vergence algorithm. The complex controller works as follows: the maximum of the sigmoid is given by the maximally firing cell, or other population-decoded quantity indicating a direction and/or magnitude to move (the average disparity and average energy was discussed earlier and it was determined these quantities are not as useful as simply using the maximally firing cell). The minimum of the sigmoid is given by the difference between where the vergence angle is and the neutral position. In other words, since this is a delta-based controller, the difference is the change required to move the vergence to the neutral position. The midpoint (threshold) and the slope of the sigmoid are determined experimentally and are dependent on the number of cells in the system. The output of the sigmoid is determined finally by the total energy coming in to it, either from the cell-centric or pixel-centric disparity estimation. The output of the sigmoid is given to an integrator as the amount by which the vergence angle needs to change.

The block at the left of Fig. 4.4C represents the bank of disparity-tuned filters for each spatial position. These filters are summed together across spatial position to yield the cell-centric (global) disparity estimate, represented by the single-pyramidal diagram, or they are summed at each pixel individually, and yield the pixel-centric disparity estimate, represented by the x-d space grid. The cell-centric and pixel-centric estimates both produce the disparity index of the maximally-firing cell

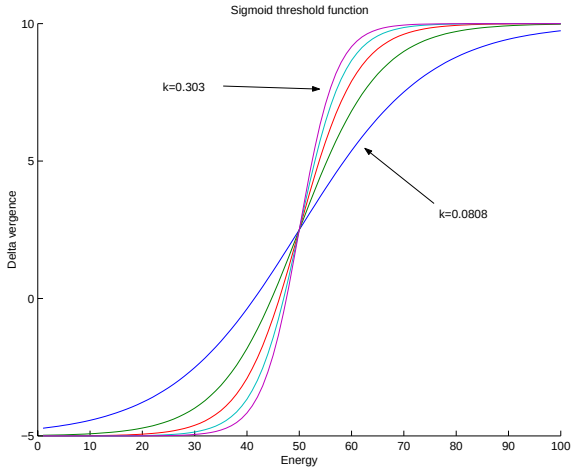



FIGURE 4.5. Sigmoid function with varying $k$ values. A *max* value of 10 and a *min* value of -5 were used. The domain is 100 and the threshold was set to the middle of the domain.

(or the index of the pixel with the most energy in the case of pixel-centric), the average index of the cells, and the average energy weighted by index of all the cells. They also produce the total energy of the cells. The switches represent the options available at the software level for how to combine the different values. The software does not implement any switches *per se*, but different sections of code can be commented in or out and the code recompiled for the same effect. Finally, either the simple or the complex controller is chosen for the vergence control. At the bottom of the diagram are phase-sensitive filters similar to those used in Lu and Shi (2000), which can be used for horizontal tracking (to be discussed in Chapter 5). Finally the vergence and *cyclopean angles* (the average conjugate angle of the eyes, i.e., the "look angle", where the eyes are pointed horizontally) are combined to produce left and right servo commands to control the cameras.

**Chapter 5**

# HORIZONTAL TRACKING

This chapter discusses two forms of horizontal tracking (controlling the cyclopean angle of the eyes) and the conversion of pixel data to real-world Cartesian coordinates.

## 5.1 Phase-Based Horizontal Tracking

Of the two forms of horizontal tracking, one gives vastly better performance than the other. The first method is similar to how Hansen and Sommer (1996) and Marefat *et al.* (1997b) used a conjugate pair of simple cells to estimate disparity and how Lu and Shi (2000) performed both tracking and vergence control. It consists of a conjugate pair of simple cells for each image which RF covers the entire field of view and which sinusoids take one complete cycle to cover the image width (lower part of Fig 4.4C). By taking the average of the arctangent of the ratio of the responses of the odd and even cells for left and right images, an angular measurement is computed, which is interpreted as belonging in the image width of $2\pi$ radians. Equation 5.1 shows this relationship.

$$p = \frac{W_{image} \cdot \left[\tan^{-1}\left(\frac{I_l(x) \star \sin \omega x}{I_l(x) \star \cos \omega x}\right) + \tan^{-1}\left(\frac{I_r(x) \star \sin \omega x}{I_r(x) \star \cos \omega x}\right)\right]}{4\pi} \tag{5.1}$$

where $p$ is the horizontal location estimate of the stimulus in pixels, $W_{image}$ is the width of the image in pixels, and $\star$ is the convolution operator. The $4\pi$ in the denominator accounts for averaging the left and right position estimates as well as converting the resulting phase from the arctangent into usable pixels; $\omega$ is chosen so exactly one cycle fits within $W_{image}$.

This estimate yields the average location of positive-contrast pixels in the image. This method is contrast dependent (it shows positive direction for white-on-black stimuli and negative direction for black-on-white stimuli), which is clearly not useful in a real-world robotics application. The value of this location is presented to the tracking controller as shown in Fig. 4.4C.

In experiments with a software-generated ideal stimulus (a vertical bar a pixel or two wide), the system worked fine; it tracked the vertical bar from left to right perfectly. The system performed horribly, however, when viewing real images: the perceived x-location of the bar when the eyes were stationary was not monotonic from left to right as expected, and when the eyes were allowed to move, they stabilized with the stimulus considerably to the left of the image, due to shadow-induced intensity gradients present in the plain-white background. The system performed a little better when only delta-position changes were used to control the tracking, as in Lu and Shi, rather than absolute position, but the performance was still not acceptable. Clearly another horizontal tracking method was needed.

## 5.2 Tracking From x-d Space

The vergence control methods presented in Chapter 4 can be used directly to achieve tracking as well. Tracking is achieved by taking the x location in the x-d space of the pixel with the most energy and running it through either the simple or the complex controller; the complex controller has the added advantage (as in vergence) of allowing the eyes to return to some neutral position in the absence of sufficient energy. Thus the vergence controller centers the stimulus in d-space, while the horizontal controller centers the stimulus in x-space. The energy centroid in x-d space after vergence and tracking have stabilized is used to estimate the exact position of the target and to account for regions in the field or filterbank where cells may not have been instantiated.

Chapter 6

# EXPERIMENTS, MEASUREMENTS, AND RESULTS

To test the system we ran some experiments to determine its range, accuracy, and fault-condition behavior (Table 6.1). The stimulus was a long vertical black bar approximately one inch wide, covering the vertical field of view, placed on the XY table (Chapter 3). A plain white background covering the entire field of view was set behind the stimulus to minimize noise. The lighting was standard office fluorescent overhead lighting. The first experiment was run once to demonstrate the feasibility of using disparity energy for vergence control. The second, third, and fourth experiments were run 10 times while the disparity-tuned-filter banks were used in pixel-centric mode and the resulting data were fed into the complex controllers.

## 6.1 Experiment 1

The first experiment (Fig. 6.1) tested the vergence capabilities of the state-machine tracker. Since this was meant only as a proof-of-concept control algorithm, not as many tests were run with this as with the disparity-tuned filter bank methods. The stimulus was centered in the XY table and the cameras were centered in their cyclopean angle. The system was started and the cameras were allowed to do their initial sweep to find the target location via disparity energy maximization. After the sweep the cameras returned to the position where they recorded the most disparity energy and performed a local maximum search. When it found the local maximum it marked this as the global maximum and waited for the energy to fall below a threshold before commencing the local-maximum search again, which in this case did not happen because the stimulus was not moved. The experiment was observed subjectively by this author and at the end of the experiment the superimposed images displayed no noticeable disparity. Thus, the error in the plot between either of the two global maxima and the computed correct vergence angle is due to the imprecision of the camera platform as discussed in Chapter 3; the error is within the tolerance estimated there.

## 6.2 Experiment 2

The second experiment tested the open-loop position-estimation accuracy by maintaining the cameras verged toward the middle of the setup table while the stimulus was moved across a range of X and Z positions in a square-wave pattern: 16 cycles of forward-sideways-backward-sideways, each point spaced $\frac{1}{2}$ inch apart. Figs. 6.2 and 6.3 shows the average X and Z position estimates, their standard deviations, and known stimulus positions versus position index. The disparity and x pixel locations were taken from the centroid of the x-d space, not the maximum point. The X axis follows rather closely in the middle of the plot where the stimulus was in the field of view. The Z axis follows only for small portions of each cycle, where the disparity was within range of the largest filterbank cells. The effects of phase-aliasing are shown in the middle of each cycle, where the Z estimate jumps from a somewhat correct value at one end of the tuning range toward the other end of the tuning range.

Fig. 6.4 shows the same data, but instead shows it as an error plot over the Cartesian space. The curved dark area of the plot is where the error was minimum. The minimum point in the whole plot is 2.10 mm; the maximum is 301.85 mm. Since the center of this region is closer to the cameras than the center of the test area, which is where the cameras were verged, this probably indicates a consistent offset in the angle estimate of the servos positions. This area fits within a v-shaped region which indicates the system's field of view over depth. This plot does not show the phase aliasing explicitly, but the transition between the dark region and the more errorsome regions appears nonlinear, reflecting the discontinuous jump of phase aliasing. The standard deviation over the 10 experiments ranged from a minimum of 0.17 mm near the center of the table to a maximum of 6.97 mm toward the rear left edge; the data were very consistent across the 10 experiments.

The parameters for this and the remaining experiments were as follows: cell widths were 27, 48, 69 pixels, each with 7 disparity tunings equally spaced across the width of the filterbank row. The cells were spaced 4 pixels apart, starting in the center of the visual field and extending to either side until the cell's RF met the edge of the visual field, yielding a total of 410 cells. Bear in mind each cell consisted of 4 convolution filters, for a total of 1640 filter computations per sample. The visual field was 112 pixels wide by 72 pixels high.

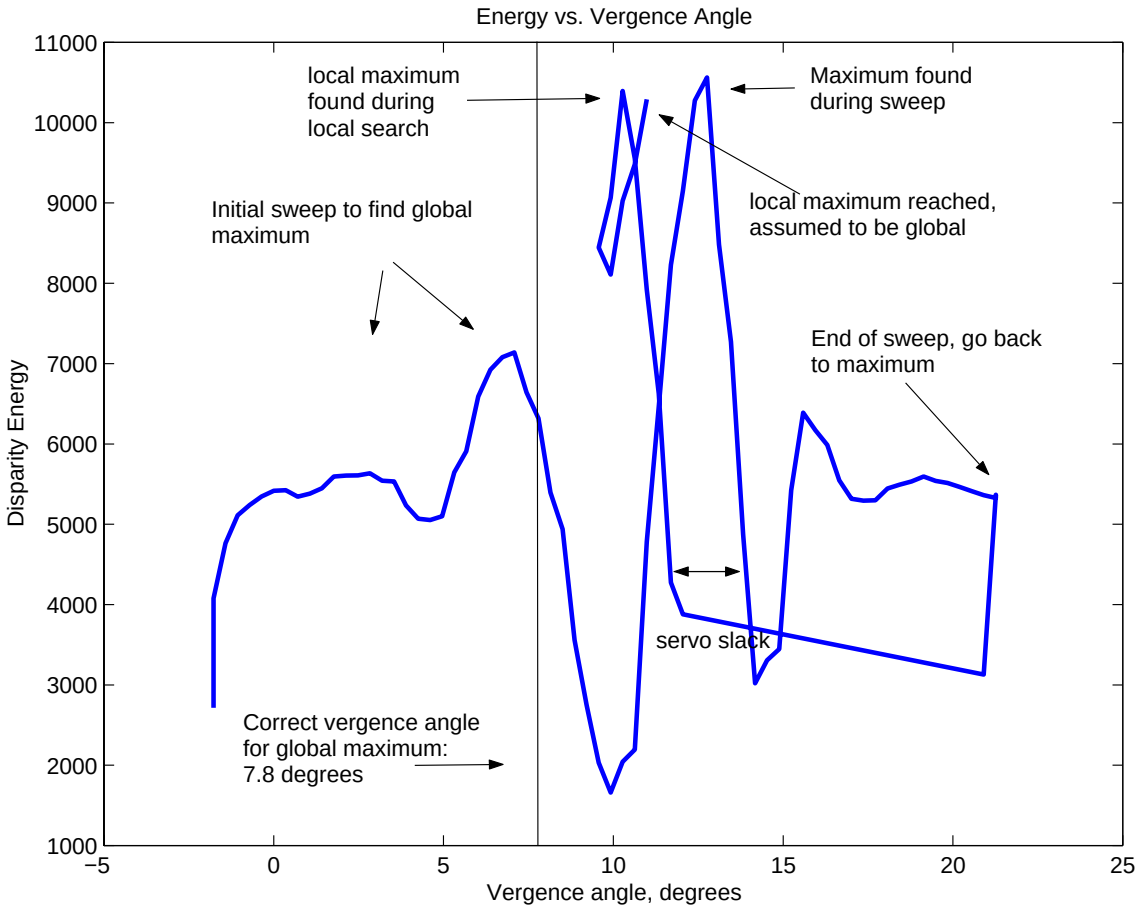



FIGURE 6.1. Experiment 1 result: Using a state machine with a single cell tuned for zero disparity, located at many spatial locations, to control vergence. The plot shows the sweep from the minimum vergence angle to the maximum vergence angle, with the energy plot matching qualitatively the disparity-tuned profiles presented earlier. At the end of the sweep on the right the system returns to a previous location to perform a local-maximum search. Once it has reached the maximum within some window then it stops and assumes it has found the global energy maximum. The correct vergence angle was computed from the known location of the stimulus (center of the XY table) and is the theoretical angle to which the cameras should verge, based on the CCD-to-Cartesian equations discussed in Appendix B. The error is within the tolerance estimated in the hardware discussion.

