Real-Time Vergence Control for Binocular Robots

Thomas J. Olson David J. Coombs

Technical Report 348
June 1990

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

Real-Time Vergence Control for Binocular Robots

Thomas J. Olson*
olson@cs.virginia.edu

David J. Coombs
coombs@cs.rochester.edu

The University of Rochester
Computer Science Department
Rochester, New York 14627

Technical Report 348

June 1990

Abstract

In binocular systems, *vergence* is the process of adjusting the angle between the eyes (or cameras) so that both eyes are directed at the same world point. Its utility is most obvious for foveate systems such as the human visual system, but it is a useful strategy for non-foveate binocular robots as well. This paper discusses the vergence problem and outlines a general approach to vergence control, consisting of a control loop driven by an algorithm that estimates the vergence error. As a case study, this approach is used to verge the eyes of the Rochester Robot in real time. Vergence error is estimated with the cepstral disparity filter. The cepstral filter is analyzed, and it is shown in this application to be equivalent to correlation with an adaptive prefilter; carrying this idea to its logical conclusion converts the cepstral filter into phase correlation. The demonstration system uses a PD controller in cascade with the error estimator. An efficient real-time implementation of the error estimator is discussed, and empirical measurements of the performance of both the disparity estimator and the overall system are presented.

Keywords: vergence, gaze control, binocular vision, disparity, cepstrum.

This material is based on work supported by the U.S. Army Engineering Topographic Laboratories under research contract no. DACA76-85-C-0001, by NSF research grants nos. DCR-8602958 and IRI-8903582, by NIH Public Health Service research grant no. 1 R01 NS22407-01, and by ONR research contract no. N00014-82-K-0193. In addition, this work was supported by the NSF under Institutional Infrastructure grant CDA-8822724, and by a Dean's Research Initiation grant from the School of Engineering and Applied Science of the University of Virginia. The government has certain rights in this material.

*Thomas Olson is with the Department of Computer Science, University of Virginia, Charlottesville, Virginia 22903.

1 Introduction

Recently a significant amount of work in computer vision has focused on the problems of acting, behaving systems, and in particular on how "active vision" differs from analysis of static scenes or vision with fixed cameras [Bajcsy, 1986; Aloimonos *et al.*, 1987; Ballard, 1989; Bandopadhay, 1986]. In many cases, giving a vision system the ability to move around in its environment simplifies many previously intractable problems. Since the summer of 1988 the Rochester vision group has been working to develop an integrated facility for the study of vision, AI and systems issues related to active vision. Briefly, the facility consists of an industrial robot arm bearing a custom-built "head". The head has two CCD television cameras which can be moved together in altitude (pitch) and independently in azimuth (yaw). The head, arm and cameras are connected to a pipelined image processor, a workstation and a set of large-scale parallel processors.

A major goal of our research is the development of a real-time gaze control system. We believe that the robot must be able to maintain fixation on world points or change fixation from one world point to another with only minimal direction from high level "cognitive" faculties. To this end, we are developing quasi-reflexive gaze control mechanisms that maintain fixation on smoothly moving targets while compensating for egomotion, and make saccadic movements to targets selected by higher level processes. We envision the gaze control mechanisms forming a layered, modular control structure along the lines described by Brooks [Brooks, 1986; Brooks, 1987], although we suspect that more sensor fusion may be required than has been employed in systems of this type in the past. The details of the control structure and module interactions are a current research topic [Brown, 1990b; Brown, 1990a; Brown, 1990c; Coombs, 1989a], but preliminary work has identified some promising approaches to the various subproblems [Brown *et al.*, 1988].

This paper describes the design, implementation and performance of a module responsible for controlling the vergence angle of the cameras. The next section discusses vergence in the abstract, presenting reasons for verging and issues that any vergence control system must address. This discussion leads to a general strategy for vergence control, described in Section 3. Sections 4, 5 and 6 describe the application of this vergence control strategy to the problem of controlling vergence on the Rochester Robot, and present empirical results on the performance of the error estimator and the overall vergence system.

2 The Vergence Problem

The *vergence angle* of a binocular system is the angle between the optic axes of its eyes or cameras. The vergence angle, baseline (or inter-ocular distance) and gaze direction of a binocular system determine a particular fixation point, as shown in figure 1. Narrowly speaking, the function of the vergence system is to control the distance from the cameras to the fixation point along some specified gaze direction. In most cases the motivation for vergence is to keep the fixation point near some target object. Thus, the vergence problem can be defined as that of controlling the vergence angle to keep the fixation depth appropriate for the current gaze target. Since the target vergence angle is directly related to target depth, any sensory cue to depth or depth changes may be useful to the vergence

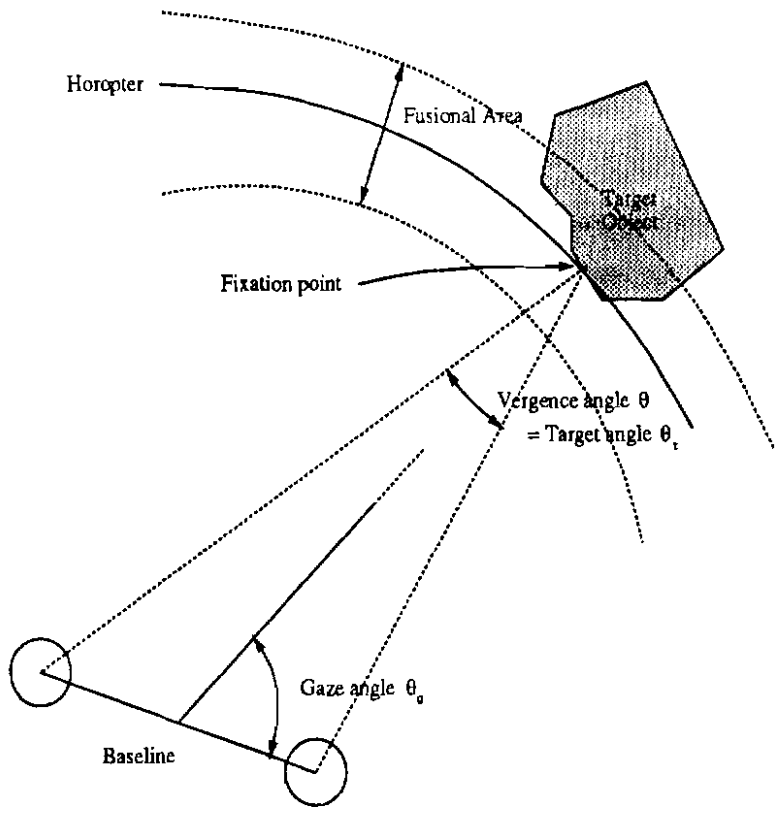


Figure 1: The goal of vergence is to keep the eyes or cameras fixed on a common world point or visual target, independent of changes in gaze angle and target distance. The distance to the world point, the length of the baseline (or inter-ocular distance) and the gaze angle θ_g combine to produce a desired vergence angle θ_t. In order to keep the world point fixated, the vergence system must generate an actual vergence angle matching the desired angle. The result of fixating a target is that the object lies near the *horopter*, which is the set of world points whose disparity is zero. The stereo images of an object that lies near the horopter have a narrow range of disparities.

system. The most commonly used cues are disparity and focus error, but other depth cues (such as motion, texture, shading, *etc.*) can also be used, as can information about depth changes (measured or predicted self motions, dilations or contractions of the visual field, and so on).

Vergence is one aspect of the larger problem of gaze control, which involves control of the gaze angle and focal depth as well. The larger problem can be broken down functionally into the subproblems of *gaze stabilization* and *gaze shift*. Stabilization involves maintaining fixation on a possibly moving visual target from a possibly moving gaze platform. Gaze shifts, usually called *saccades*, transfer fixation from one visual target to another. Vergence control must meet different demands in each of these activities. During stabilization, a change in the target position relative to the observer produces a smooth change in the desired vergence angle. A saccade transfers the fixation point almost instantaneously from

one visual target to another, producing a step change in the desired vergence angle.

The treatment of vergence presented here reflects current models of vergence in primates. Primate visual systems exhibit sophisticated vergence responses that meet the varied demands of the vergence task. Recent experiments have challenged traditionally held views (*e.g.*, those of Yarbus [Yarbus, 1967]) of ocular vergence and its control. It has been thought that vergence changes are always smooth and much slower than smooth pursuit (tracking) movements, and that vergence changes required to shift gaze to a target are achieved by smooth vergence movements superimposed on conjugate (equal and symmetric) saccades in the two eyes. Under natural viewing conditions, however, Erkelens *et al.* [Erkelens *et al.*, 1989a; Erkelens *et al.*, 1989b] observed in humans not only smooth vergence movements rivaling the speeds of other smooth eye movements, but also vergence changes mediated almost entirely by saccades that incorporated a vergence change explicitly, rather than merely superimposing symmetric saccades on smooth vergence movements. Similar behaviors have also been observed in monkeys [Maxwell and King, 1990].

2.1 Related Work

The recent surge of interest in active vision has produced a growing body of literature on vergence and gaze control for robotic vision systems. Clark and Ferrier [Clark and Ferrier, 1988] built a gaze control system based on the model described in [Robinson, 1987]. The system acquires and tracks white and black blobs using the first few moments and intensity value of each object. The mechanical design of their head decouples the control of the gaze and vergence angles. The gaze angle is controlled by rotating the head about its neck, and the cameras are verged symmetrically by a mechanical linkage. This aspect of the design prevents intermingling the control of vergence and gaze angles, and vergence control follows the model of Yarbus. Vergence has recently been used cooperatively with focus and stereopsis for surface reconstruction [Abbot and Ahuja, 1988] and active exploration of the environment [Krotkov, 1989]. It has demonstrated advantages in both robustness of results and increased speed in stereoscopic processing.