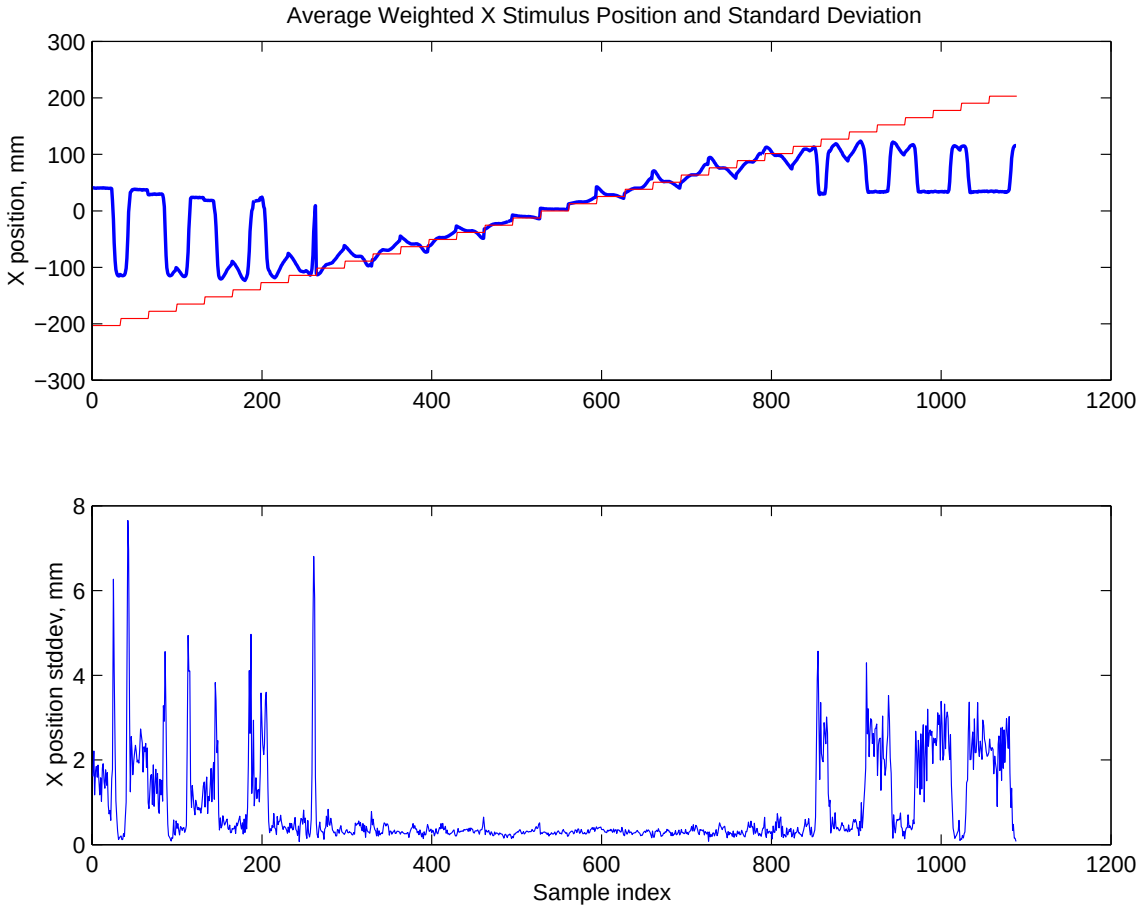



FIGURE 6.2. Experiment 2 results: X position estimate versus index. The cameras were verged and centered to the middle of the table while the stimulus was moved through a grid of points $\frac{1}{2}$ inch apart. This plot shows the average over all the runs of the X estimates (thick lines) and known stimulus positions (thin lines) versus position index. The system estimates the X position closely for a region in the center of the field of view (the X location increased monotonically over all the points). The second plot shows the standard deviation of these measurements, indicating a very consistent measurement in the center of the field of view. The standard deviation is shown on a separate plot for this reason – it is not visible when plotted at the same scale.

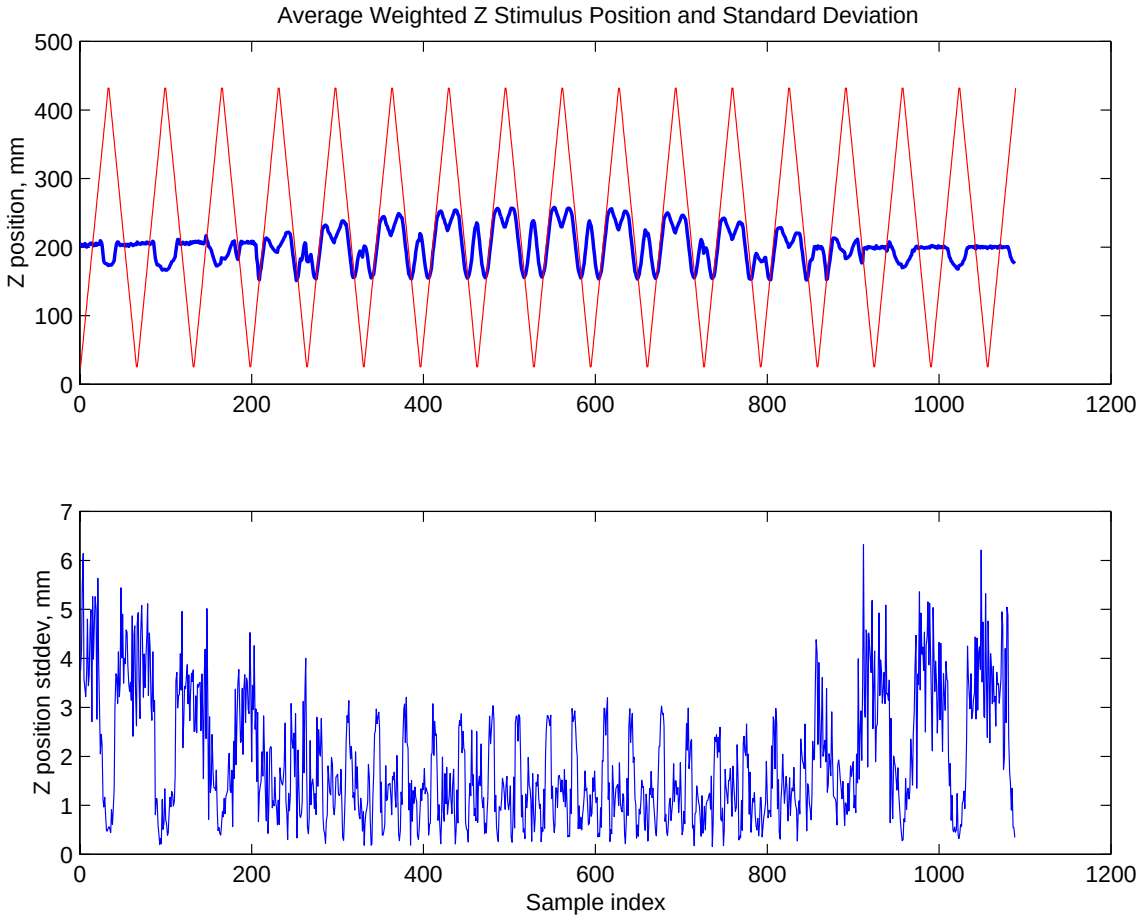



FIGURE 6.3. Experiment 2 results: Z position estimate versus index. The cameras were verged and centered to the middle of the table while the stimulus was moved through a grid of points $\frac{1}{2}$ inch apart. The top plot shows the average over all the runs of the Z estimates (thick lines) and known stimulus positions (thin lines) versus position index. The system estimates the Z position closely for some regions in the center of the field of view, where the disparity was within range of the cells. The second plot shows the standard deviation of these measurements, indicating a very consistent measurement in the center of the field of view. The standard deviation is shown on a separate plot for this reason – it is not visible when plotted at the same scale.

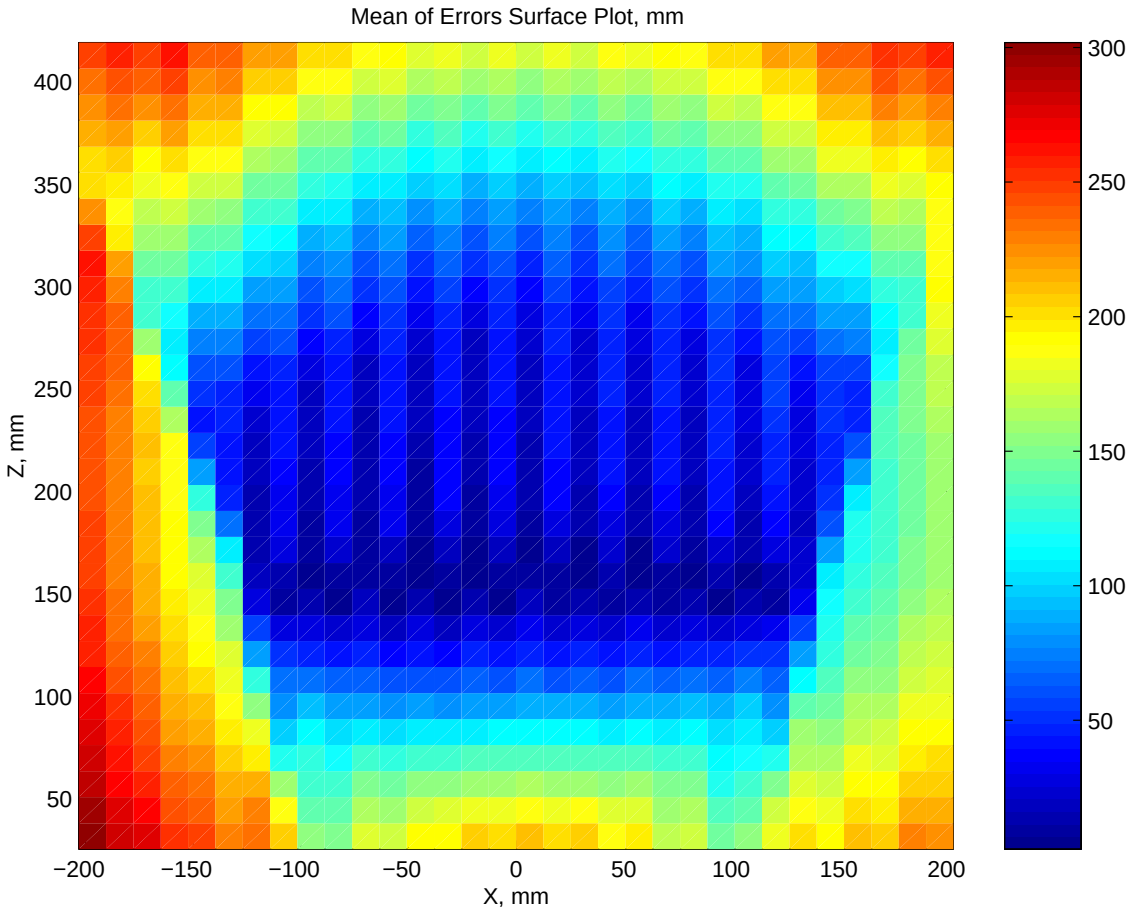



FIGURE 6.4. Experiment 2 results: Stimulus location estimate error versus position. The cameras were verged and centered to the middle of the table while the stimulus was moved through a grid of points $\frac{1}{2}$ inch apart. This shows a surface plot of the mean estimate error across the Cartesian plane across all 10 iterations of the experiment. The dark region in the center shows the best estimate (least error) and reveals an offset in the camera position estimate. The minimum point in the plot is 2.10 mm, the maximum point 301.85 mm.

| Number | Experiment name | Basic Operation | Camera Control |
|---|---|---|---|
| 1 | Energy maximize | single cell hill climbing | state mach. verg. only |
| 2 | Estimation accuracy | pixel-centric filterbank<br>open-loop estimation | not controlled |
| 3 | Peripheral vision | pixel-centric filterbank<br>closed-loop control<br>open-loop residual est.<br>eyes centered each cycle | complex control of<br>vergence and cycl. |
| 4 | Tracking accuracy | pixel-centric filterbank<br>closed-loop control<br>open-loop residual est. | complex control of<br>vergence and cycl. |

TABLE 6.1. Table of experiments. This enumerates the experiments used to test both the disparity estimation and the disparity-driven tracking capability of the system. Experiment 1 maximized the energy produced by one cell at each location via an ad-hoc hill-climbing method. Experiment 2 tested the estimation accuracy of the x-d space representation. Experiment 3 tested the step-response and peripheral range of the system using closed loop x-d space control and residual open-loop estimation. Experiment 4 tested the closed-loop tracking accuracy combined with the open-loop estimation accuracy of the x-d space representation.

## 6.3 Experiment 3

The third experiment tested the system's step response and range of sensitivity ("peripheral vision"). For each point in the square-wave pattern, the cameras were initialized to point at the middle of the setup table. Then they were allowed to verge and track to whatever stimulus they saw. Figs. 6.5 and 6.6 show the average X and Z position estimates, their standard deviations. and known stimulus positions versus position index, calculated identically to the data of Experiment 2. It is clear from these figures that allowing the cameras to move resulted in a much wider range of low-error stimulus position estimates, compared to those of Experiment 2, although the errors themselves seemed larger (Fig. 6.7). In areas near the cameras the errors seem excessive. Upon closer inspection, the large depth errors (the discontinuous spikes) occur mostly near the cameras, where the disparity was beyond the limits of the system, and the system found some sidelobe on which to lock. The non-spiked depth errors seem to occur far away from the cameras in the right half of the field. The reasons for this are not clear, since the error seems to grow continuously toward that region, indicating the system "knows" in general where the stimulus is (i.e., is not stuck on a sidelobe), but is not able to report its position correctly, probably because of some error in the mechanical apparatus. It appears also that the largest errors were accompanied by the largest standard deviations, further implicating the nonreliability of the mechanical system to resolve fine movements, especially for far target distances where small movements result in large changes in the estimated position.

Other regions of large error occur when there is very little energy in the system (bottom subplots of Figs. 6.5 and 6.6). In these cases the cameras did not move much away from their original center position since the energy was not enough to "trigger" the sigmoidal function in the complex controllers. In these cases the error was less than the other two error types because the system just assumed via the neutral position of the cameras that the stimulus was in the middle of the table, rather than in some distant location.

Fig. 6.7 shows a surface plot of the errors. Errors displayed were limited to 500 mm to provide better contrast. These plateau regions show the areas of greatest error: immediately in front of the cameras, and far away to the right. The nearby errors indicate strong sidelobe effects and the inability of the system to verge to these large disparities. The far-away errors indicate the loosenes of the mechanical hardware.

Fig. 6.8 shows the disparity energy across the stimulus region. Black areas indicate places where the stimulus was completely out of the field of view of the cameras. Light areas show that there was enough energy, even at the periphery, for the cameras to jump to that location and record the data. The v-shape therefore indicates the field of view of the cameras. The energy toward the back of the setup table (large Z) was used to set the center-region and slope of the sigmoidal controllers.

Fig. 6.9 shows the residual errors (x-d energy centroid distances away from x-d center) and again the energy after camera stabilization. Since the system was running in closed loop, the errors

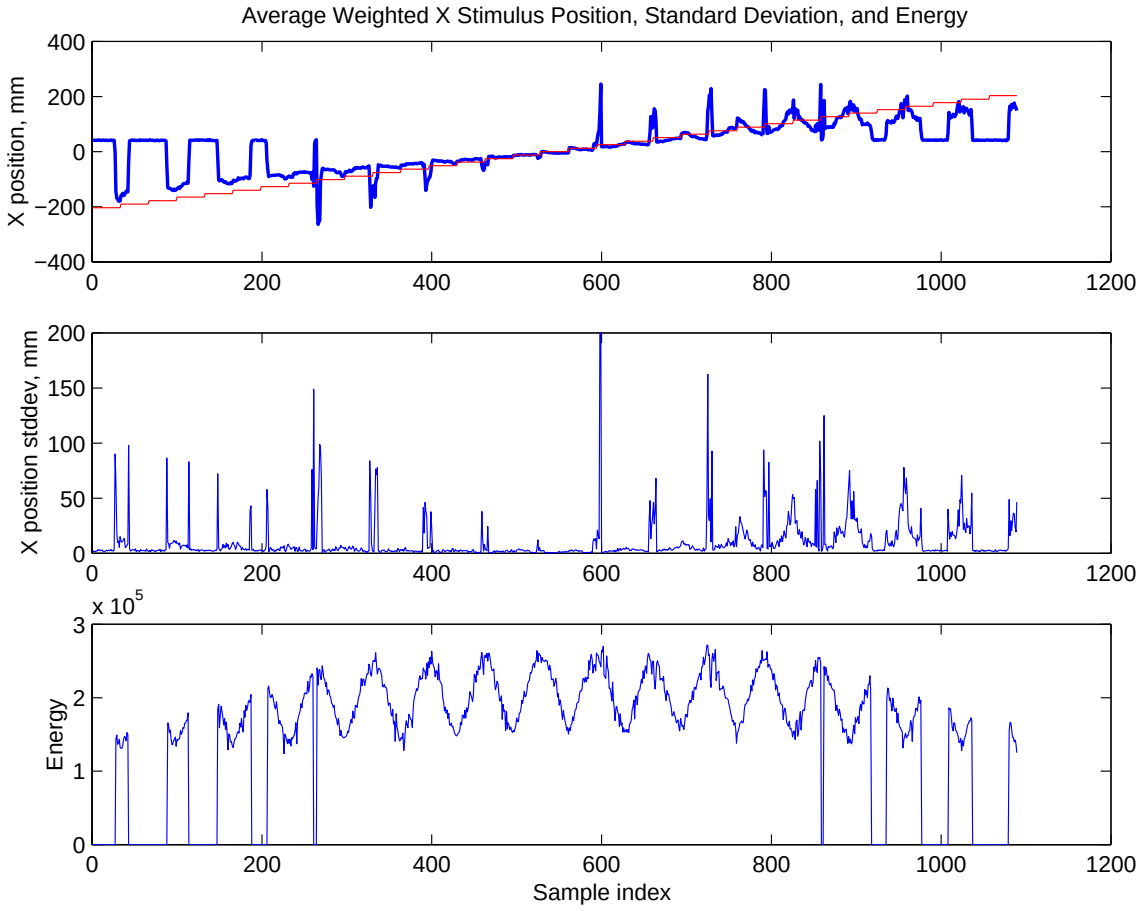



FIGURE 6.5. Experiment 3 results: X position estimates versus index. Thick lines indicate estimated stimulus position; thin lines indicate actual stimulus position. The top plot shows the average X position over all 10 runs of the experiment; the middle plot shows the standard deviation, and the bottom plot shows the total disparity energy. The second plot was truncated to 200 to allow some contrast to be seen. The large spike seen in this plot had a value of 497.

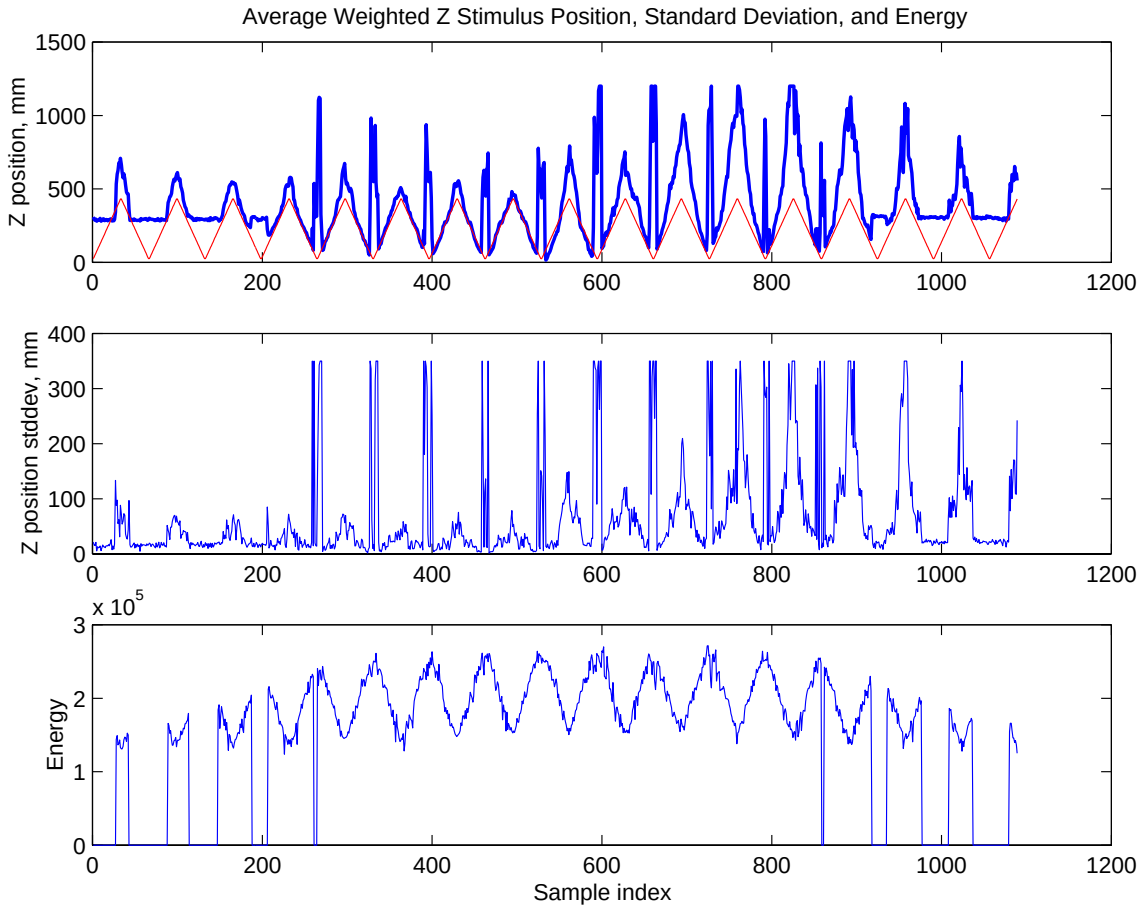



FIGURE 6.6. Experiment 3 results: Z position estimates versus index. Thick lines indicate estimated stimulus position; thin lines indicate actual stimulus position. The top plot shows the average Z position over all 10 runs of the experiment; the middle plot shows the standard deviation, and the bottom plot shows the total disparity energy. The second plot was truncated to 350 to allow some contrast to be seen.

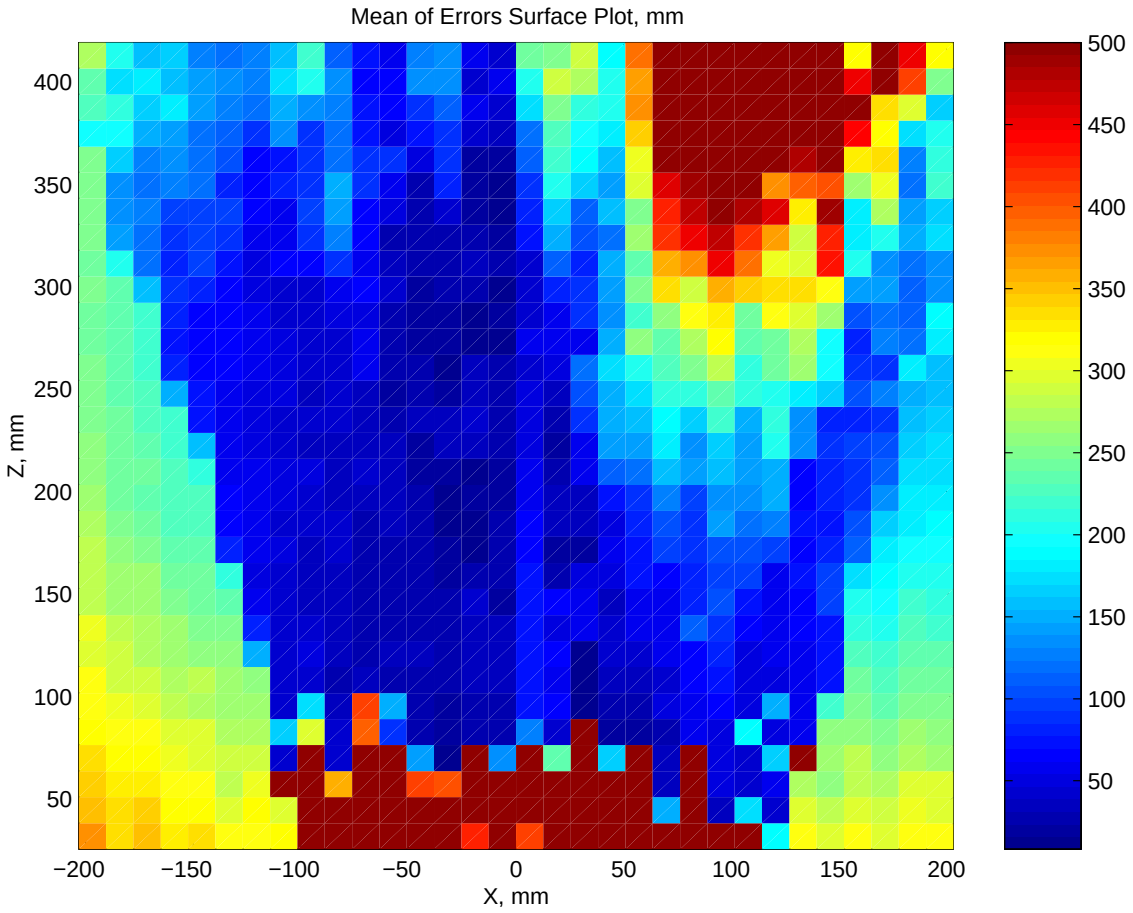



FIGURE 6.7. Experiment 3 results: Stimulus location estimate error versus position. This plot shows that the largest errors occured in regions near the camera and in a region far away and to the right. Otherwise, error increased with distance from the cameras. The maximum errors were capped at 500 mm to improve contrast in the image.

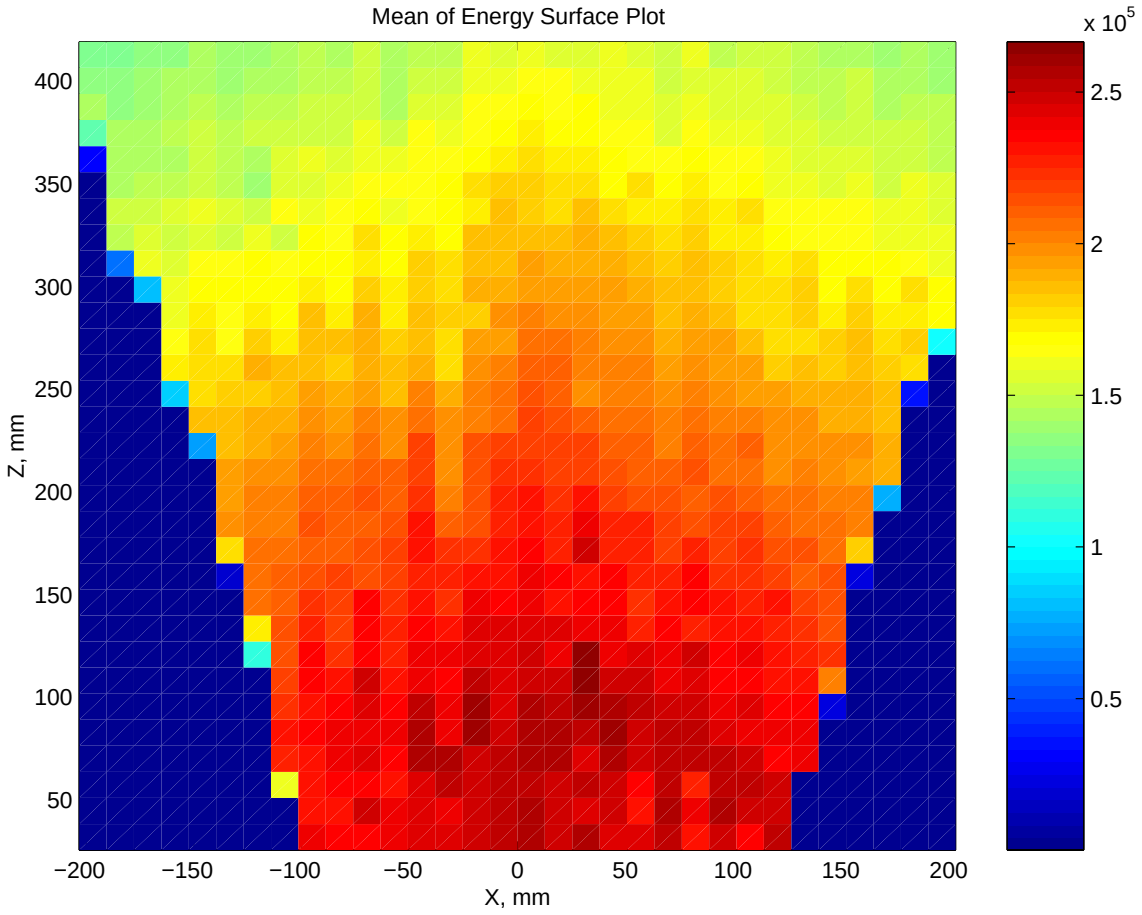



FIGURE 6.8. Experiment 3 results: Average energy versus stimulus position. Energy was near zero in regions beyond the field of view of the cameras. Energy decreased with distance (projected stimulus width) from the cameras.

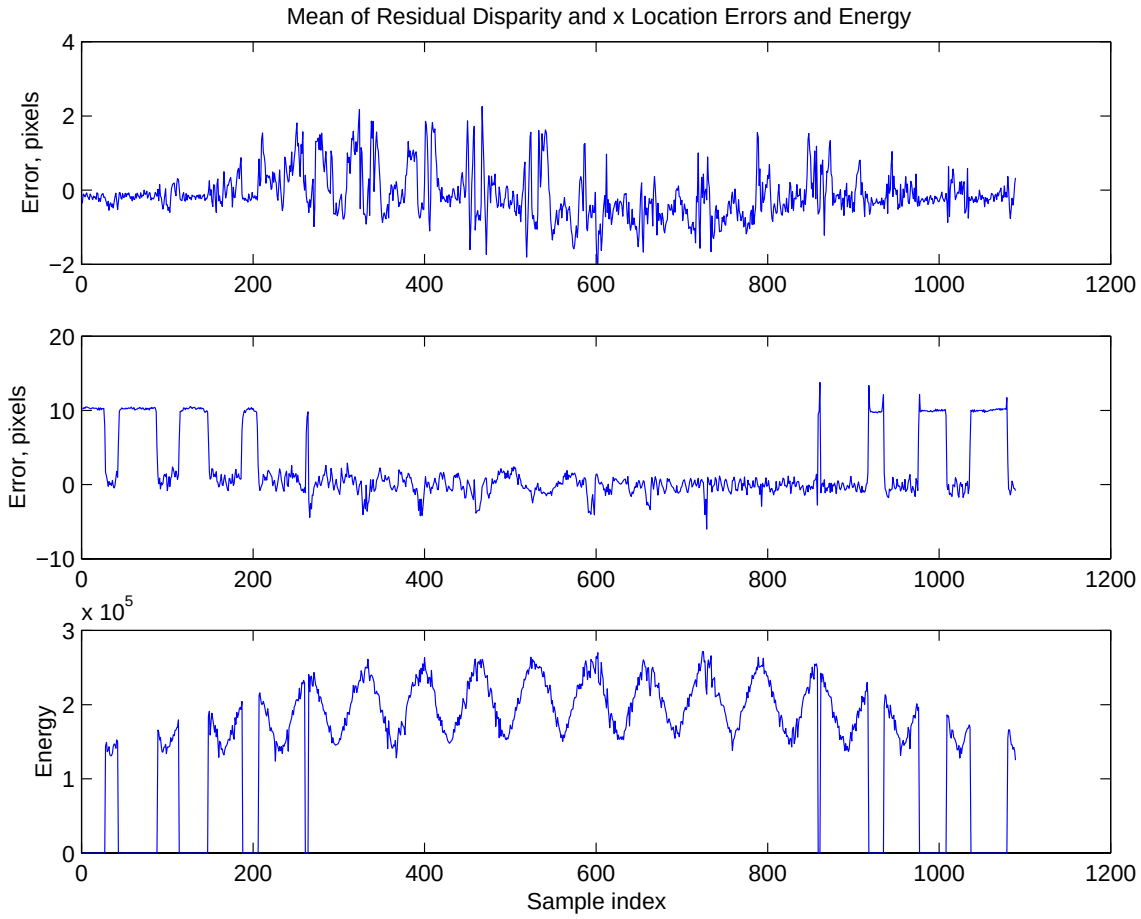



FIGURE 6.9. Experiment 3 results: Mean of residual errors. The disparity errors are with the range of ±2 due to the spacing of the cells within the filterbank. The x errors are near zero in regions where there is enough energy to trigger the cameras to move; values in regions of low energy are probably caused by intensity gradients in the background of the scene.

should remain around zero. Even in the case of large position-estimate errors where the system completely missed the location of the stimulus, the residual disparity error was still low, indicating the system interpreted the overly large disparities (sidelobes) as within range. The disparity limit was ±2 because the large disparity cells were spaced apart in tuning such that -2, 0, and +2 were the closest three cells around zero. Any disparity larger than 2 would have triggered the vergence controller to reduce the disparity error given that the energy was large enough. Similarly for the x-error, the error remained around zero for regions where there was enough energy to trigger the controller.