2.2 Why Verge?

Part of the motivation for studying vergence comes from an interest in human vision: human eyes verge, and we would like to know more about how they do it. The human visual system's need for a vergence control system is obvious, and follows from the extremely non-uniform spatial resolution of the photoreceptor array. Vergence movements allow humans to register an object of interest on the fovea (central, high-resolution region of the retina) of each eye, so that the greatest possible amount of information about the object can be extracted.

Currently most robot vision systems do not have foveas, and so the most obvious motivation for vergence control in humans does not apply to them. However, vergence has many advantages even for systems without foveas.

Mathematical Simplification: Fixating an object of interest puts points on the object near the optic axis in both eyes. In some cases this permits the use of simplifying assumptions (*e.g.* replacing perspective projection with orthography) that make

analysis significantly easier. For example, Ballard and Ozcandarli [Ballard and Ozcandarli, 1988] used this fact to develop a simple and efficient kinetic depth estimator for systems that fixate.

Facilitating Stereo Fusion: By definition, the fixation point has a stereoscopic disparity of zero, and points nearby tend to have small disparities. This makes it possible to use stereo algorithms that accept only a limited range of disparities. Such systems can be very fast, and are amenable to hardware implementation [Mead and Mahowald, 1988]. Olson is currently working on a stereo system that has high spatial resolution at small disparities and lower resolution elsewhere [Olson, 1990]. This allows the system to devote the bulk of its resources to areas of interest without losing track of the rest of the scene.

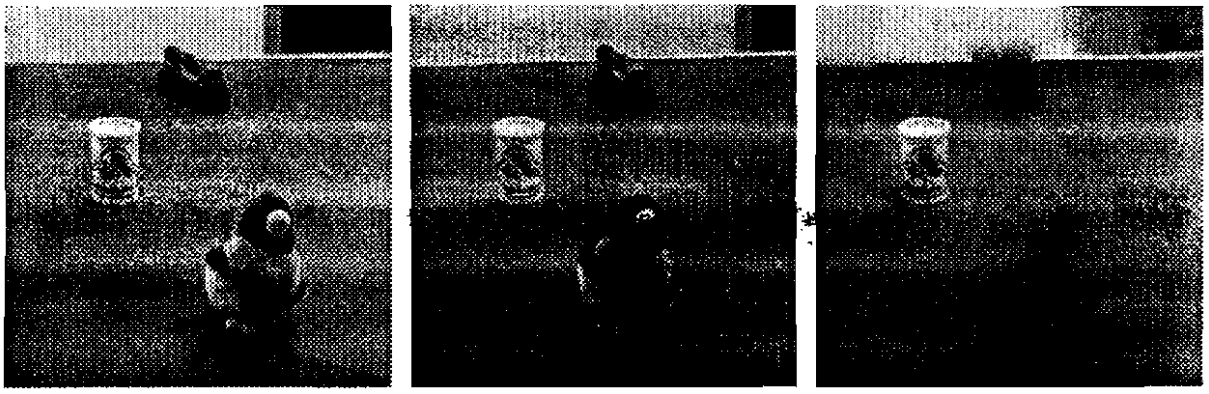
Useful Coordinate Systems: As Ballard argues [Ballard, 1989], having a unique fixation point at the intersection of the visual axes defines a coordinate system that is related as much to the object being observed as it is to the observer. It is thus a step in the direction of an object-centered coordinate system.

Disparity-based Segmentation: On the assumption that gaze will normally be directed toward objects of interest, it may be appropriate for binocular agents to ignore features at large disparities. That is, disparity may be used to filter objects that are not currently of interest out of the scene. Figure 2 shows an example of this sort of filtering. There is some evidence that biological visual systems filter images precategorically using disparity information. The gain of the optokinetic effect seems to be modulated by disparity [Howard and Simpson, 1989], and Miles *et al.* [Miles *et al.*, 1990] have proposed that primates might use disparity information to help parse optical flow.

An argument can be made for the ultimate necessity of non-uniform resolution, in order to provide both high resolution and a wide field of view [Tsotsos, 1987]. Thus future robot systems may be equiped with foveas. If so they will require vergence systems for the same reasons that humans require them. Work on spatially-variant visual sensors is beginning [Van der Spiegel *et al.*, 1989; Tistarelli and Sandini, 1990], so it may become possible to build camera systems with foveas in the near future.

3 A General Strategy for Vergence Control

At the most abstract level, any solution to the vergence problem will have three major components, as shown in Figure 3: a *sensory system* that determines how the current vergence angle differs from the ideal, a *controller* that generates a response to the errors, and a *motor system* that executes the controller's commands. These three components can be mapped onto the traditional block diagram of a feedback system, shown in Figure 4. The input to the system is the desired vergence angle, θ_t. The error estimator and cameras or other sensors correspond to the summation unit in the block diagram. Their function is to compare the actual vergence angle θ to the target angle θ_t to produce a residual error θ_e. The controller converts that error to a set of control signals that direct the camera



a) b) c)

Figure 2: Using disparity to filter a scene. Images a) and b) above show right and left camera views of a scene containing objects at several disparities. These frames were factored into laplacian pyramids [Burt and Adelson, 1983], after which each pixel in the left image pyramid was rescaled by a factor of one minus the normalized difference between it and the corresponding pixel in the right pyramid. Reassembling the left pyramid produced image c). Since pixels in a laplacian pyramid represent corrections to values inferred from lower spatial frequency information, the effect of the filter is to suppress high frequency information associated with objects at high disparities.

motors (*i.e.*, the plant) to produce appropriate changes in θ. This section discusses general considerations in the design and use of these three components.

3.1 Motor System

The quality of the motor system or plant is determined by how quickly and faithfully it translates control signals into changes in vergence angle. Current generation CCD cameras and motor controllers make it relatively easy to move the cameras quickly. Care is required, however, to insure that the camera mounting is able to tolerate the stresses generated by rapid eye movements. The large accelerations required for saccadic movements can cause "ringing", *i.e.* vibrations that persist after the motors have come to a stop. Avoiding these problems involves mechanical engineering considerations that are beyond the scope of this paper, so we will not discuss them further.

Another aspect of motor system design that is important for vergence control is the number of degrees of freedom offered by the hardware—that is, what parameters of the camera position are controllable. A number of systems in current use (*e.g.* [Clark and Ferrier, 1988; Krotkov, 1989]) constrain the gaze angle to be at right angles to the baseline. In this type of system vergence angle is controlled by a single motor that converges both cameras symmetrically via a mechanical linkage, such as a rack and pinion driving a pair of levers that rotates the cameras about vertical axes, as sketched in Figure 5. Gaze angle can be altered by rotating the entire system about vertical and horizontal axes through the center of the baseline. The advantage of this design is that gaze angle and vergence angle are controlled by separate motors and are orthogonal—either parameter can be altered without

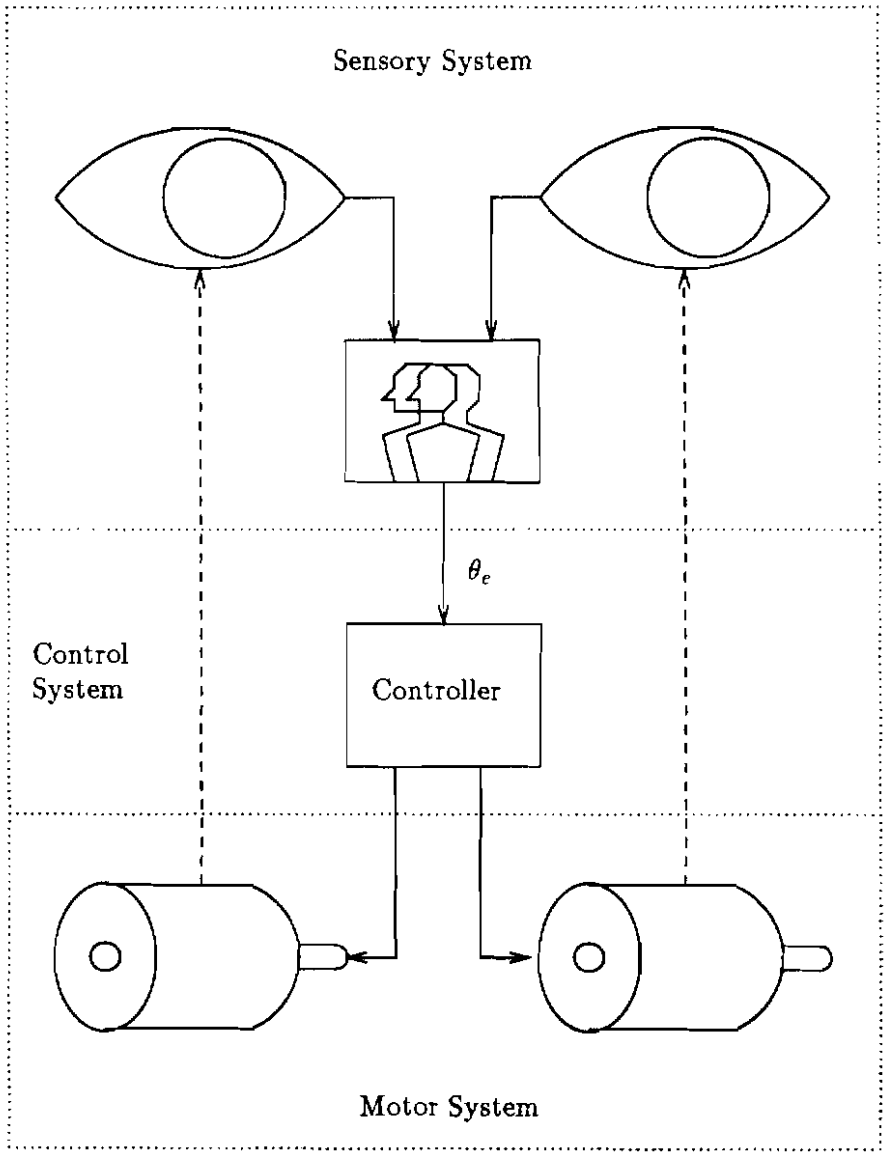
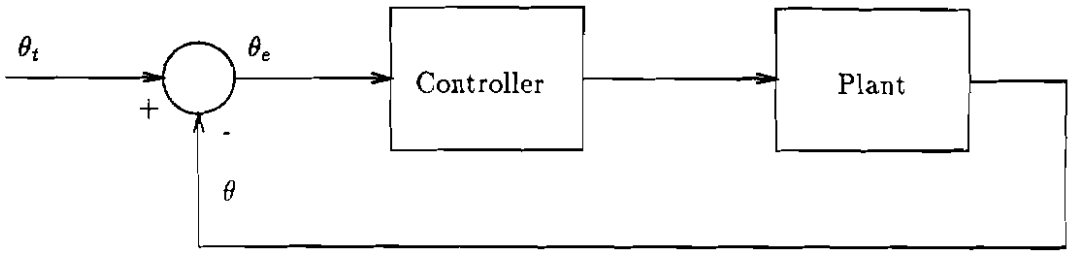