## 6.4 Experiment 4

The fourth experiment tested the system's ability to verge onto and track a moving target. It started with the stimulus in the center of the table and the cameras verged and centered on the stimulus. The stimulus then moved in 8 concentric circles of expanding radius while the cameras were allowed to track the stimulus in both X and Z. Figs. 6.10 and 6.11 show the mean X and Z components of the known and estimated stimulus position across all 10 runs, as well as their standard deviations.

Each circle was traversed 3 times per experiment for a total of 30 traversals per radius. The disparity and x pixel locations were taken from the energy-weighted centroid of the x-d space (PDM2), not the maximum point, although the tracking and vergence controllers did use the

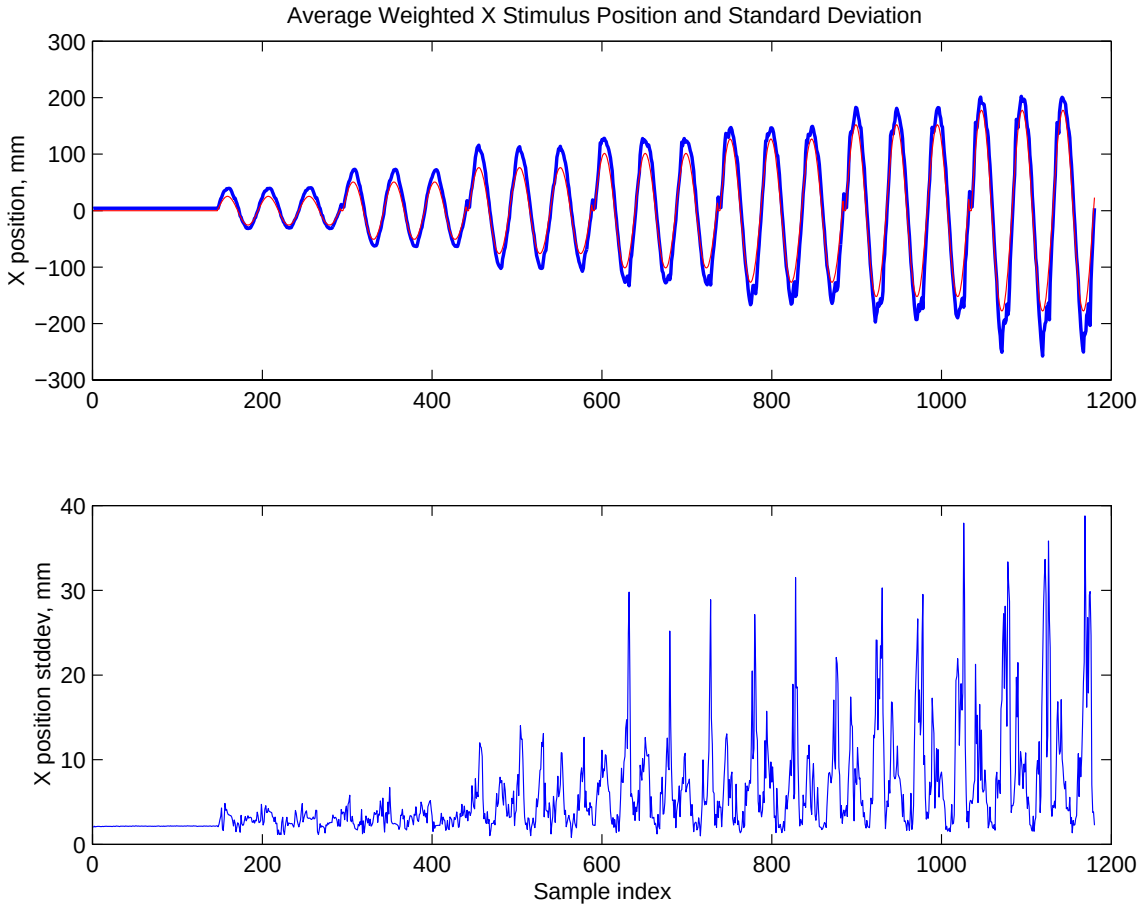



FIGURE 6.10. Experiment 4 results: X trajectory plot and standard deviation. The cameras were verged and centered to the middle of the table and allowed to track the stimulus while it moved around in concentric circles of increasing radius. The top plot shows the average of the estimated trajectories (thick line) and the real trajectory (thin line) across all 10 runs of the experiment. The second plot shows the standard deviation of these estimates.

maximum point in the x-d space (PDM1). There were 48 points per circle, each of which was traversed 3 times per experiment. There were 8 radii (starting at 0, in increments of 1 inch), and 10 total experiments, for a total 11520 samples over 384 unique locations.

Fig. 6.12 shows the remaining disparity and x-location after the cameras settled on each point. Although the maximally-responding cell was always in the center of the x-d space at the end of each sample (by definition of closed-loop, the end of the sample was reached when the maximally responding x-d location was in the middle of x-d space, i.e., when both x and d were zero), the average location of the energy (centroid) in the x-d space was not always centered perfectly. For a closed loop system where the centroid is always near zero, this error is useful to cover "holes" in the x-d space where cells do not exist, and to provide an estimate of where the target is around the intersection of the cameras' axes. Any increase in the density of cells in the x-d space would improve the error estimate as well as provide cells closer to zero which would allow the error to be reduced even further.

Fig. 6.13 shows the same error data compared with the real Cartesian space, rather than the index of points. It is much clearer to see where the system had difficulty estimating the position of the stimulus, and there is corroboration with Experiment 3 in terms of where the system had the most difficulty – toward the back and sides. The system apparently had little problem near the cameras, unlike in Experiment 3, in which it was allowed and able to track the stimulus as it

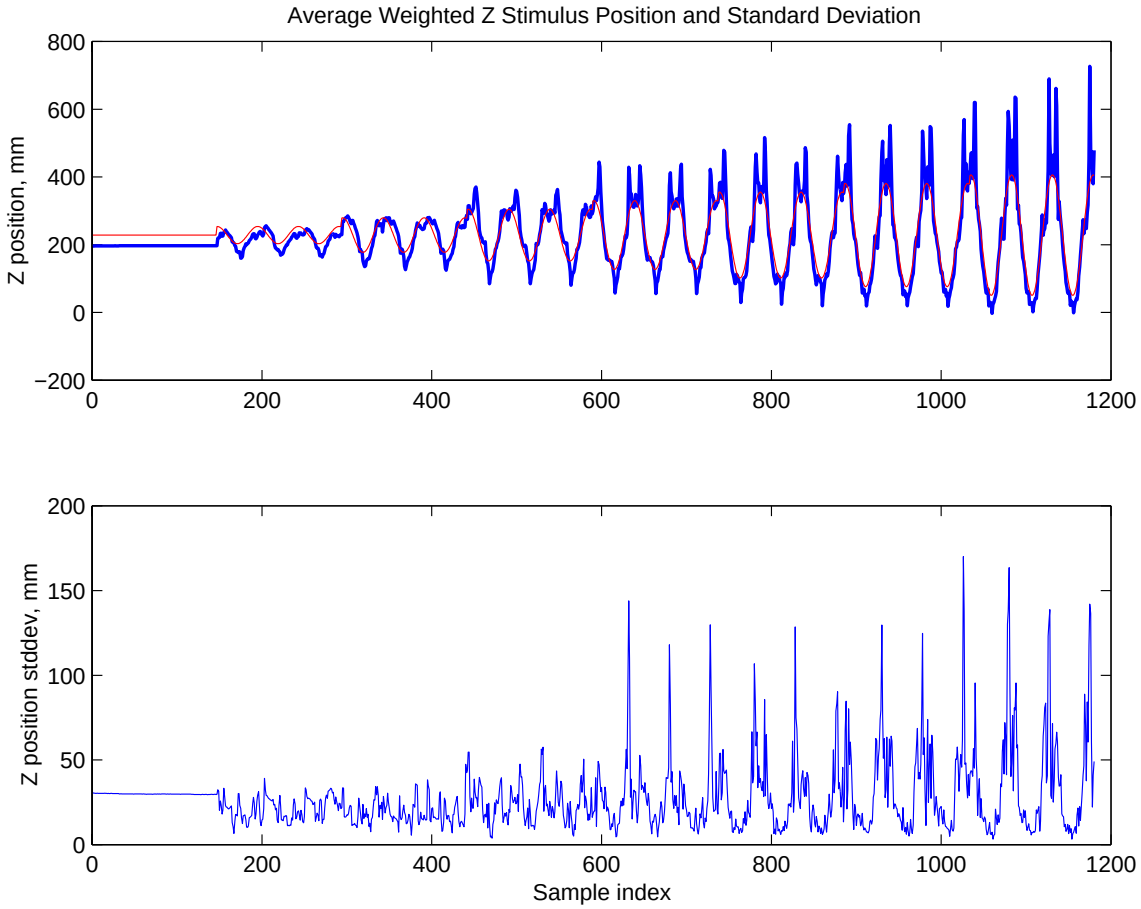



FIGURE 6.11. Experiment 4 results: Z trajectory plot and standard deviation. The cameras were verged and centered to the middle of the table and allowed to track the stimulus while it moved around in concentric circles of increasing radius. The top plot shows the average of the estimated trajectories (thick line) and the real trajectory (thin line) across all 10 runs of the experiment. The second plot shows the standard deviation of these estimates.

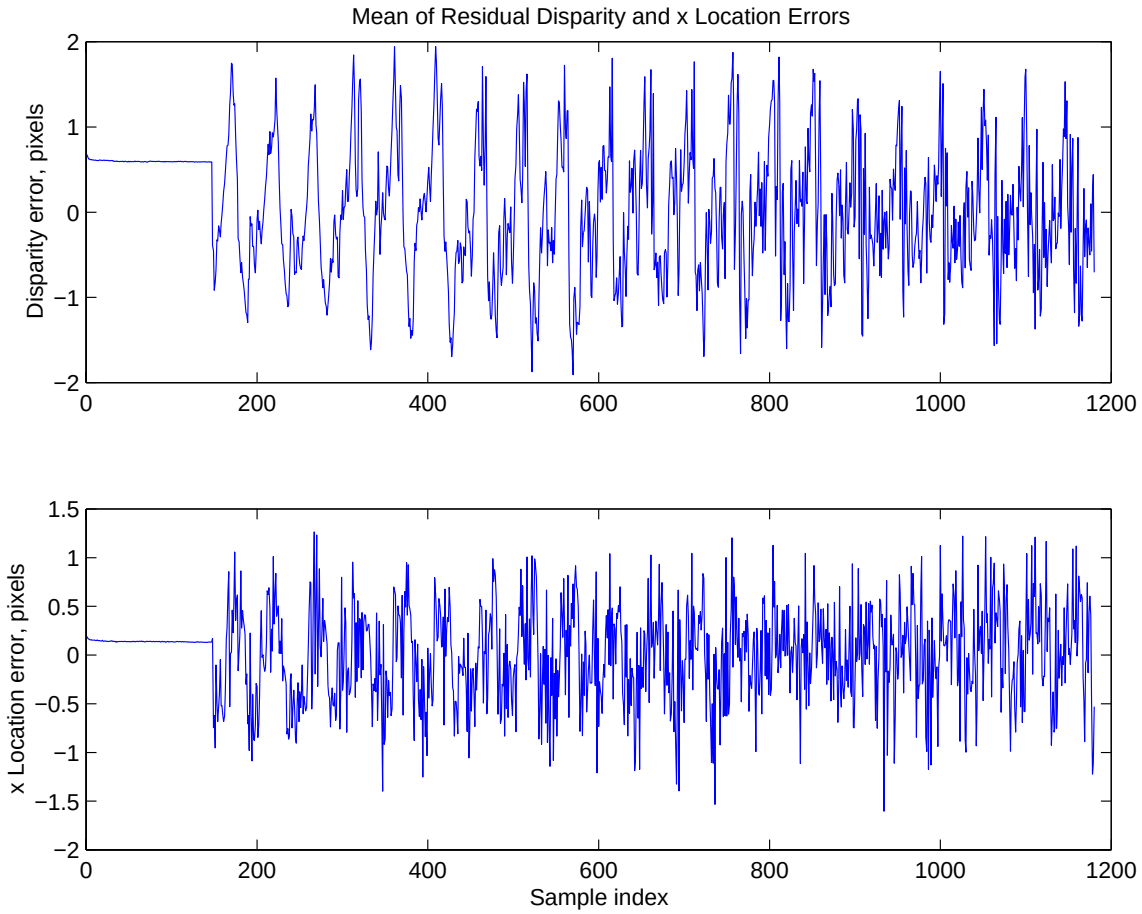



FIGURE 6.12. Experiment 4 results: Mean of residual errors. The cameras were verged and centered to the middle of the table and allowed to track the stimulus while it moved around in concentric circles of increasing radius. This plot shows the average weighted location of the energy in the x-d space for each sample after the cameras had settled in their new positions.

came closer. The minimum, mean, and maximum are shown with the same scale; the difference between the maximum and minimum is formalized in the standard deviation. There appear to be two regions toward the back of the table (large Z) where the system had difficulty. It is not clear why these regions exist; it was expected that the error would increase gradually over distance, rather than be focused in one or two regions. The mechanical system is the mostly likely cause, since Experiment 2 shows the smoothness and symmetry of the error with no camera movements, i.e., the error in Experiment 2 is "well-behaved", whereas for Experiments 3 and 4 the error is less "well behaved".

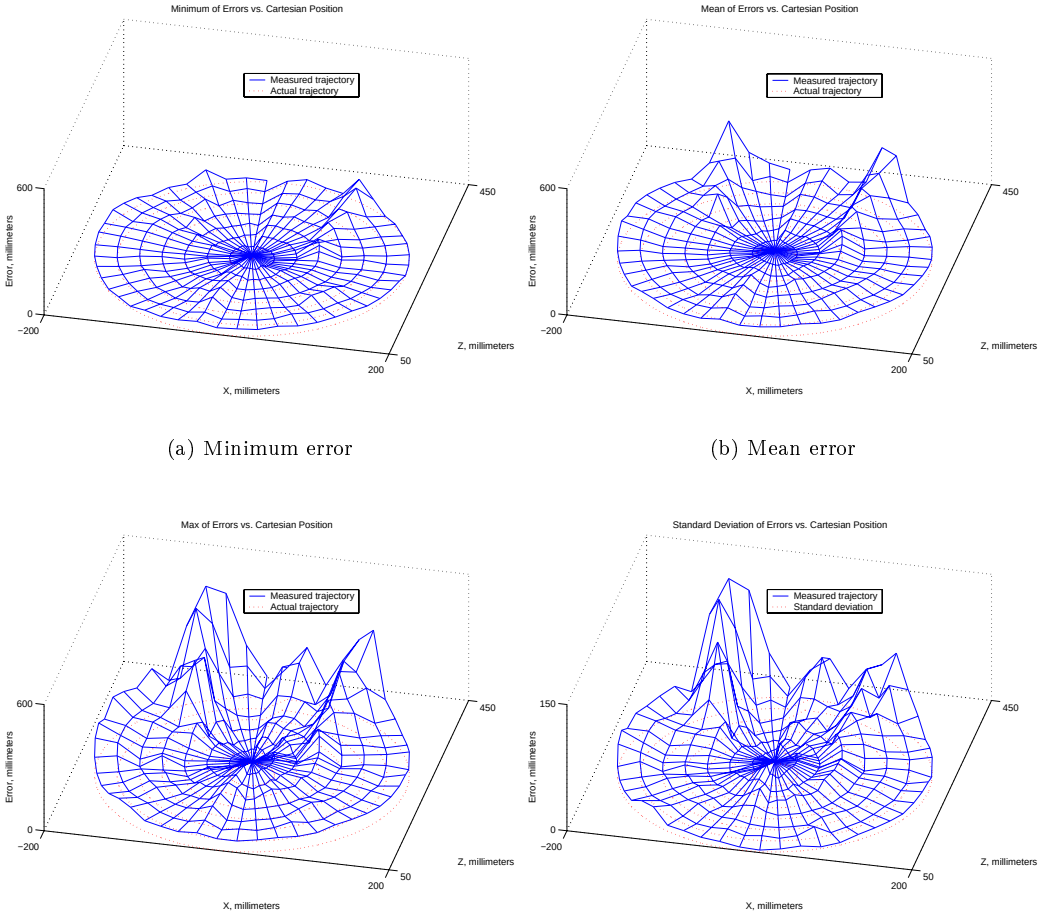
## 6.5 Discussion

The data above show that overall, the system "works": it can estimate stimulus positions around the horopter given positions for the cameras, and can also move the cameras to center the target in x-d space. From a subjective/observational point of view, the tracking performed extremely well; the stimulus remained in the center of the GUI window which shows the superposition of the left and right images, and with a reasonable density of cell tuning in the filter bank, this image was sharp at the edges, indicating minimal residual disparity. In the experiments shown here the residual disparity error was rather large (~2 pixels) because of the large spacing between cell tunings for any given RF width. Informal experiments revealed this error to decrease to below 1 pixel when the cells were spaced one pixel apart. The tuning density was reduced to speed up the experiment time. Each perception cycle took about 1-2 seconds and the experiment required the cameras to not move for 2 cycles after stabilizing before moving the stimulus again. Thus, each stimulus point during the tracking experiment consumed on the order of 5-10 perception cycles. This certainly cannot be considered "real-time".

Another issue is the errors that are shown in Fig. 6.13. These errors appear quite large in this view, but the same data in Figs. 6.10 and 6.11 seem less severe; other experiments showed "about the same" errors when the X and Z dimensions were viewed separately. Since the cameras remained verged on the stimulus within an x-d distance of 2 pixels (Fig. 6.12), the grossness of the error must come from the fact that a) the angular estimates are not very good, and b) small errors in the angle estimate and in the location of the centroid in x-d space result in large estimate errors when the stimulus is far away, i.e., when the vergence angle is small. What remains unclear is why there appear to be two regions of larger-than-average errors in the plots of Fig. 6.13 and one (unexplained) region of extremely large error in Fig 6.7.

Another point that must be taken into account is that these errors are only in the conversion of CCD space to Cartesian space. The majority of the points in the data shown here are points that have been properly centered and verged upon. A mobile robot which is based on biological principles will probably not need to know the position of a stimulus in any particular unit, only that it is "over there" by some amount relative to something else. The ability of the system to verge on, track, and ultimately interact with an object is not related to how well it can report to the outside world the position of that object. The ability of a biologically-inspired robot to function properly without expensive, accurate hardware can be considered an advantage over robots that require knowledge of units such as meters, seconds, etc.

These experiments show that tracking and vergence are indeed possible using disparity energy. The next layer of processing in a mobile robot must decide what to do with the image after the gaze is on the target. By incorporating the features of this software into a real-time hardware system with continuous control of the motors, it should be possible to have a very responsive and robust vergence and tracking system for a mobile robot.




(a) Minimum error

(b) Mean error

(c) Maximum error

(d) Standard deviation of error

FIGURE 6.13. Experiment 4 results: Error min, mean, max, and standard deviation versus position. The cameras were verged and centered to the middle of the table and allowed to track the stimulus while it moved around in concentric circles of increasing radius. This plot shows the minimum (a), mean (b), maximum (c), and standard deviation (d) of the distance from the known stimulus positions to the measured stimulus positions (error) over the Cartesian plane across the 10 runs of the experiment. The camera location is centered in the X axis (of the figure coordinates), nearest the viewer.

Chapter 7

# PROPOSED VLSI ARCHITECTURE AND FUTURE WORK

This system has been designed from the beginning with a hardware implementation in mind. The goal is to place an animate-vision system with vergence and tracking capabilities on a mobile robot. The subgoals of low power consumption, high reliability, and continuous time-and-value operation must be met. Work in the Higgins lab and elsewhere has demonstrated that a multichip analog-VLSI approach using the address-event-representation (AER) (Boahen, 1999; Higgins and Shams, 2000; Higgins and Koch, 2000) to communicate between chips is likely to be a feasible solution. This chapter discusses an architecture (based on a simplification of Fig. 4.4) which fits into the AER neuromorphic analog-VLSI framework, and which satisfies the above goals.

## 7.1 Introduction

It becomes immediately clear that the resources and flexibility in a hardware system are more restricted than in a software system, but the tradeoff is real-time operation and vastly lower power consumption (Higgins and Shams, 2000). In the software system the number and shape of filters and cells may be freely chosen, but a hardware system is more likely to require these parameters to be fixed or severely limited in flexibility, or require a large amount of hardware to match the magnitude of the system found in software. This architecture should be expandable to support many different RF widths and disparity tunings, as well as be able to support the addition of motion-detection circuitry to attain a system of cells which are tuned for a particular combination of disparity, spatial frequency, and temporal frequency (Qian, 1994).

An AER architecture sends the addresses of the locations of *events* from a *sender* chip to a *receiver* chip in real-time via an asynchronous high speed digital bus. Typically the address of a sender chip's pixel (the event) is sent via the AER bus to effect an event on a pixel in the receiver chip. This digital address can be changed in mid-path to manipulate how the sender chip's events map on to the receiver chip's computational circuitry. Both sender and receiver chips consist of arrays of parallel circuitry which perform various useful functions. Since the AER bus is approximately 3 orders of magnitude faster than the analog circuitry's bandwidth (MHz versus kHz), it is feasible to send large numbers of (near) simultaneous events across the bus. The effect is the low-powered, highly parallel connection between layers of computational circuitry. This scheme can be extended to more than 2 layers by adding one or more *transceivers* between sender and receiver, which perform functions of both. Our design is a four layered scheme comprising four unique custom analog VLSI chips.

## 7.2 Description of Architecture

Let us start with two (left and right) sender chips: analog CMOS chips consisting of light-sensitive pixels which communicate their activity via the AER bus. Refer to Fig. 7.1. Each pixel is an adaptive photocell as used in Higgins and Shams (2000). These sender chips should have some "high" resolution, perhaps 100x100 pixels, but we shall only consider a one-dimensional array of pixels for simplicity. These chips send their data to a pair of transceiver chips which perform the required spatial (Gabor-like) filtering. These filters send their events to a transceiver complex-cell chip which contains the disparity-energy computation circuitry. This data is sent via, for example, some address-remapping EEPROMs or FPGAs. These EEPROMs remap the addresses of the sender chips to shift and subsample the images, the effect of which is the cell's disparity tuning and the quadrature-phase relationship between pairs of spatial filters. The complex-cell transceiver chip's pixels each contain one "complex cell" – a circuit which sums and squares four incoming signals to produce a disparity energy output (Fig. 2.1). The receptive field width with which a complex-cell chip works is determined cooperatively by the spatial filters and by the remapping done by the EEPROMs.

For a chosen RF and disparity tuning $\phi = \phi_l - \phi_r$, the number of pixels corresponding to $90^\circ$ and $\phi$ are computed. These are defined as $p_{90}$ and $p_\phi$, respectively, and are used as subscripts to indicate pixel offset from $X_0$. Refer to Fig. 7.2. The EEPROMs are programmed so that for a given "center" pixel located at $X_0$, say, in the left sender chip, the events from that $X_0$ pixel and of the corresponding $\phi$-offset pixel ($X_{p_\phi}$) in the right sender chip are both sent as "left" and

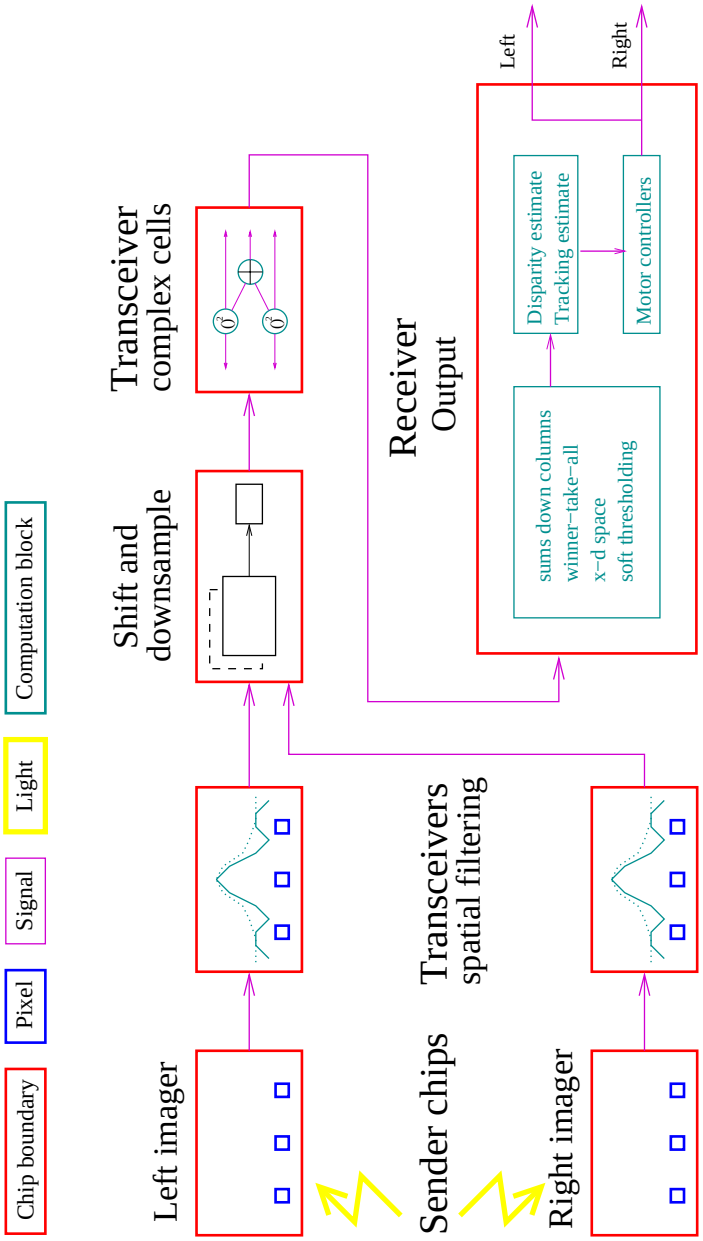



FIGURE 7.1. Hardware architecture block diagram overview. CMOS imagers feed into spatial (Gabor-like) filters. The filtered images are passed to the complex-cell chip via EEPROM mapping which shifts and downsamples the image for the appropriate spatial-frequency and disparity tuning. The complex-cell transceiver then passes its information to a receiver which generates motor control signals. All interchip arrows are AER bus lines.

"right", respectively, to the $X_0$ pixel in the complex-cell transceiver. The events from pixels $X_{p_{90}}$ and $X_{p_{90}+p_\phi}$ are sent as left and right, respectively, to pixel $X_1$ in the transceiver, where $X_1$ is adjacent to $X_0$. This has the effect of downsampling the sender chip an appropriate amount for the RF chosen. After this remapping has taken place, each pixel in the complex-cell transceiver has four inputs – two spaced $\phi^\circ$ apart from its "own" sender pixel, and two spaced $\phi^\circ$ apart from a pixel 90° away.

A cell in the complex-cell transceiver will perform disparity energy computations using its own left and right signals, as well as the left and right signals from a cell adjacent to it. This adjacent cell contains the left and right signals of the quadrature-phase pixel 90° away. Adjacency in the complex cells simplifies greatly the on-chip routing requirements. Since the cells in a row of a complex-cell transceiver are interconnected and share disparity tuning, they are independent of cells in other rows and so any row can be programmed (via the EEPROMs) for any legal disparity without regard to other rows. Thus, if the downsampling leaves enough "free" cells in the complex-cell transceiver, those cells can be tuned for a disparity other than that for which the "used" cells are tuned, thus utilizing more of the chip. Since the cells in any row all share the same tuning, the rows can be organized in any manner, such as by interleaving them. The number of disparities for which a single complex-cell transceiver can be tuned is equal to the ratio of downsampling between the spatial filtering transceiver and the complex-cell transceiver. Bear in mind that the subsampling must be in both x and y dimensions for this to work. The architecture can be further augmented by adding more spatial filters tuned for different frequencies, and by adding more complex-cell transceivers in the case that a single chip cannot provide the number of disparity tunings desired. Of course a new complex-cell transceiver must be added for each new spatial-frequency. Refer to Fig. 7.3.

A smaller RF results in fewer choices of disparity tuning, but also allows for the detection of disparity in high spatial frequency images. The high-resolution limit of course is a system tuned for zero disparity in the highest spatial frequency allowed by the resolution and spacing of the sender pixels. Thus the complex-cell transceiver should have as many pixels as the senders and spatial-filtering transceivers, even though for any usable (nonzero) disparity tuning, not all of them would be used. The upper limit on disparity is the 90° pixel distance for a given RF width; a wider disparity tuning requires a wider RF. A cell tuned for 90° will alias heavily, so this degree of tuning is not recommended. By downsampling we also get the added benefit of adjacent receiver pixels sharing information, thus alleviating problems with arbitrary connectivity in the complex-cell transceiver.