Figure 3: Schematic diagram of the vergence system.



θ_t — target vergence angle
θ — vergence angle
θ_e — vergence error

Figure 4: Block diagram of the vergence system.

disturbing the other. This makes them well suited to traditional models of gaze control, in which vergence is considered to be largely independent of other types of eye movements.

An alternative design mounts the cameras together on a platform that can be tilted up and down, and allows each camera to pan from side to side independently (*e.g.*, the Rochester Robot [Brown *et al.*, 1988], sketched in Figure 6). With this design, the gaze angle and vergence angle are no longer independent. Vergence angle is equal to the difference between the two camera angles, and gaze angle is a function of both camera angles. This design lends itself well to systems in which vergence control is tightly integrated with other types of eye movements. It would also support systems that directly control the positions of each camera without explicitly controlling the vergence angle, allowing vergence to emerge from the eye positions. A mechanical advantage of this design is its simplicity: the compact mechanism and fairly direct linkages facilitate rapid saccades.

Another important aspect of the mechanical design is the relation of the axes of rotation to the nodal points of the cameras. The nodal point of a camera is the point about which a camera rotation results in a pure translation of the image projected on its image plane. It corresponds to the pinhole of a pinhole camera, or to the front nodal point of an ideal thick lens system [Horn, 1986]. The image translation induced by rotation about the nodal point is a function only of the projected image and the rotation, and is independent of the depth of the object being imaged. For the purposes of gaze control it is desirable to mount the cameras so that their axes of rotation pass through the nodal points. Doing so makes it possible to predict the effects of a rotation without knowing the depths of objects in the scene.

If the axes of rotation do not pass through the nodal point of the camera system, each camera movement necessarily includes a small translational component as well as the desired rotation. For example, changing the vergence angle will alter the baseline of the camera system, complicating the depth computation for large disparities. Unfortunately, designing a system that rotates the cameras about their nodal points is difficult. The camera pivots may be far from the camera's center of gravity, and the nodal point moves when the lenses are changed or moved to adjust focus. However, the nodal point does not move very far, and the distortion induced can often be ignored.

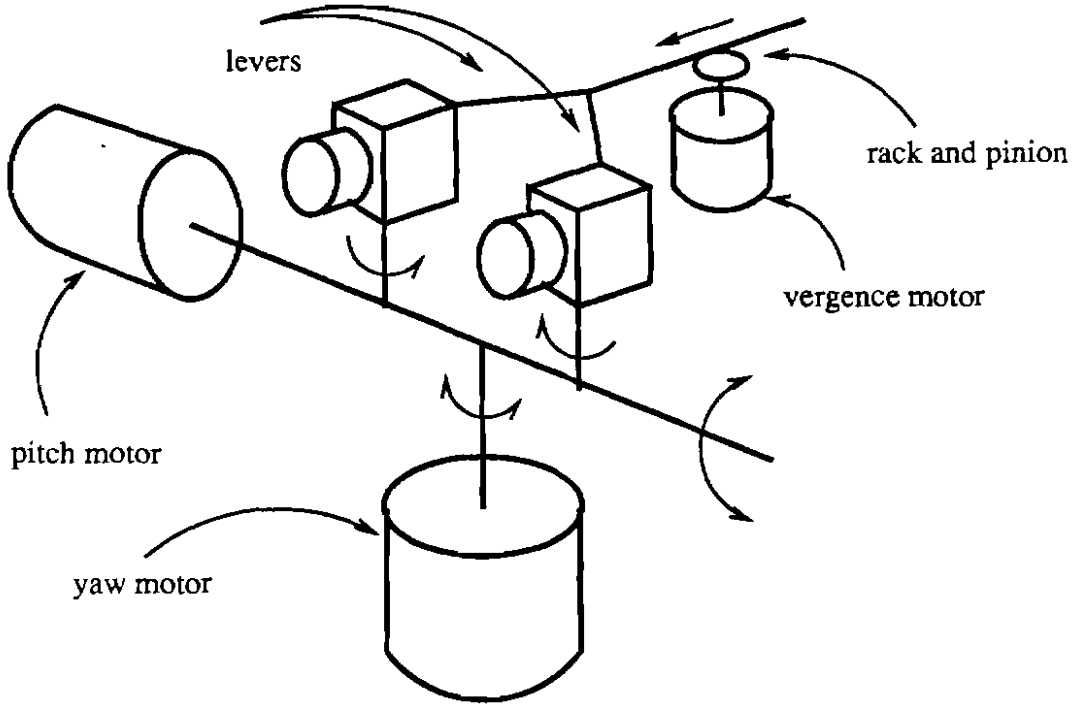


Figure 5: Independent gaze and vergence controls are provided by separate motor systems. (This sketch is designed for illustrative purposes rather than compactness of mechanism.)

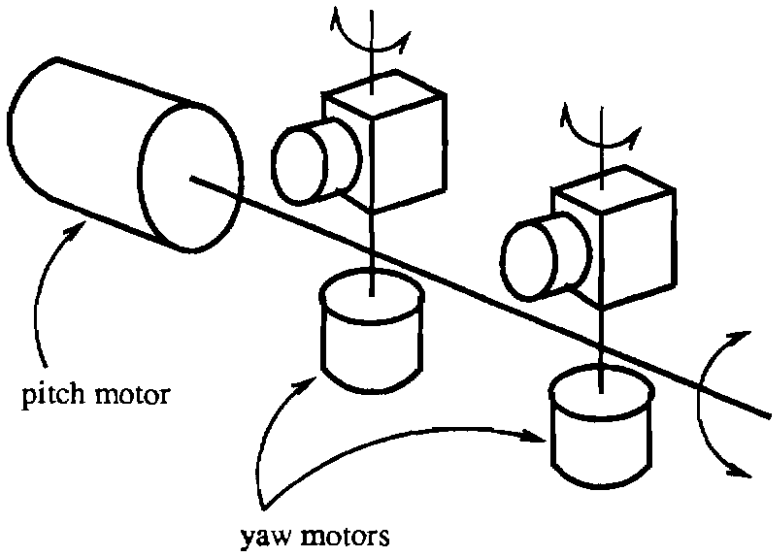


Figure 6: Gaze and vergence controls are combined by this design.

More sophisticated binocular systems may have several other control parameters to consider. One is focus depth. Interactions between focus depth and vergence angle have been explored recently in robotic vision experiments [Abbot and Ahuja, 1988; Krotkov, 1989]. Another possible degree of freedom is torsion, *i.e.* rotation about the optic axis. Torsional movements can be used to stabilize gaze against head rotations about an axis parallel to the gaze direction. Torsional stabilization in humans can be readily observed by watching one's eyes in a mirror while rolling the head from side to side. Torsional movements also serve to align the vertical axes of the cameras when gaze is directed up or down from the horizontal plane. An example of such a misalignment can be seen in Figures 2a and 2b: the two images of the can are slightly tilted with respect to one another. This type of misalignment cannot be corrected in the Rochester Robot since its head has no torsional degree of freedom. Torsion control and measurement of torsion error provide information that can be used to help judge the tilt of an object toward or away from the observer [Mayhew and Longuet-Higgins, 1982].

3.2 Controller

A critical parameter in the design of a vergence control system is the nature of the input signal – how the target angle θ_t changes with time. This will depend to some degree on details of the system design, particularly the nature of the processes responsible for other aspects of gaze control and movement. In the absence of detailed information about these processes, it seems reasonable to base our expectations on the known characteristics of human eye movements [Yarbus, 1967], and on the general view of gaze control described in Section 2. That is, we expect eye movements to consist of intervals of smooth pursuit or fixation punctuated by discontinuous jumps (saccades).