At this point the shift-and-downsample blocks must send to the complex-cell transceiver an event from pixel $X_0$ for each disparity for which the transceiver is tuned. Since the $X_0$ pixel is not shifted, its events get sent to the multiple transceiver rows which require it. This generates the need for a one-to-many mapping between the spatial-filter transceiver and the complex-cell transceiver. Rather than shifting only the right pixel $\phi^\circ$, we can shift both the left and right pixels $\frac{\phi}{2}^\circ$ in opposite directions. This not only improves the spatial symmetry of the response to disparity stimuli, but also alleviates the shift-and-downsample block from having to service a one-to-many request as would be required with a multiple-disparity complex-cell transceiver receiving multiple events from $X_0$. The advantage is that inexpensive EEPROMs, which can only deliver a one-to-one or a many-to-one mapping between addresses, can still be used, rather than having to use custom digital logic such as an FPGA.

## 7.3 Spatial Filtering

So far we have not discussed the Gabor functions. The discussion in Chapter 2 suggests that "perfect" Gabor functions are not required for achieving disparity sensitivity, however some filtering is required. By not filtering (spatially smoothing) the pixels as we downsample them, we run the risk of aliasing some region of high spatial frequency (via Nyquist in the space domain). We also do not tune the system for vertical lines as the Gabor filters in this research do, a requirement which informal experiments have shown is critical for proper functioning of disparity measurement. It is possible to perform two-dimensional spatial filtering by including circuitry in the sender chips (Shi, 1999), but this greatly limits the system's scalability and flexibility. Therefore, the filtering here has been shown to take place outside the sender chips, where multiple spatial frequencies may be filtered. The spatial filtering circuits described by Shi are not perfect Gabor filters, but instead they exhibit a Laplacian rather than a Gaussian exponential decay modified by a sinusoid.

This architecture suggests to use only even filters at all spatial locations, rather than even and odd filters as is possible with the Shi circuits. This is another simplification, although the architecture certainly does not prevent one from adding odd filters in addition to the even ones. By implementing only even filters we are not approximating the disparity tuning or the quadrature phase by changing the phases of the sinusoids, as has been discussed in this thesis, but by shifting

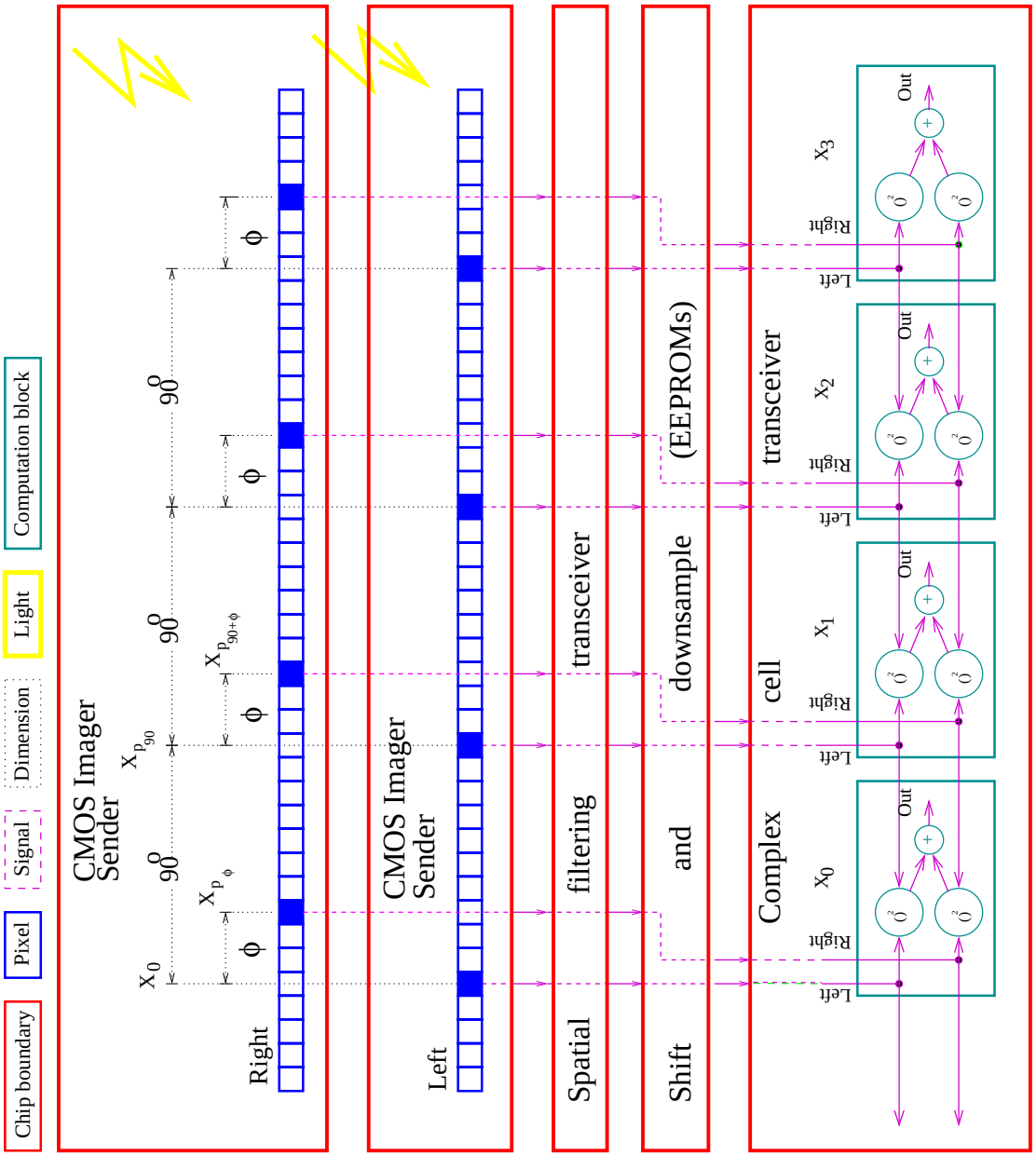



FIGURE 7.2. Hardware architecture block diagram detail. This figure shows how the sender chips are filtered and subsampled to approximate disparity tuning and to effect the quadrature-phase nature of the disparity energy circuitry. Each pixel in the complex-cell transceiver consists of an $(a+b)^2+(c+d)^2$ type of circuit which calculates the energy of its inputs. A pixel in this transceiver uses input from cells in the sender which are $\phi^\circ$ apart, as well as input from its adjacent pixel which provides similarly-spaced input from pixels 90° away.

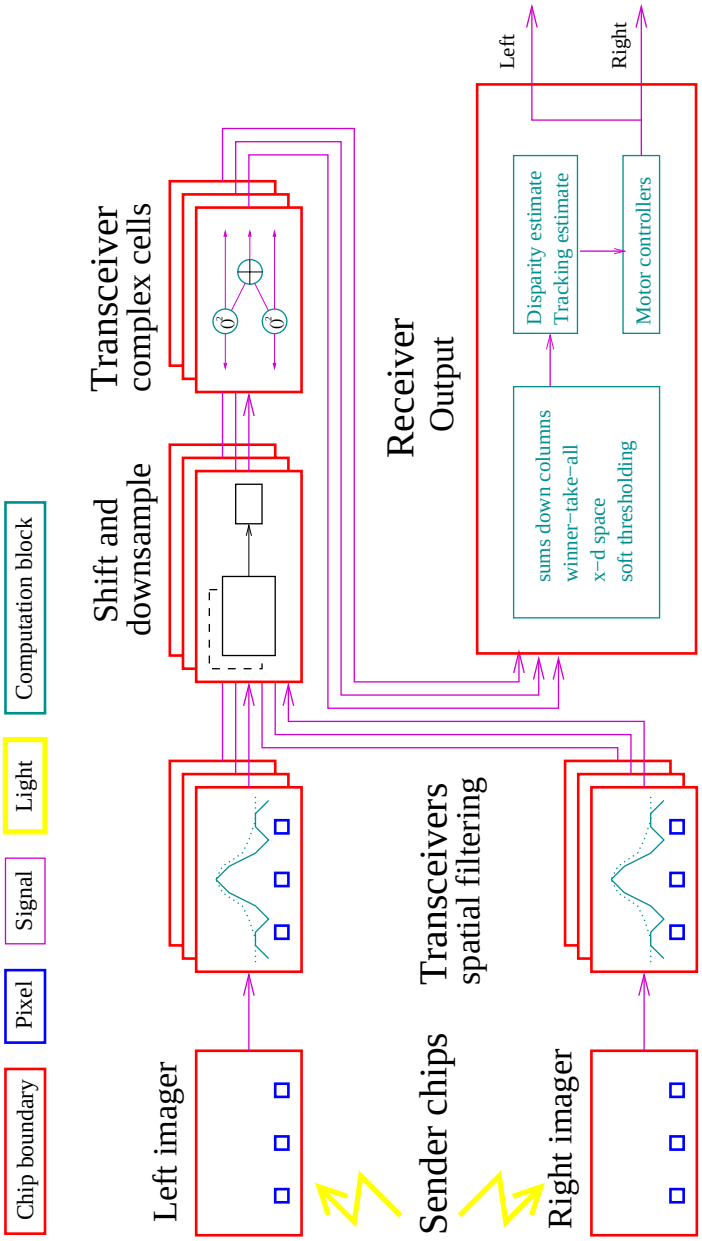



FIGURE 7.3. Expanded hardware architecture block diagram. This figure shows additional spatial filtering and disparity-tuned transceivers, connected via additional shift-and-downsample blocks. The outputs of the complex-cell transceivers still combine in the final receiver chip.

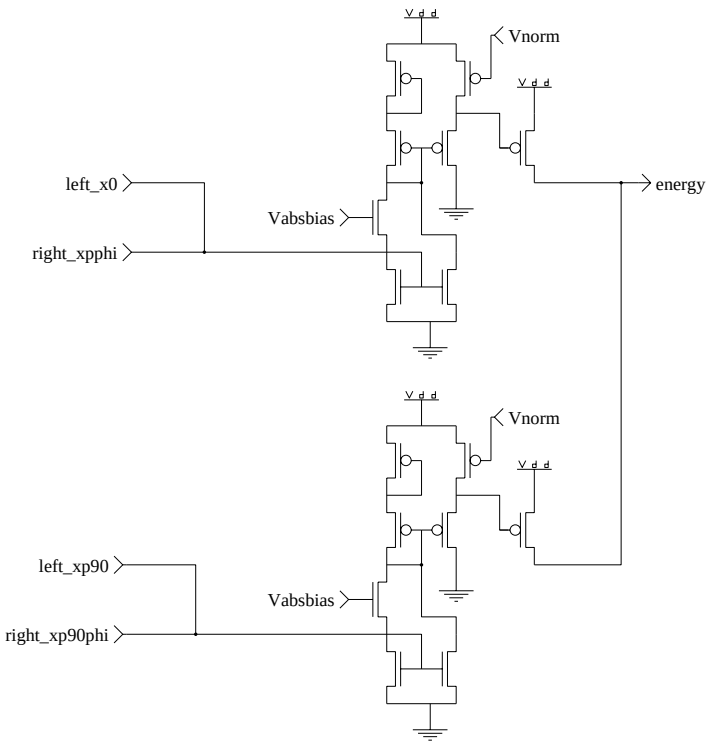



FIGURE 7.4. Complex cell circuit diagram. This circuit implemements the function $k\left|a+b\right|^2 + k\left|c+d\right|^2$, where $a, b, c, d$ and $k$ are left_x0, right_xpphi, left_xp90, right_xp90phi, and a normalizing factor. Each pair of currents is summed at its common node via KCL (Kirchoff's Current Law) to create $a+b$ and $c+d$. The lower three transistors in each half-circuit implement an absolute-value function on these currents, generating currents $|a+b|$ and $|c+d|$. The upper five transistors in each half circuit perform a normalized squaring of these currents, generating the output currents $k\left|a+b\right|^2$ and $k\left|c+d\right|^2$ which get summed at their output via KCL to yield $k\left|a+b\right|^2 + k\left|c+d\right|^2$.

the Gaussian and the sinusoid the required amount for the tuning. This has been shown in Ohzawa *et al.* (1997) not to be the correct model for a disparity-tuned cell (in cats), but for small disparity tunings, the approximation may be made (Qian, 2000). In fact, the contention between this phase-shift vs. position-shift model has been in the neurobiological literature for some time, and it is not clear that it has been entirely resolved. Care must be taken to tune the Shi filters to match the desired RF of the cell so they match the tuning programmed into the EEPROMs.

## 7.4 Circuit Level Implementation

Fig. 7.4 shows a proposed circuit diagram for a cell in the complex-cell transceiver. It is based on circuitry used in an implementation of the Adelson-Bergen motion algorithm (Higgins and Korrapati, 2000), which computes motion energy similarly to the disparity energy used here (Fig. 2.1). The current-mode translinear circuit consists of 16 MOSFETS operating in their subthreshold region. Each half of the circuit operates on either the even or odd left and right input signals. A summation of two current signals is performed via Kirchoff's Current Law (KCL) by simply sharing a node between the two signals. The signal names correspond to the pixel locations in Fig. 7.2. The bottom portion of each half-circuit creates an absolute value of the sums of the incoming signals, so as to allow bidirectional inputs. This absolute value is fed into the top portion of the circuit, which performs a squaring operation, multiplied by some factor $k$ determined by Vnorm. The outputs of both half-circuits are summed together at the output node.

## 7.5 Output

The outputs of the complex-cell transceivers feed into a final receiver chip which sums the outputs of the complex cells. This chip will sum across spatial frequency for each disparity tuning at each spatial location; in effect, it implements the sum-of-columns of the filterbank. The receiver chip will also incorporate the integrative and thresholded controllers for vergence and horizontal tracking. The output of the chip will be a position signal for the left and right cameras.

## 7.6 Future Work

Another point of future work is the integration of the disparity-based vergence and tracking system with motion-sensitive cells, or with cells that are tuned for both disparity and motion (Qian, 1994). By combining these two, a much more complete early-vision system can be developed, which bases horizontal tracking on motion rather than on disparity energy, and which opens the possibility of creating a complete early-vision system, including vergence, target-tracking, saccades (short, quick movements of the eyes), and VOR (vestibular-ocular-reflex). This system would also be able to use depth and motion to avoid obstacles, navigate, etc., if coupled to higher-level control structures. The separation of the imagers and the spatial filters allows a large scale system in which motion and disparity cells can be tuned for a wide array of spatial frequencies, and in which the outputs of the spatial filters can be shared between the disparity and motion cells.

# Chapter 8

# SUMMARY AND CONCLUSION

The purpose of this project was to develop a system to control the angular positions of a pair of stereoscopic cameras for use in a mobile robot. Specifically, the goal was to control the vergence of the cameras and to perform horizontal tracking of objects. Other researchers have achieved these tasks by various means, but this author did not find any who used a biologically-inspired representation of image disparity to do so.

Disparity energy has been found to be the mechanism used in cats for the perception of depth, and it was this energy that has been used to control the cameras. An ad-hoc hill-climbing algorithm based on an 8-state finite state machine was initially developed to verify that disparity energy could indeed be used to control vergence. This solution did not always find the maximum point, and proved rather fragile once it did. A phase-based horizontal tracking method was also explored, but this also proved overly sensitive to shadows and intensity gradients in the image, and was therefore quite unusable. Neither of these solutions was biologically-inspired.

A biologically-inspired filter-bank method of disparity-energy-based vergence control was then developed. The control system uses population encoded complex cells which give weight to movement in various dimensions. The cells feed into "complex" controllers which integrate position error over time to control the motion of the cameras. These controllers are parallel and independent, suitable for use in a "subsumption architecture"-based robot, wherein the computational models are small in scope and predictably reactive to achieve more complex behavior than would be possible using classical robotic programming techniques.

We have seen via a series of experiments that the system can estimate reasonably well and very consistently the position of an idealized stimulus (the vertical bar) placed in front of stationary cameras, and to be able to track the stimulus with the cameras. This tracking allows the stimulus to move in the complete range of the system's field of view and to maintain the stimulus centered left-to-right and within the allowable fusible region in depth.

Finally, an architecture to realize this system in analog VLSI hardware has been introduced and discussed. This hardware continues the biologically-inspired theme of low-power, highly parallel computation, and brings the possibility of a robust, advanced vision system closer to the domain of mobile robotics.

During the investigation of the use of disparity energy in controlling the cameras, several problems were encountered, including phase aliasing and the monoscopic response problem. Two solutions to the phase aliasing problem were introduced, one of which was implemented, and one solution to the monoscopic response problem was introduced and implemented. The error in the physical platform was estimated and determined to be caused by slack and imperfections in the servo motors and pushrods, and by the discrete nature of the servo controller. Finally, as neuroscience has contributed to engineering by providing us with models for cells in cat visual cortex, this engineering project provides for neuroscience a prediction of neural arrangement as a solution to the phase aliasing problem.

A GUI was developed to observe the status of the system and to control the cameras manually as needed for the experiments. The GUI also served as a "common sense" indicator to let the experimenter know if the system was stuck in a sidelobe and if the system was working overall.

The use of disparity energy has been shown to be sufficient and useful for the control of stereoscopic vergence, and its use in horizontal tracking has been shown as well. By implementing a population-encoded disparity-energy-based control system in analog VLSI hardware, a robust, real-time, early-vision system for mobile robots can be developed.

**Appendix A**

# FINITE STATE MACHINE

Here we describe the 8-state finite state machine used in the single-cell hill-climbing vergence method. Please refer to the bubble diagram in Fig. 4.2. Since the energy may be low-pass filtered with each cycle or require other preprocessing steps, and since the cell update is called separately from the state machine (i.e., the state machine does not call the cell update when it needs a new energy value; first the cell energy is updated, then the state machine is called) an auxiliary state is created which is called one or more times between real states. Keep in mind therefore that the energy in any one state is the energy derived from the vergence movements of the previous states. Also keep in mind that any vergence "increment" used in this pseudo-code refers to a movement in the current direction (either positive or negative) with the current increment distance.

There are three hysteretic windows which are constantly changing, and which provide thresholds above and below a locally-relevant maximum value for the system to decide to move or not. These are the local hysteretic window (LHW), recomputed with each move, the local-maximum hysteretic window (LMHW), recomputed each time a local maximum is found, and the global-maximum hysteretic window (GM HW), recomputed each time a local maximum is reached (after having been found).

## A.1 State Machine Pseudocode

```
Clear global maximum, thresholds, etc.
Set vergence increment distance to some value > 0.
Set vergence increment direction to +1.

state 999 (auxiliary state):
    Low-pass filter energy (running average).
    Update global maximum energy if required.
    If energy < global hysteretic window (GMHW) then
        Set vergence increment distance to some value > 0.

state 0 (initialize sweep):
    Start global scan by initializing to some min. sweep vergence angle.
    Goto state 1.

state 1 (sweep):
    Record energy for this vergence angle.
    If energy > all previous energies, then mark this vergence angle.
    Increment vergence.
    If vergence is at maximum sweep angle then
        If multiple-sweeps desired and number of sweeps not been then
            Reverse vergence increment direction.
        Else
            Goto state 2.
    Else
        Goto state 1

state 2 (jump to best known angle):
    Jump to vergence angle which gave the most energy from state 1.
    Goto state 3.

state 3 (begin maximum search, compute threshold):
```

```
    If vergence increment distance != 0 then
        Increment vergence.
        Compute a hysteretic window (LHW) as a percentage of energy.
        Goto state 4.

state 4 (compare energy change to previous LHW):
    If vergence increment distance != 0 then
        If energy < LHW then (energy decreased with movement)
            Change direction (going the wrong way).
            Increment vergence three times (to overcome motor slack).
            Recompute LHW with current energy.
            Goto state 4.
        Else if energy > LHW then (energy increased with movement)
            Increment vergence (keep climbing!).
            Recompute LHW with current energy.
            Goto state 5.
        Else if energy is within LHW then (energy did not change much)
            Increment vergence (keep looking for a slope).
            Goto state 4.

state 5 (compare energy change to previous LHW, again):
    If vergence increment distance != 0 then
        If energy < LHW then (last vergence was better than this one)
            Recompute LHW with current energy.
            Compute LMHW with local maximum energy (from state 4).
            Mark energy used in state 4 as local maximum.
            Mark vergence used in state 4 as best local vergence.
            Change direction (go back and find local maximum).
            Increment vergence.
            Goto state 6.
        Else (keep hill climbing)
            Increment vergence.
            Goto state 5.

state 6 (go back looking for local maximum):
    If vergence increment distance != 0 then
        If energy < LMHW then
            If vergence - best local vergence > 3 then
                (Missed the local max by a few jumps, search again.)
                Change direction.
                Goto state 4.
            Else
                Increment vergence.
               Set LHW around current energy.
                Goto state 6.
       Else
           (Local maximum has been achieved.  Assume it's global max.)
           Set global maximum to current energy.
           Set local maximum to current energy.
           Set GMHW around current energy.
           Set LMHW around current energy.
```

```
Set LHW around current energy.
Set vergence increment value to 0 (so it does not move)
Goto state 4.
```

## Appendix B

# CCD TO CARTESIAN SPACE

The majority of the disparity and complex cell discussion has taken place solely within *CCD Space*. This information needs to be converted to real-world *Cartesian-space* so we can measure how well the system works. We must use the vergence and cyclopean angles, along with the camera geometry to convert x in pixels to X in millimeters, and to convert d in pixel-disparity to Z in millimeters. The equations used for this conversion are rather tedious and not very relevant to the scientific contribution of this thesis, so they and their derivations appear here. Also see Woods *et al.* (1993) for another perspective, and refer to Fig. B.1 for a geometric visualization of the setup.

According to the diagram, $b$ is the baseline distance between the nodal points (the points around which the cameras pivot and through which all rays are drawn; point NL and NR in the diagram); $b_l$ and $b_r$ are the left and right components of $b$; $\alpha_c$ is the cyclopean angle (the average left-to-right angle of the two cameras, $\alpha_c \equiv \frac{\alpha_L - \alpha_R}{2}$); $\alpha_v$ is the vergence angle (the angle between the centerlines of the two cameras $\alpha_v \equiv \alpha_L + \alpha_R$); $\alpha_L$ ($\alpha_R$) is the angle of the left (right) camera toward the origin relative to the line L1-L2 (R1-R2); $Xccd$ ($Yccd$) are the horizontal (vertical) dimensions of the CCD imaging plane in pixels; $Xlc$ ($Xrc$) is the horizontal distance between the centerpoint of the left (right) CCD imaging plane to the target's projection through the nodal point on that plane in millimeters (the target is point B in Fig. B.1); $lpx$ ($rpx$) is the left (right) x-pixel coordinates; $lpy$ ($rpy$) is the left (right) y-pixel coordinates; $xres$ ($yres$) is the number of pixels across the whole imaging plane in the x (y) dimension for both cameras; finally $f$ is the focal length of the lenses. When $lpx$ and $rpx$ are both the same then the target is at the horopter, and when they are both zero, the target is at the axial intersection, point A in the diagram. The entire goal of this project is to move point A so it coincides with point B. The value of Y does not really interest us since we are only doing horizontal disparity and tracking, but it is included for completeness.

Consider $\angle AL1B$. This is equal to $\alpha_l$, which is $\tan^{-1}\left(\frac{X_{lc}}{f}\right)$. Therefore

$$\angle L2L1B = \alpha_L + \tan^{-1}\left(\frac{X_{lc}}{f}\right) \tag{B.1}$$

Similar reasoning yields

$$\angle R2R1B = \alpha_R + \tan^{-1}\left(\frac{X_{rc}}{f}\right) \tag{B.2}$$

Now consider $\triangle BL1R1$.

$$\tan\left(\angle L2L1B\right) = \frac{b_l}{Z} = \frac{X + \frac{b}{2}}{Z} \tag{B.3}$$

and

$$\tan\left(\angle R2R1B\right) = \frac{b_r}{Z} = \frac{\frac{b}{2} - X}{Z} \tag{B.4}$$

so that

$$\tan\left(\angle L2L1B\right) + \tan\left(\angle R2R1B\right) = \frac{b_l + b_r}{Z} = \frac{b}{Z} \tag{B.5}$$

$$\Rightarrow \quad Z = \frac{b}{\tan\left(\angle L2L1B\right) + \tan\left(\angle R2R1B\right)} \tag{B.6}$$

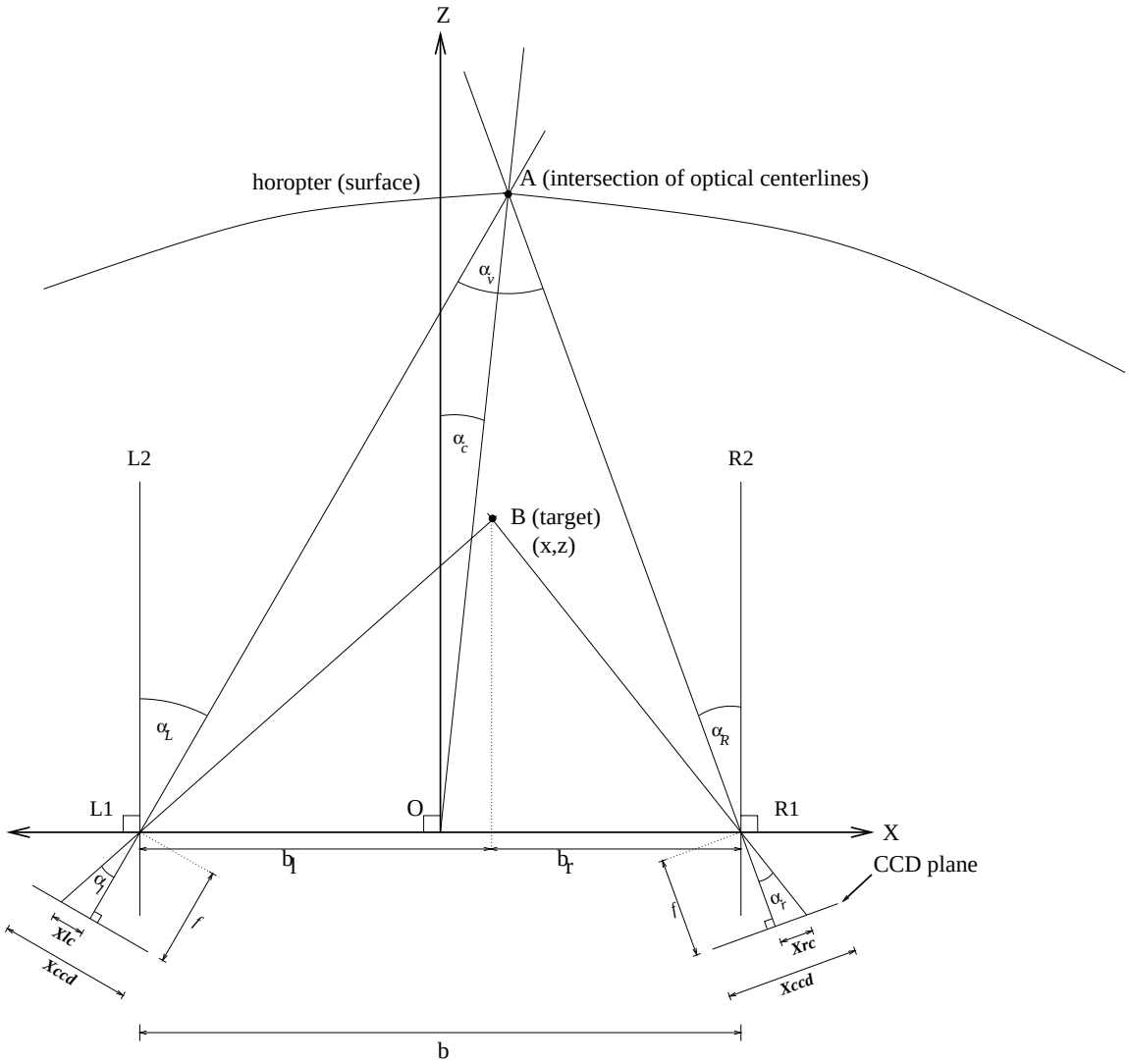



FIGURE B.1. Geometric setup for conversion from CCD space to Cartesian space. This is a plan view of the stereoscopic camera arrangement.

Subtracting Eq. B.4 from Eq. B.3, solving for X and substituting Eq. B.6 yields

$$X = (\tan(\angle L2L1B) - \tan(\angle R2R1B)) \cdot \frac{Z}{2} \tag{B.7}$$

$$= \left(\frac{\tan(\angle L2L1B) - \tan(\angle R2R1B)}{\tan(\angle L2L1B) + \tan(\angle R2R1B)}\right) \cdot \frac{b}{2} \tag{B.8}$$

Substituting Eqs. B.1 and B.2 into Eq. B.6 yields

$$Z = \frac{b}{\tan\left(\alpha_L + \tan^{-1}\left(\frac{X_{lc}}{f}\right)\right) + \tan\left(\alpha_R + \tan^{-1}\left(\frac{X_{rc}}{f}\right)\right)} + Z_{offset} \tag{B.9}$$

and

$$X = \frac{Z}{2} \cdot \tan\left(\alpha_L + \tan^{-1}\left(\frac{X_{lc}}{f}\right)\right) - \tan\left(\alpha_R + \tan^{-1}\left(\frac{X_{rc}}{f}\right)\right) \tag{B.10}$$

where $Z_{offset}$ is the distance from the edge of the stimulus table to the origin of the camera setup and

$$\alpha_L = \frac{\alpha_v}{2} + \alpha_c \tag{B.11}$$

$$\alpha_R = \frac{\alpha_v}{2} - \alpha_c \tag{B.12}$$

$$Xlc = \left(-lpx + \frac{xres}{2}\right)\frac{Xccd}{xres} \tag{B.13}$$

$$Xrc = \left(rpx - \frac{xres}{2}\right)\frac{Xccd}{xres} \tag{B.14}$$

$$Ylc = \left(lpy - \frac{yres}{2}\right)\frac{Yccd}{yres} \tag{B.15}$$

Although we are not using the Y dimension, it is still possible to derive its value from the CCD data:

$$Y = \frac{Z\,(Ylc + Yrc)}{2f} \tag{B.16}$$

where

$$Yrc = \left(rpy - \frac{yres}{2}\right)\frac{Yccd}{yres} \tag{B.17}$$

Fig. B.2 shows a mapping of CCD space to Cartesian space for several values of vergence and cyclopean angle.