The two types of expected changes in target angle differ in fundamental ways. During pursuit and fixation, changes in target angle are determined by the dynamics of observer and object motion. The laws of physics restrict what can happen; for example, accelerations and velocities must be finite. Furthermore, although a target may cross the visual field with high velocity, rapid changes in target vergence angle will be rare. Rapid changes in target angle correspond to very rapid movements in depth (especially near the observer), so the target will soon pass through the image plane (if it is approaching) or recede to a depth at which target angle changes more slowly (if it is moving away from the observer.) In short, the input to the control loop during pursuit and fixation will be smooth, with finite second derivatives and small first derivatives.

During a saccade the input signal will behave quite differently. A saccade can produce a step change in the desired vergence angle, as well as a discontinuity in its temporal derivative. The magnitude of these changes may be predictable if the saccade is to a previously visited target, or if target depth and relative motion are approximately known from other depth cues. At the very least, the fact that a saccade is occurring can serve as a warning that discontinuities in the input signal are to be expected.

The fact that there are two distinct types of changes in desired vergence angle suggests a need for two modes of control. The normal operating mode should be optimized for the smooth, continuous changes expected during pursuit and fixation. Saccadic movements

should replace the smooth movements of the normal control loop with brief intervals of what is sometimes called "bang-bang" control. That is, the estimated error should be corrected by an open-loop move to the new desired vergence angle at the maximum rate of which the motor system is capable. The open-loop vergence correction can be performed during the saccade, if the distance to the saccade target is known; if not there will be some delay while the error estimator determines a new target angle. The error estimator may need to be suppressed during a saccade, because of the possibility that motion blur and/or shearing deformations (caused by camera movements during a single video frame interval) may corrupt the results. Before the normal control loop is restarted, it should be reinitialized to prevent any tendency to smooth target angle velocity across the saccade. The details of how this is done will of course depend on the natures of the smooth control loop, the underlying hardware and the saccade generating process.

3.3 Error Estimator

In order to keep the eyes verged on a target, the vergence system must measure the current vergence error (and, perhaps, its derivatives.) The most important source of this information is the visual system, but other sources may also be useful. We have already noted the possibility of predicting the error that will result from a saccade to a target of known depth. Vergence changes due to self motion can also be taken into account, either by making predictions based on planned, voluntary head movements, or by sensing head accelerations via the vestibular system (as in the human vestibulo-ocular reflex.) However, vision is the only source of information for target motion, and visual cues also provide the ultimate measure of vergence performance. The rest of this section, therefore, is restricted to consideration of visual error estimators.

A number of different types of visual information are available for estimating vergence error. One feature that is correlated with desired vergence angle under ordinary conditions is blur, which has been used cooperatively with vergence and stereo to construct depth maps [Abbot and Ahuja, 1988; Krotkov, 1989]. Any depth cue can be used if the absolute vergence angle of the system is known, because desired vergence angle is a function of target distance. Humans apparently make use of cues that may indicate change in depth, since changes in the size of a visual target induce transient vergence responses [Erkelens and Regan, 1984].

The most useful visual cue to vergence error, however, is binocular disparity. The mapping from disparity to vergence error is particularly simple, and (unlike monocular depth cues) does not require knowledge of the absolute vergence angle of the system. Reliable disparity estimates can be computed more easily and quickly than depth estimates, permitting shorter processing delays and simpler control strategies. These advantages may be reflected in the structure of the human vergence control system; although vergence in humans can be driven by a variety of cues, responses are much slower under monocular viewing conditions than they are when disparity information is available [Erkelens *et al.*, 1989b].

Disparity measurement has been studied extensively in the context of stereo depth reconstruction [Barnard and Fischler, 1982]. Unfortunately most of the disparity estimators used for stereopsis are poorly suited to the real-time vergence application. They are optimized

for positional accuracy and density rather than for robustness, and depend on optimization of global criteria to yield a more robust disparity field. They cannot provide single disparity estimates without essentially solving the stereo problem, which entails considerable computational expense.

For real-time vergence what is needed is a simple algorithm that estimates a single disparity in a fixed amount of time. This narrows the field to image processing methods such as cross-correlation. Past attempts to use such methods for stereo depth recovery have uncovered many problems (see [Horn, 1986] for a review). However, a class of operators that are closely related to correlation appears to work quite well for vergence. These operators are described in Section 5 and the appendices. If correlation methods prove too slow, related methods such as phase comparison [Jepson and Jenkin, 1989] may be suitable, provided that care is taken to detect and compensate for various predictable errors [Fleet *et al.*, 1989].

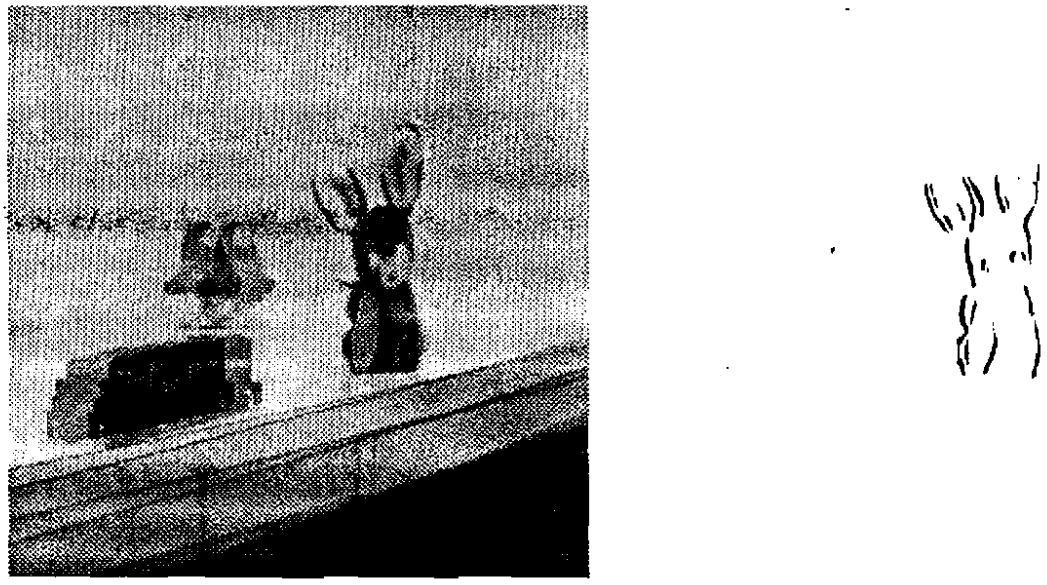
Handling Multiple Disparities

An important issue in correlation-based disparity estimation for vergence concerns the handling of scenes with multiple disparities. It is tightly linked to the selection of sample window size. The sample windows must be large enough to handle the expected range of disparities, but this almost guarantees that multiple disparities will be present at least some of the time. Therefore, some additional processing will almost certainly be required to insure that the desired correlation peak dominates the output.

The success or failure of a vergence system is determined by how well it maintains fixation on a designated target object. Therefore, an obvious way to improve the performance of the error estimator is to filter the input images so that the target object is more prominent. If detailed information about the target location is available, the image can be multiplied by a mask that emphasizes details near that location. A simpler approach is to let the target be designated implicitly by one of the eyes. That is, consider one of the eyes to be dominant, and define the region near its optic axis to be the target. This strategy requires only that the dominant eye image be multiplied by a centrally weighted mask before correlating.

Another masking strategy arises from disparity-based segmentation. First, the two sample windows are processed by a derivative operator in the horizontal direction, producing a pair of vertical edge images. The images are then combined by a multiplicative or 'ANDing' operator that attenuates pixels that are weak in either image and amplifies pixels that are strong in both images. This produces an image containing only edges that appear at the same location in both images. Thus, this zero-disparity filter passes only edges that lie in the *horopter*, or "shell" of zero disparity (illustrated in figure 1) plus some possible aliased (accidentally aligned) edges. Figure 7 demonstrates the effect of this filter on a real scene. The zero-disparity image is then blurred and used as a mask, preemphasizing points that are at small disparities, as proposed in [Coombs, 1989b].

Another strategy is to perform the ANDing operation several times, incorporating small rightward and leftward shifts of one of the images. This produces a series of images that contain edges on several closely spaced pseudo-horopters. Counting the number of 'on' pixels in each depth plane gives an estimate of the average disparity of pixels near the horopter, which can be used to derive a vergence error estimate.



a) b)

Figure 7: Image a) shows images from the two cameras superimposed by pixelwise intensity averaging. The left and right camera views were processed with a vertically oriented Sobel operator to produce a pair of stereo images of vertical edges. These images were combined by a pixelwise multiplicative 'AND' operator to produce image b). The multiplicative operator has the effect of attenuating pixels that are weak in either image and amplifying pixels that are strong in both images. It thus tends to suppress edges that have non-zero disparity, leaving an edge image that is dominated by objects at the horopter.

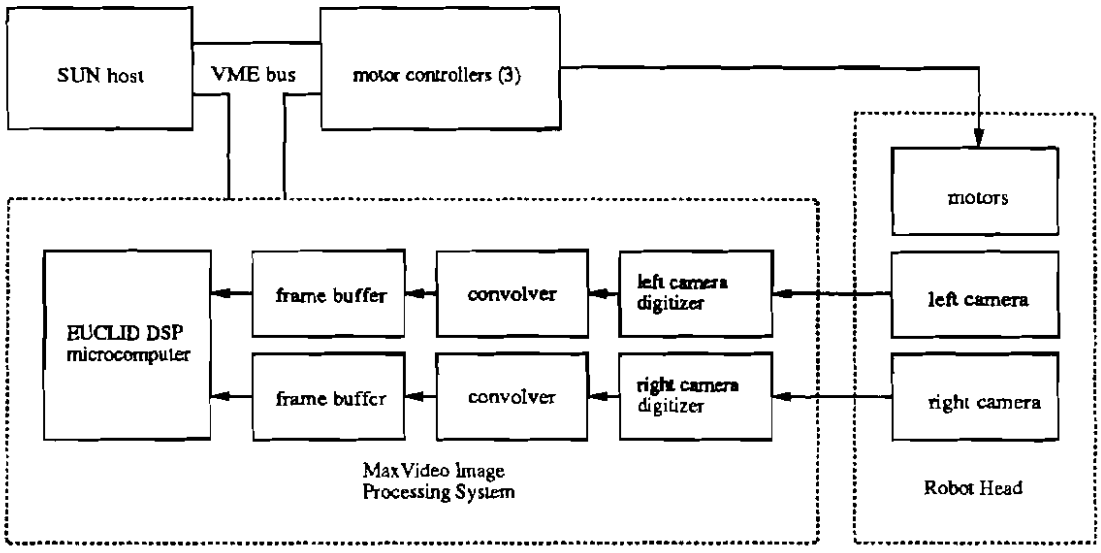


Figure 8: Hardware configuration used for vergence control experiments on the Rochester Robot.

These disparity filters can be performed at video rates with relatively inexpensive hardware. Moreover, the combination operator (as well as the edge operator) can be designed to accentuate or suppress weaker edges, edges at particular orientations, *etc.*, to suit the desired application. For instance, a "zero-disparity" filter that is used as a mask to emphasize objects in the horopter for vergence should be broadly tuned to "thicken" the horopter, so objects will remain visible in the near vicinity of the depth of gaze. It should also pass only vertical edges since these supply the most information for a disparity estimator. However, a family of disparity filters that is used to derive a depth estimate might work better if each member filter is narrowly tuned.

4 Vergence on the Rochester Robot

The general considerations discussed in the preceding sections formed the basis for the vergence system used on the Rochester Robot. This section and the two that follow describe the motor, sensory and control components of the system, and discuss its performance as measured in the laboratory.

We begin by summarizing those aspects of the Rochester Robot's cameras, motor system, and computing resources that affected the design of the vergence system. A more detailed description of the robot and laboratory resources is given elsewhere in this issue. Figure 8 shows a block diagram of those parts of the system that are involved in vergence control.

As illustrated in Figure 6, each of the robot's two cameras is panned from side to side by its own motor. As discussed in Section 3, this type of system facilitates rapid saccades at the cost of some increased complexity in the control system, which arises from the non-orthogonality of the vergence and gaze angles. A third motor serves to pitch the cameras

up and down, *i.e.*, to rotate them about their baseline. These three motors can generate saccades with peak speeds of more than 400 degrees per second. Reduction gearing gives the cameras theoretical angular resolutions of 1/278 degree in yaw and 1/2500 degree in pitch. Gear lash reduces the resolution to an unknown degree, but camera positioning is still accurate to substantially better than the angle subtended by one pixel with the 16mm lenses that are normally used.

The mechanical design of the camera platform is such that the nodal points of the cameras do not lie precisely on the axes of rotation. This means that eye movements necessarily include a small translational component as well as the desired rotation, so that (for example) increasing the vergence angle slightly reduces the baseline of the camera system. For most purposes the translational movement can be ignored, although its effects were noticeable in the experiments described below.

The host computer for the robot commands the motors via intelligent stepping motor controllers that allow the control program to issue commands in terms of absolute position, relative position, velocity or velocity profile. The ability to issue buffered velocity commands enables the control program to generate smooth movements without paying constant attention to the motors.

The EUCLID digital signal processing microcomputer included in the MaxVideo^{TM}image processing system was used for estimating disparity. The EUCLID computer is based on the ADSP-2100 digital signal processor [Analog Devices, 1987], which is optimized for operations such as convolution, finite impulse response filtering and Fast Fourier Transforms.

The mechanical design of the motor and camera system was dictated by a desire to perform saccadic movements at speeds comparable to those of humans. It seemed probable that a camera platform powerful and rigid enough to perform saccades quickly and without noticeable ringing would be able to handle the gentler movements required for vergence with relative ease. To date this has proven to be the case, and the performance of the vergence system has been limited by the speed and accuracy of the error estimator rather than the capabilities of the motor system.

5 Error Estimation

The vergence error estimator is based on disparity, since (as argued in Section 3) disparity is the most direct and reliable measure of vergence error. One approach to disparity estimation would have been to use the MaxVideo^{TM}convolution/correlation hardware to compare central patches of one image to the other image. However, previous attempts in our lab to use that approach for tracking had encountered many difficulties. Instead the disparity estimator, previously described in [Olson and Potter, 1989], is based on the cepstral filter [Yeshurun and Schwartz, 1989]. The cepstral estimator performed well from the beginning, but the reasons for its success were initially unclear. Our efforts to achieve a better understanding of the algorithm led us to an interpretation of the cepstral disparity estimator as one of a family of operators of which phase correlation [Kuglin and Hines, 1975] is a logical endpoint. This section describes the basic operation and performance of the cepstral disparity estimator. The reasons for its success and its relationship to phase correlation are explored in Appendix A.

5.1 Measuring Disparity with the Cepstral Filter

The *cepstrum* of a signal is the Fourier transform of the log of its power spectrum[1]. It was developed by Bogert *et al.* [Bogert *et al.*, 1963] as a tool for analyzing signals containing echoes. Such signals can be modeled as an original signal $S(t)$ convolved with a train of impulses, *i.e.*,

$$R(t) = S(t) * (\delta(t) + a_0\delta(t - t_0) + a_1\delta(t - t_1) + \ldots)$$

where $*$ denotes convolution. Taking the log of the power spectrum transforms the received signal into a sum of two terms, one of which depends only on $S(t)$ and the other of which is a combination of distorted sinusoids with frequencies related to t_0, t_1, *etc.*If the cepstrum of $S(t)$ does not overlap the frequencies of the echo terms, conventional linear filtering techniques can be used to extract the values of the echo delays.

Recently Yeshurun and Schwartz [Yeshurun and Schwartz, 1989] developed a way of using the two-dimensional cepstrum as a disparity estimator. The first step of their method is to extract sample windows of size $h \times w$ from the left and right images. The sample windows are then spliced together along one edge to produce an image of size $h \times 2w$. Assuming that the right and left images differ only by a shift, the spliced image may be thought of as an original image at $(0, 0)$ plus an echo at $(w + d_h, d_v)$, where d_h and d_v are the horizontal and vertical disparities. The periodic term in the log power spectrum of such a signal will have fundamental frequencies of $w + d_h$ horizontally and d_v vertically. These are high frequencies relative to the window size. The image-dependent term, by contrast, will be composed of much lower frequencies, barring pathological images. Thus, as Yeshurun and Schwartz show, the cepstrum of the signal will usually have clear, isolated peaks at $(\pm(w + d_h), \pm d_v)$.

5.2 Implementation

Early experiments with a workstation-based implementation of the cepstrum showed that it was robust enough for the vergence application, provided that the sample windows were of adequate size. The smallest acceptable size was found empirically to be 32×32, obtained by subsampling over central 256×256 regions of the left and right input images. Unfortunately, even at this greatly reduced resolution the original implementation required a few seconds per computation on the Sun that acts as the robot's system controller. In order to obtain a more useful sample rate the algorithm was re-implemented on the MaxVideo image processing system. The images are formed by the robot's CCD cameras, which are synchronized so that right and left images reflect the state of the world at the same point in time and become available simultaneously. The video signals are digitized and convolved with anti-aliasing filters (Gaussian, $\sigma = 2.5$ pixels) before being stored in frame buffer memory. The EUCLID DSP microprocessor then subsamples the images and computes the cepstral disparity estimate. The cepstral estimator incorporates a number of optimizations suggested by our analysis and summarized in Appendix B. The final implementation computes the cepstral disparity estimate for 32×32 windows in approximately 51 milliseconds,

[1]This is sometimes referred to as the *power cepstrum* to distinguish it from the *complex cepstrum*, which is the Fourier transform of the complex log of the Fourier transform

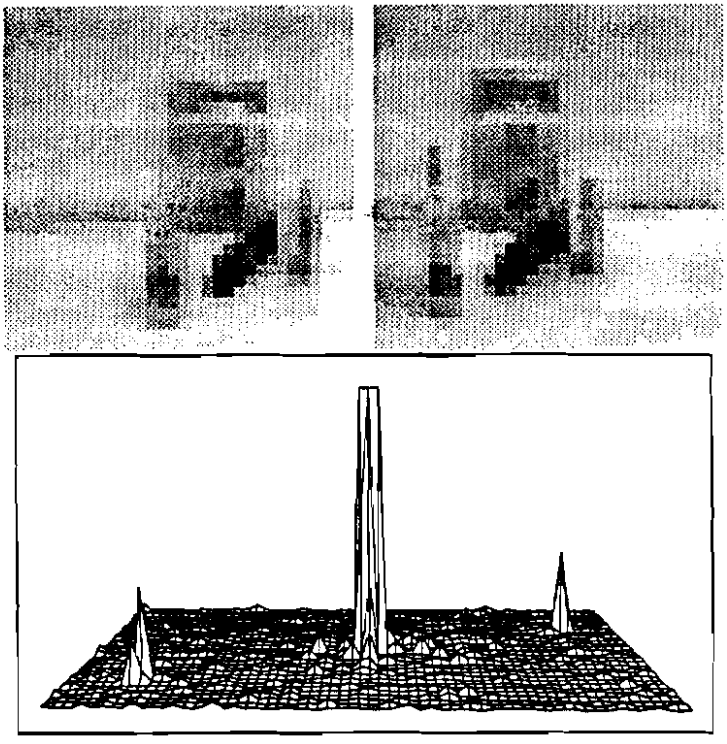


Figure 9: Cepstral disparity estimator sample input and outputs. At top are two 32×32 subsampled images taken by the left and right cameras of the Rochester Robot. Below is a surface plot of the power spectrum of the cepstral filter implementation described in Appendix B. The central peak, which is due to the autocorrelation of the joint image, has been truncated for display. The smaller peaks at left front and right rear give the disparity. Note the splitting of the foreground peak due to the presence of multiple disparities. The dominant disparity in this case is that corresponding to the textbook at the rear of the scene.

not including digitization time or the 8 ms required to acquire the VME bus and read the sample arrays from the frame buffer. Figure 9 shows a sample input and a plot of the cepstral output.