A preliminary test was run to see if the above equations worked, and for the most part they did, except there was too much offset error in the Z direction, and a gain error in the X direction. At first this was thought due to an error in the size of the CCD itself, which is quite small (3.6mm in the horizontal direction), but it was difficult to account for the data with this allowance. In addition to a measurement (modeling) error, it was discovered that one of the assumptions made

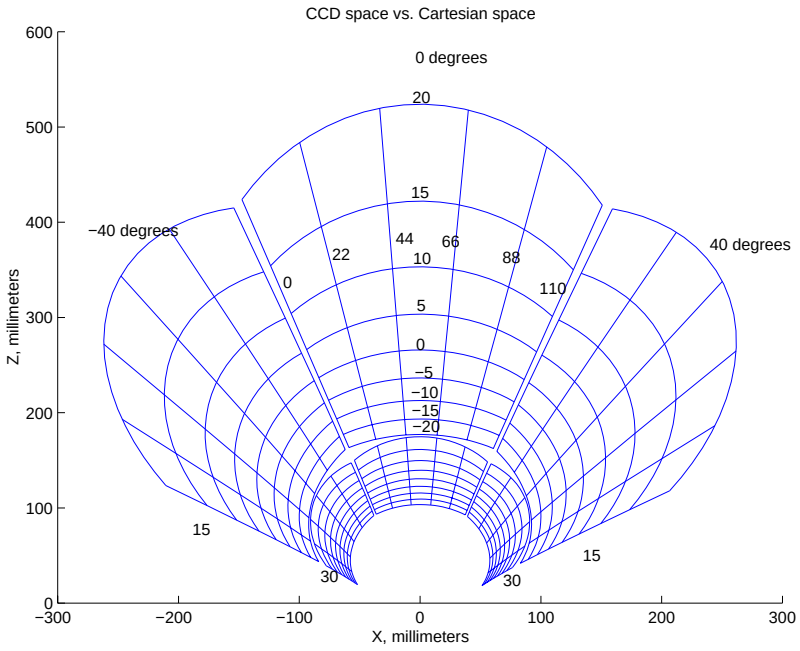



FIGURE B.2. Mapping CCD space to Cartesian space. Plot A shows an example of mapping CCD space to Cartesian space using several values of vergence and cyclopean angle. The grid within each section is the x-d (CCD) space. The parameters used were b=70mm, f=5mm, Xccd=3.6mm, yielding an FOV of 35 degrees. Each section is 110x40 pixels, spaced 22 and 5 pixels in the x and d directions, respectively, to reduce clutter. The cyclopean angles are -40, 0, and 40 degrees from left to right, and the vergence angles are 15 and 30 degrees. Note that although the sections do not intersect each other in the plot for clarity, in fact the cyclopean and vergence angles can be continuous, and thus a continuum of sections exists. Note that the Z depth represented by disparity changes not only with vergence and cyclopean angle (sections), but also within a section, particularly at small vergence angles. The pixel dimensions within the top middle section and the vergence and cyclopean angles are shown.

in the geometry was incorrect for the cameras: The nodal point and the pivot point of the cameras in the drawing are the same. The real system, however, has an offset in the Z direction between the nodal point and the pivot point, resulting in a baseline between nodal points $b$ which shortens with increasing vergence angle. The new equations to determine X and Z are the same as the old, except for $b$ and $z_{offset}$ which instead of being constant, are now

$$b_{nodal} = b_{pivot} - \alpha_v \sin(z_{nodal}) \tag{B.18}$$

$$z_{NewOffset} = z_{offset} - \alpha_v \cos(z_{nodal}) \tag{B.19}$$

where $b_{nodal}$ is the distance between nodal points, $b_{pivot}$ is the distance between pivot points, $z_{nodal}$ is the distance between the nodal and pivot points along the camera's optical axis, and $z_{NewOffset}$ is the new distance between the center of the nodal points and the center of the pivot points. Although the discovery of this error in assumption was inspired by funny data, it is not clear that the discrepancy had much of an effect. The distance between the pivot and nodal points is so small as to render the correction negligible, and thus only the original equations were used in the data shown in Chapter 6.

# References


Alvarez, Tara L., John L. Semmlow, and Weihong Yuan (1998). Closely spaced, fast dynamic movements in disparity vergence. *J. Neurophysiology* pp. 37–44.

Alvarez, Tara L., John L. Semmlow, Weihong Yuan, and Paula Munoz (1999). Dynamic details of disparity convergence eye movements. *Annals of Biomedical Engineering* 27: 380–390.

Araujo, Helder, Jorge Batista, Paulo Peixoto, and Jorge Dias (1996). Pursuit control in a bincoluar active vision system using optical flow. In *Proc. ICPR*, pp. 313–317.

Asai, Tetsuya, Masahiro Ohtani, Hiroo Yonezu, and Naoki Ohshima (1999). Analog MOS circuit systems performing the visual tracking with bio-inspired simple networks. In *Proc. 7th Intl. Conf. Microelectronics for Neural, Fuzzy, and Bio-inspired Systems*, pp. 240–246.

Batista, Jorge (2000). A focusing-by-vergence system controlled by retinal motion disparity. In *Proc. ICRA*, pp. 3209–3214.

Batista, Jorge, Paulo Peixoto, and Helder Arauho (1996). Real-time visual behaviors with a binocular active vision system. In *Proc. IEEE/SICE/RSJ Intl. Conf. Multisensor Fusion and Integration for Intelligent Systems*, pp. 663–670.

Batista, Jorge, Paulo Peixoto, and Helder Arauho (1997). Real-time vergence and binocular gaze control. In *Proc. IROS*, pp. 1348–1354.

Bernardino, Alexandra and Jose Santos-Victor (1996). Vergence control for robotic heads using log-polar images. In *Proc IROS. 1996*, pp. 1264–1271.

Boahen, Kwabena A. (1999). Point-to-point connectivity between neuromorphic chips using address events. *IEEE Trans. on Circuit and Systems* pp. 100–117.

Brooks, Rodney (1986). A robust layered control system for a mobile robot. In *Cambrian Intelligence*. MIT Press, Cambridge, MA.

Chen, Tieh-Yuh, William N. Klarquist, and Alan C. Bovik (1994). Stereo vision using Gabor wavelets. In *Proc. IEEE Southwest Symp. Image Analysis and Interpretation*, pp. 13–17.

Cova, A. C. and H. L. Galiana (1994). A bilateral model of vergence nystagmus. In *Proc. 16th Annual Intl. Conf. IEEE*, pp. 263–264.

Cova, A. C. and H. L. Galiana (1995). Bilateral control of vergence and accommodation. In *IEEE-EBMC 17th Annual Conf.*, Vol. 2, pp. 1453–1454.

Cozzi, Alex, Bruno Crespi, Franco Valentinotti, and Florentin Worgotter (1997). Performance of phase-based algorithms for disparity estimation. *Machine Vision and Applications* pp. 334–340.

Encyclopedia Britannica Online (2000). Eye disease: Squint `http://www.britannica.com/bcom/eb/article/7/0,5716,117507+7+109528,00.html`. No author listed.

Erten, Gamze and Rodney M. Goodman (1996). Analog VLSI implementation for stereo correspondence between 2-d images. *IEEE Trans. Neural Networks* 7(2): 266–277.

Etienne-Cummings, Ralph, Viktor Gruev, and Cai Donghui (1999). A high density focal-plane image processing array. In *Proc. 33rd Conf. Information Sciences and Systems*, pp. 866–870.

Grigo, Antie and Markus Lappe (1998). Interaction of stereo vision and optic flow processing revealed by an illusory stimulus. *Vision Research* 38(2): 281–290.

Hansen, Michael and Gerald Sommer (1996). Active depth estimation with gaze and vergence control using Gabor filters. In *Proc. ICPR*, pp. 287–290.

Haralick, Robert M. and Linda G. Shapiro (1992). *Computer and Robot Vision Volume 1.* Addison-Wesley Publishing Company, Inc., Reading, MA.

Higgins, C. M. and C. Koch (2000). A modular multi-chip neuromorphic architecture for real-time visual motion processing. *Analog Integrated Circuits and Signal Processing* 24(3).

Higgins, C.M. and S.K. Korrapati (2000). An analog VLSI motion energy sensor based on the Adelson-Bergen algorithm. In *Proceedings of the International Symposium on Biologically-Inspired Systems.*

Higgins, C.M. and S.A. Shams (2000). A multi-chip neuromorphic architecture for spatial motion integration. In preparation.

Howard, Ian P., Robert S. Allison, and James E. Zacher (1997). The dynamics of vertical vergence. *Exp. Brain Res.* 116: 153–159.

Hung, George K. (1998). Dynamic model of the eye movement system: Simulations using MATLAB/SIMULINK. *Computer Methods and Programs in Biomedicine* 58: 59–68.

Hung, George K., Huimin Zhu, and Kenneth J. Ciuffreda (1997). Convergence and divergence exhibit different response characteristics to symmetric stimuli. *Vision Research* 37(9): 1197–1205.

Jiang, B. C. (1996). Accommodative vergence is driven by the phasic component of the accommodative controller. *Vision Research* 36(1): 97–102.

Kandel, Eric R., James H. SchwarZ, and Thomas M. Jessell (1995). *Essentials of Neuroscience and Behavior.* Appleton and Lange, Norwalk, CT.

Knight, Joss and Ian Reid (2000). Active visual alignment of a mobile stereo camera platform. In *Proc. ICRA*, pp. 3203–3208.

Kolesnik, Marina and Gregory Baratoff (2000). 3-d interpretation of sewer circular structures. In *Proc. ICRA*, pp. 1453–1458.

Lande, Sverre Tor, editor (1998). *Neuromorphic Systems Engineering Neural Networks in Silicon.* Kluwer Academic Publishers.

Lu, Ziyi and Bertram E. Shi (2000). Visual tracking with subpixel resolution using an analog VLSI computational sensor. In *Proc. International Conference on Robotics and Automation*, pp. 1676–1681.

Mahowald, M. and T. Delbrück (1989). Cooperative stereo matching using static and dynamic image features. In Mead, C. and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pp. 213–238. Kluwer Academic Publishers.

Mallot, Hanspeter A., Anke Roll, and Petra A. Arndt (1996). Disparity-evoked vergence is driven by interocular correlation. *Vision Research* 36(18): 2925–2937.

Marefat, Michael M, Liwei Wu, and Christopher C. Yang (1997a). Gaze stabilization in active vision - I. vergence error extraction. *Pattern Recognition* 30(11): 1829–1842.

Marefat, Michael M, Liwei Wu, and Christopher C. Yang (1997b). Gaze stabilization in active vision - II. mulit-rate vergence control. *Pattern Recognition* 30(11): 1843–1853.

Marr, D. and T. Poggio (1976). Cooperative computation of stereo disparity. *Science* 194: 283–287.

Marshall, Jonathan M., George J. Kalarickal, and Elizabeth B. Graves (1996). Neural model of visual stereomatching: Slant, transparency, and clouds. *Computation in Neural Systems* 7: 635–669.

Intel Corp (2000). *Pentium III Processor for the SC242 at 450 MHz to 1.13 GHz Datasheet*. Intel Corp., Santa Clara, CA. Document Order Number 244452-008.

Mead, Carver A. (1989). *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA.

Ohzawa, I., G.C. DeAngelis, and R.D. Freeman (1997). Encoding of binocular disparity by complex cells in the cat's visual cortex. *J. Neurophysiology* 76(6): 2879–2909.

Ohzawa, I., Gregory C. DeAngelis, and Ralph D. Freeman (1990). Stereoscopic depth discrimination in the visual cortex: Neurons ideally suited as disparity detectors. *Science* 249: 1037–1041.

Ohzawa, I., Gregory C. DeAngelis, and Ralph D. Freeman (1996). Encoding of binocular disparity by simple cells in the cat's visual cortex. *J. Neurophysiology* 75(5): 1779–1805.

Ohzawa, Izumi and Ralph D. Freeman (1986). The binocular organization of simple cells in the cat's visual cortex. *J. Neurophysiology* 56: 221–241.

Ohzawa, Izumi and Ralph D. Freeman (1990). On the neurophysiological organization of binocular vision. *Vision Research* 30(11): 1661–1675.

Olson, Thomas J. and Robert D. Potter (1989). Real time vergence control. In *Proc. CVPR*, pp. 404–409.

Palmer, Steven E. (1999). *Vision Science*. The MIT Press, Cambridge, MA.

Patel, S.S., H. Ogmen, J.M. White, and B.C. Jiang (1997). Neural network model of short-term horizoontal disparity vergence dynamics. *Vision Research* 37(10): 1383–1399.

Piater, Justus H., Roderic A. Grupen, and Krithi Ramamrithan (1999). Learning real-time stereo vergence control. In *Proc. IEEE Int'l. Symp. Intelligent Control/ Intelligent Systems and Semiotics*, pp. 272–277.

Pinker, Steven (1997). *How the Mind Works*. W. W. Norton and Company, Inc., New York, N.Y.

Popple, Ariella V., Harvey S. Smallman, and John M. Findlay (1998). The area of spatial integration for initial horizontal disparity vergence. *Vision Research* 38(2): 319–326.

Qian, Ning (1994). Computing stereo disparity and motion with known binocular cell properties. *Neural Computation* 6: 390–404.

Qian, Ning (1997). Binocular disparity and the perception of depth. *Neuron* 18: 359–368.

Qian, Ning (2000). Relationship between phase and energy methods for disparity computation. *Neural Computation* 12: 303–316.

Qian, Ning and Yudong Zhu (1997). Physiological computation of binocular disparity. *Vision Research* 37(13): 1811–1827.

Sanger, T.D. (1988). Stereo disparity computation using Gabor filters. *Biological Cybernetics* 59: 405–418.

Sciencenet.org (2000). Senses and behavior: What is binocular vision and why is it important? `http://www.sciencenet.org.uk/database/social/senses/s0012b.html`. No author listed.

Semmlow, John L., Weihong Yuan, and Tara L. Alvarez (1998). Evidence for separate control of slow vergence eye movements: Support for Hering's law. *Vision Research* 38(8): 1145–1152.

Shi, Bertram E. (1999). Focal plane implementation of 2d steerable and scalable Gabor-type filters. *J. VLSI Signal Processing* 23: 319–334.

Stevenson, Paul B, Paul E. Reed, and Jiang Yuang (1999). The effect of target size and eccentricity on reflex disparity. *Vision Research* 39: 823–832.

Woods, Andrew, Tom Docherty, and Rolf Koch (1993). Image distortions in stereoscopic video systems. In *Proc. SPIE Stereoscopic Displays and Applications IV*, Vol. 1915.

Yim, Changoon and Alan C. Bovik (1994). Vergence control using a heirarchical image structure. In *Proc. IEEE Southwest Symp. Image Analysis and Interpretation*, pp. 118–123.

Zigmond, Michael J., Floyd E. Bloom, Story C. Landis, James L. Roberts, and Larry R. Squire (1999). *Fundamental Neuroscience*. Academic Press, San Diego, CA.