Although the implementation described above is adequate for some purposes, its accuracy is limited by the coarse quantization of the sample windows. For example, with the standard 16mm lenses each pixel in the subsampled cepstral output subtends about 27 arc minutes, or nearly half a degree of visual angle. The current implementation obtains sub-pixel resolution by first finding the peak pixel value in the cepstral output region and then interpolating to better localize the disparity peak. Only the scan line containing the peak value is considered, reducing the problem to 1D peak finding. The 1D sample set is modeled as a discrete approximation to a delta function (*i.e.* a rectangle of width one pixel and unknown height) sampled by integration over adjacent regions of width one pixel. Thus the output of a given sample as a function of disparity should be the convolution of its rectangular sampling window with the rectangular disparity pulse, *i.e.* a triangular pulse

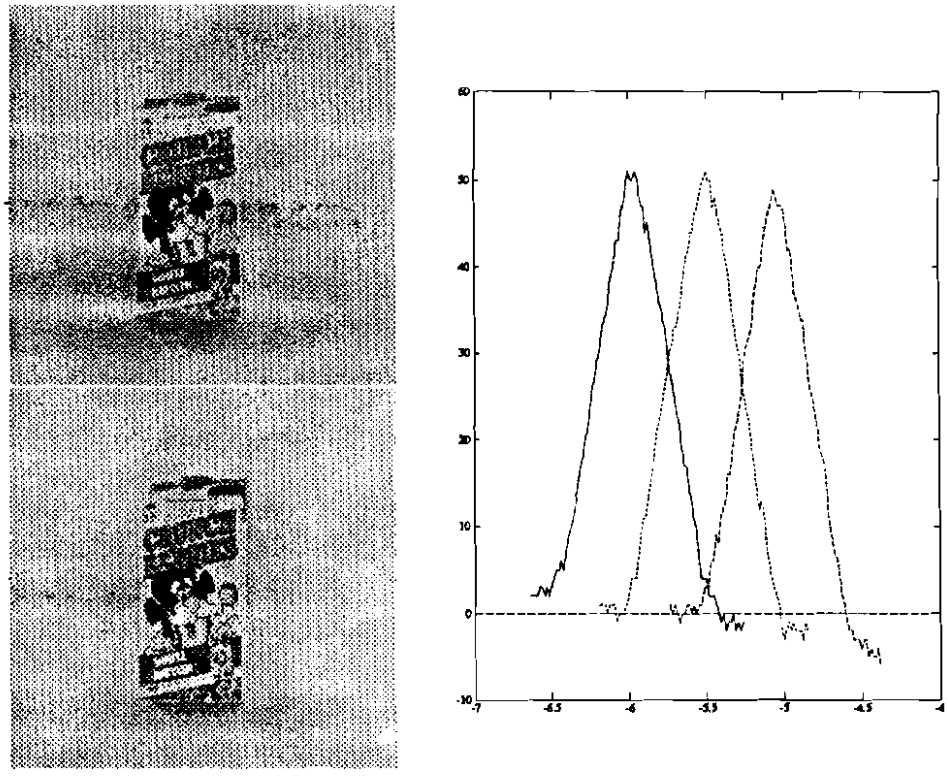


Figure 10: Cepstral output sample responses. At left are right and left camera views of a simple scene. At right is a plot of pixel value versus vergence error in degrees for three adjacent pixels.

whose base is two pixels wide. The interpolation strategy suggested by this model is to return the centroid of the peak pixel and the larger of its two neighbors.

The assumptions underlying the interpolation strategy were tested by recording values of the output samples as a function of sub-pixel disparities generated by moving the cameras while viewing a static scene. Figure 10 shows the responses of three adjacent output samples to a simple scene. The responses have the predicted shape and slope, showing that the model is an accurate description of what happens with real scenes. However, the triangular pulses are broadened slightly at the base. In practice this means that the interpolation strategy described above will produce discontinuities in the estimated disparity at the crossing point of the response curves for the left and right neighbors of the peak. In our implementation these discontinuities are avoided by incorporating a variable fraction of the smaller neighbor into the centroid. The fraction is equal to one minus the difference between the two neighbors divided by the difference between the smaller neighbor and the peak, or:

$$\text{frac} = 1 - \frac{\text{max} - \text{min}}{\text{peak} - \text{min}}$$

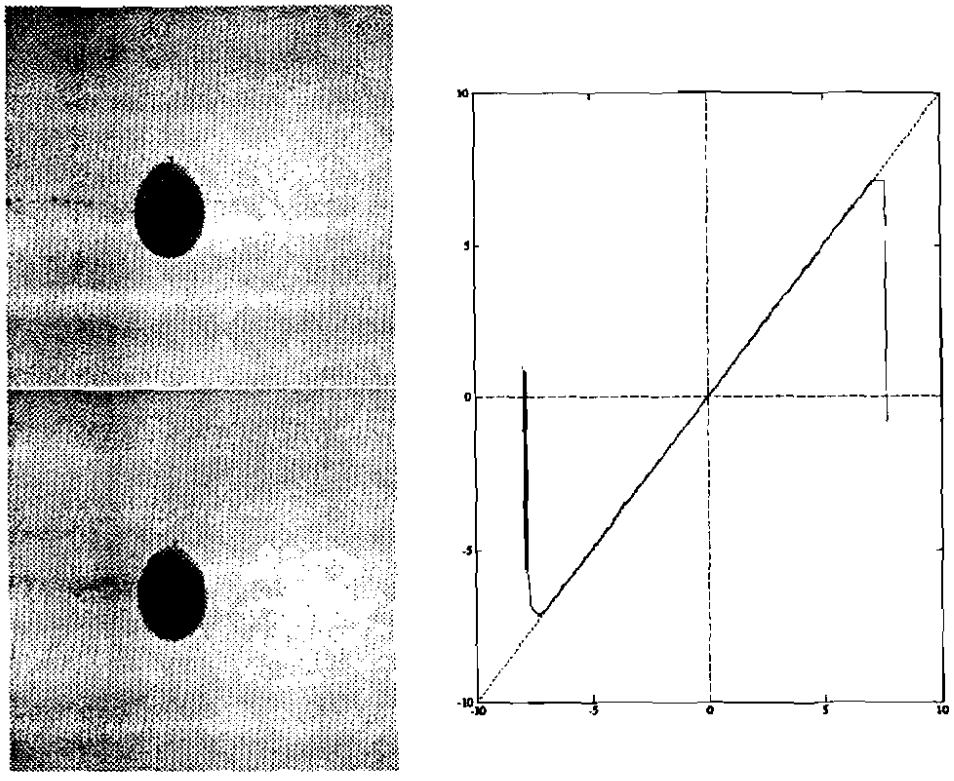


Figure 11: Performance of the cepstral disparity estimator on a simple scene. At left are right and left camera images, taken near zero disparity. At right is a plot of measured versus actual disparity. Data were taken with 16mm lenses, giving a total field of view of 27 degrees. Sample windows used to compute the cepstral estimate were 14 degrees wide. Dotted line shows ideal response, solid shows measured response.

5.3 Performance of the cepstral disparity estimator

The implementation of the cepstral estimator described above was tested in the laboratory on several scenes. For each test, the robot was directed to face the scene and the cameras were manually adjusted to approximately the correct vergence angle for the scene. Taking that angle as the home or zero-disparity position, the test program then swept the cameras over a range of vergence angles. At each position it recorded the actual disparity (represented by the difference between the commanded position and the home position) and the disparity reported by the cepstral estimator running on the EUCLID DSP computer.

Since the home position was only approximately correct at the start of each run, most runs showed a systematic error of one or two pixels. In the plots below, these biases were removed by adding a constant that minimizes the RMS deviation from the ideal $x = y$ response. Figure 11 shows results of a test run on a nearly ideal scene consisting of a balloon against a contrasting background. The estimator fails badly at the extremes of its range, because at disparities exceeding ± 7 degrees the target object is no longer visible in

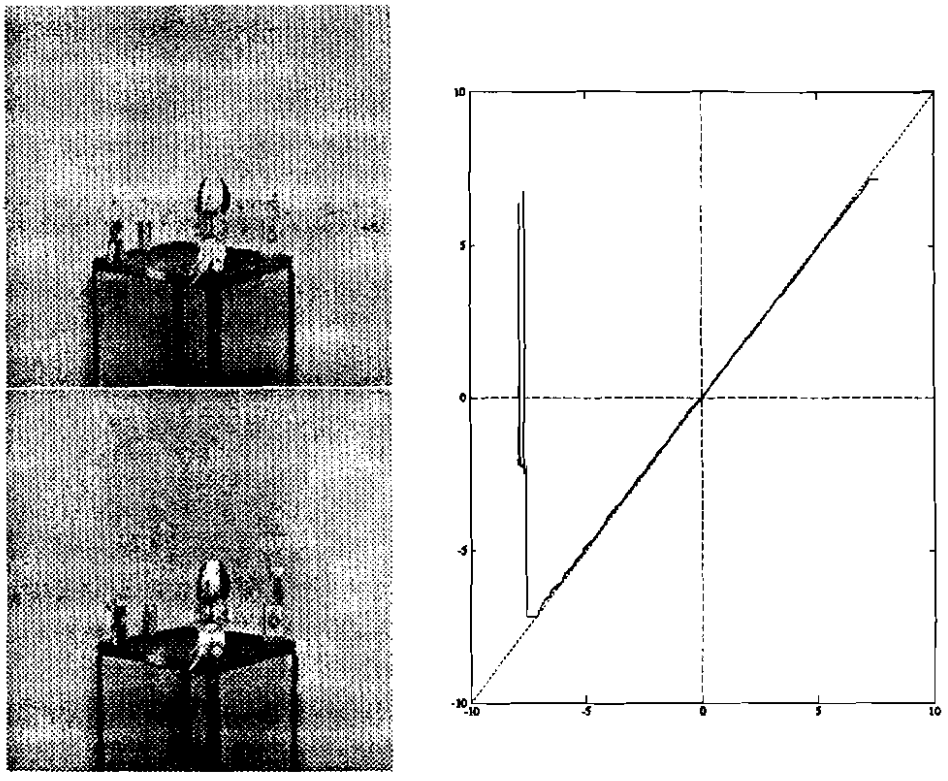


Figure 12: Performance of the cepstral disparity estimator on a more complex scene. Dotted line shows ideal response, solid shows measured response.

both sample windows[2]. Within a ± 7 degree range, however, performance is good. The RMS error is 0.57 pixels, which (with the standard 16mm lenses) corresponds to 1.86 arc minutes. In other words, the estimate is accurate to a little more than half the width of an image pixel. This is quite good, particularly in view of the fact that the cepstral implementation subsamples by a factor of eight. Relative to its sample window resolution, the cepstral RMS error is on the order of one sixteenth of a pixel.

Figure 12 shows results from a more typical laboratory scene. Although the plot looks similar to that in the previous figure, the RMS error for this test was 1.31 image pixels (4.44 arc minutes). The loss of accuracy is primarily due to a small error in the empirically determined constant multiplier used to convert error in pixels to error in degrees. Because the axes of rotation of the cameras do not pass through their nodal points, the nodal points undergo some translation when the cameras rotate. This means that the conversion constant has a small dependence on the depth of the target. Compared to a best-fit straight line, this data set has an RMS error of 0.64 pixels (2.24 arc minutes), roughly comparable to the results in the ideal case. The systematic error could be removed by taking target depth (inferred from current eye position and approximate disparity) into account when converting

[2]Errors of this type can be detected with high probability because they result in anomalous vertical disparities. The control software can use anomalous vertical disparities as a warning to disregard the measured horizontal disparity, and perhaps trigger a reaquisition process.

from pixels to degrees. This has not been necessary to date, because small errors at large disparities have a negligible effect on the performance of the control loop. High accuracy is important only at disparities near zero, where errors or discontinuities can cause the target angle to overshoot or oscillate around the desired value.

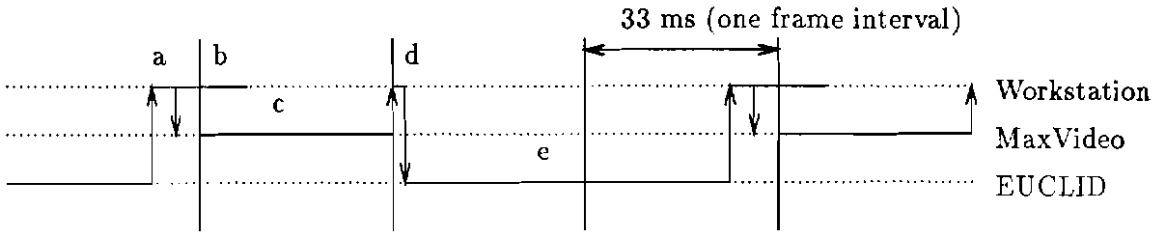
6 Control

The goal of the vergence system is to generate smooth eye movements that correct the vergence error. The vergence control loop consists of three stages: digitization, error estimation, and error correction. Digitization is done under control of the Sun™ host using the MaxVideo™ digitizers, convolvers and frame stores (one each per camera). It takes between one and two RS-170 frame times (33 to 67 milliseconds), depending on how much time remains in the current video frame when the command to acquire the next frame is issued. The Sun is free to do other things during digitization. Once the images are available in the frame store, the Sun signals EUCLID to extract the images from the frame buffers and estimate the disparity. This process takes approximately 59 milliseconds, after which EUCLID places the disparity estimate in a known location in shared memory and issues an interrupt to signal completion. The Sun converts the pixel disparity to angular coordinates by multiplying it by an empirically determined constant, and executes the control law to issue the appropriate velocity command to the eye motors. The Sun issues the motor commands *after* initiating the next digitization in order to allow digitization to proceed concurrently with motor control. This causes a slight delay in issuing the motor commands, but permits a substantially higher overall sampling rate. Figure 13 illustrates the timing of the vergence loop. The loop consistently takes 3 frame times to complete. Thus, the system achieves a servo rate of 10 Hz[3].

6.1 The Controller

The vergence system uses a proportional-derivative (PD) controller (*e.g.*, see [Dorf, 1980]) in cascade with the eye motor in a feedback loop, as shown in figure 14. (Although the target and actual vergence angle are continuous variables, since the entire system under our control is digital or presents digital interfaces we model the system discretely.) The summation node that produces the error signal represents the process of estimating vergence angle error from the disparity of binocular images acquired from the cameras. The controller gains were chosen empirically to obtain slightly underdamped response, resulting in a small overshoot in the step response. The system controls the velocities of the motors to achieve smooth responses to smoothly varying stimuli; controlling the accelerations explicitly would require more computational expense and constant attention of the Sun host.

[3]Since disparity estimation takes 59 ms, the maximum theoretical servo rate is 15 Hz. Attempts to attain this rate have been thwarted by technical difficulties with capturing images and issuing motor commands concurrently with estimating disparity.



a — Set up frame buffers to capture images

b — Issue motor commands

c — Frame buffers capture images

d — Fork cepstral disparity estimator on EUCLID

e — EUCLID grabs subsampled images and estimates disparity

NB: times are approximate, for illustrative purposes.
Figure 13: Vergence Loop Timing Diagram

Since the system directly estimates only the vergence angle error θ_e, an estimate of its derivative, $\dot{\theta}_e$, is numerically derived using the approximation

$$\dot{\theta}_e(kT) = \frac{\theta_e(kT) - \theta_e(kT - T)}{T}$$

where $T = 100$ms. This approximation obviously enhances any noise already in the estimate of vergence error, θ_e. One could employ an α-β tracker or optimal linear predictor (*e.g.*, Kalman filter) to smooth the estimates of θ_e and/or $\dot{\theta}_e$ [Bar-Shalom and Fortman, 1988].

6.2 Performance

The demonstration system's responses to step and sinusoidal stimuli were measured in the lab. Representative camera angle traces of step and sinusoidal responses are shown in Figures 15 and 16. Figure 17 summarizes the system's response to sinusoidal stimuli of frequencies up to 2 Hz. For ease of measurement, the system was not run in the normal mode of compensating for half the error with each camera, but rather one camera alone was moved to correct the entire error and the angle of this camera was recorded.

The step stimulus was produced by manually misconverging the "verging" camera prior to starting the system. The same effect could be achieved by misconverging the camera in the dark and then switching on the lights suddenly at time 0. In the response (Figure 15), observe the single time step (0.1 second) latency in detecting the disparity. As a consequence of this delay, the estimated disparity is seen to lag behind the camera's convergence angle, even though this disparity estimate provides the error signal that drives the vergence system. The small overshoot results from slight underdamping.

The effect of the proportional gain, K_p, is to drive the cameras at higher velocities when the error is larger. The derivative gain, K_d, helps accelerate the response when it is falling behind and decelerates the response when it is overtaking the stimulus, which can

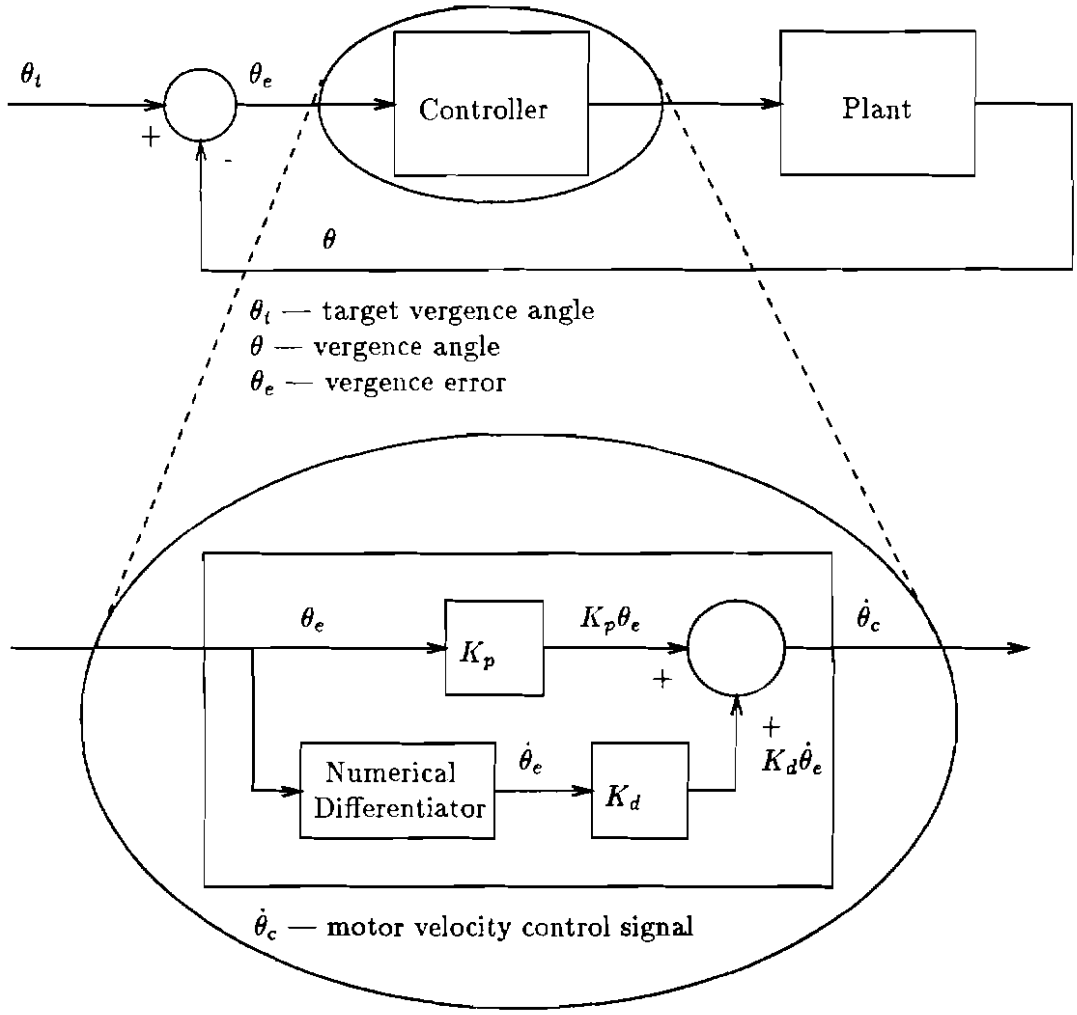


Figure 14: Block diagram of the vergence system.

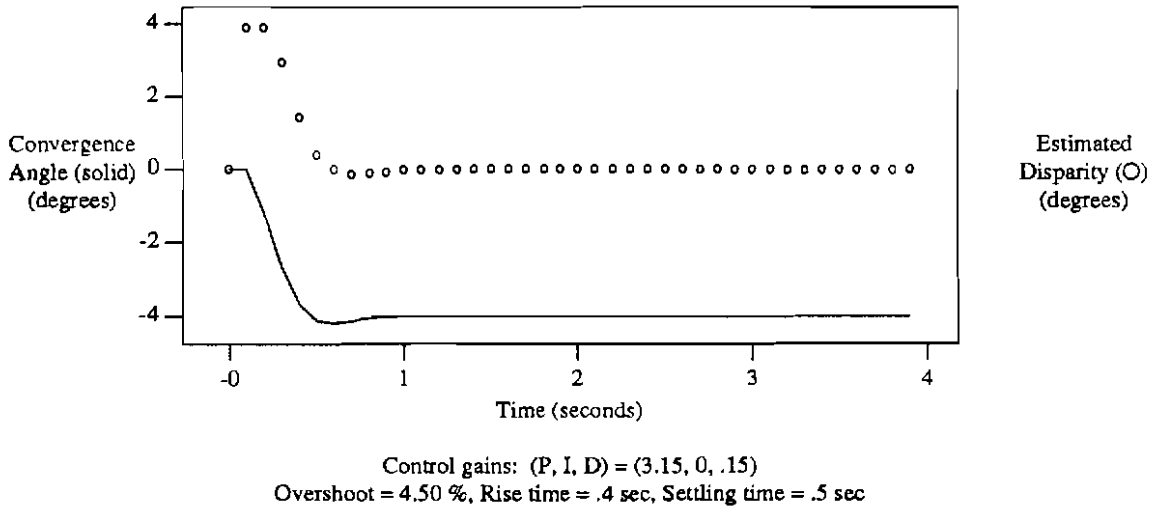


Figure 15: Response to a step in disparity: *Rise time* is the earliest time the response reaches 90% of its final (steady-state) value, and *settling time* is the earliest time the response stays within 5% of its final value. Note that the sample interval is 0.1 seconds.

speed the response. However, a higher derivative gain produces oscillatory response. If K_p is increased, the overshoots become larger. If K_d is increased, oscillations appear in the steady-state response; if K_d/K_p is too large the system becomes unstable. The time delay in the system contributes further to oscillations if the response frequency is so high that responses are large enough to overshoot before they can be detected.

Analogous to the step stimulus, the sinusoidal stimuli were generated by rotating the non-verging camera sinusoidally. If the verging camera is held still, this generates a sinusoidally oscillating disparity signal. Thus, the target vergence angle was defined by the angle of the non-verging (stimulating) camera.

The effect of the time delay on phase lag can be seen by comparing the 0.05 Hz and 0.1 Hz responses in Figure 16: the same time lag contributes proportionately more to the phase lag at higher frequencies, since the time course of each cycle is shorter at higher frequencies.

The vergence responses to sinusoidal stimuli were measured for frequencies ranging from 0.05 to 2 Hz. The gain and the phase shift of the system's responses are summarized in the Bode plot of Figure 17. The system's behavior suggests that it may be a second order system. However, the constant time delay seems to produce a linear phase shift, as shown in Figure 18, since a constant time delay contributes proportionately more to phase shift at higher frequencies.

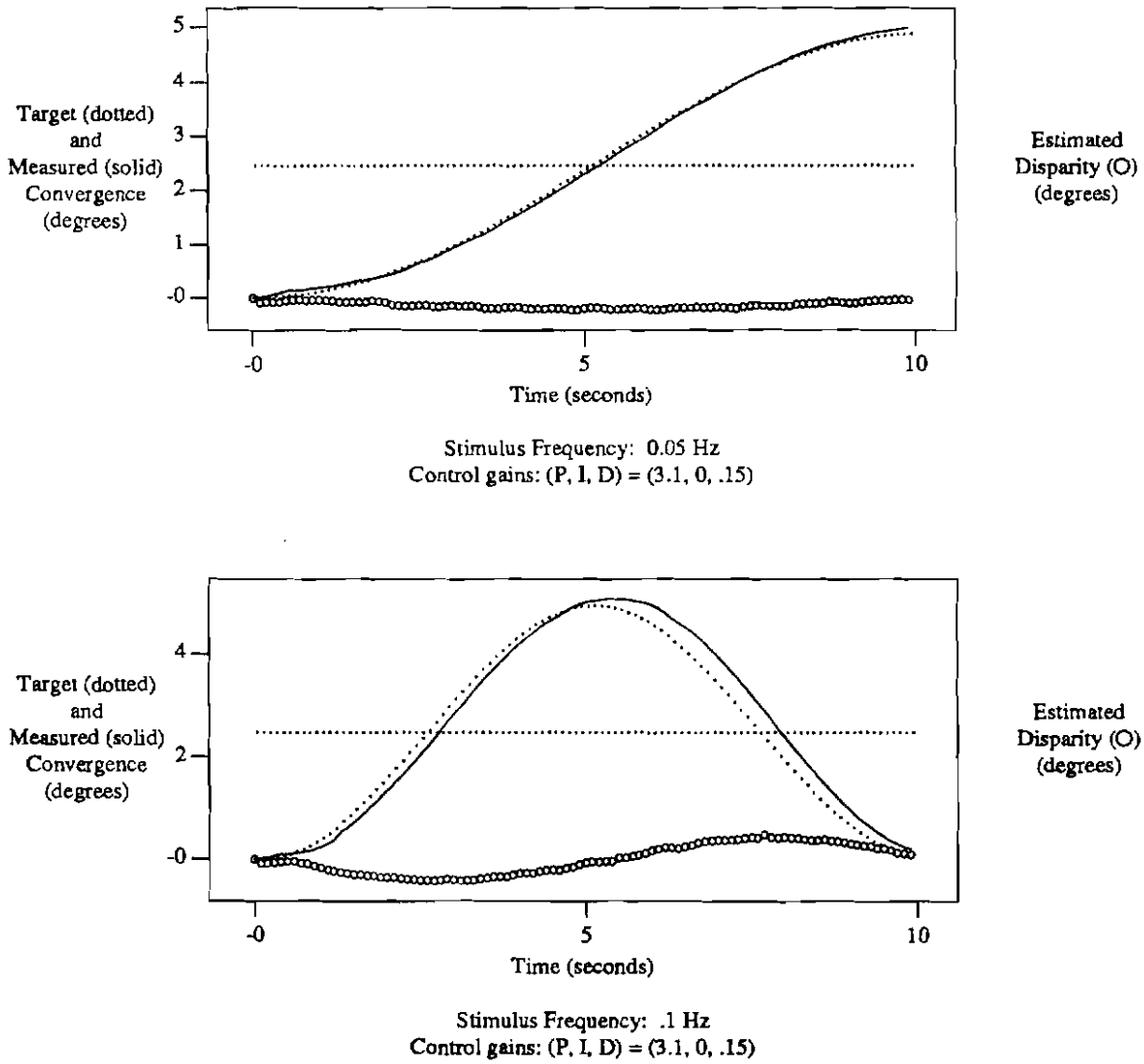


Figure 16: Response to sinusoidal disparity stimulus.

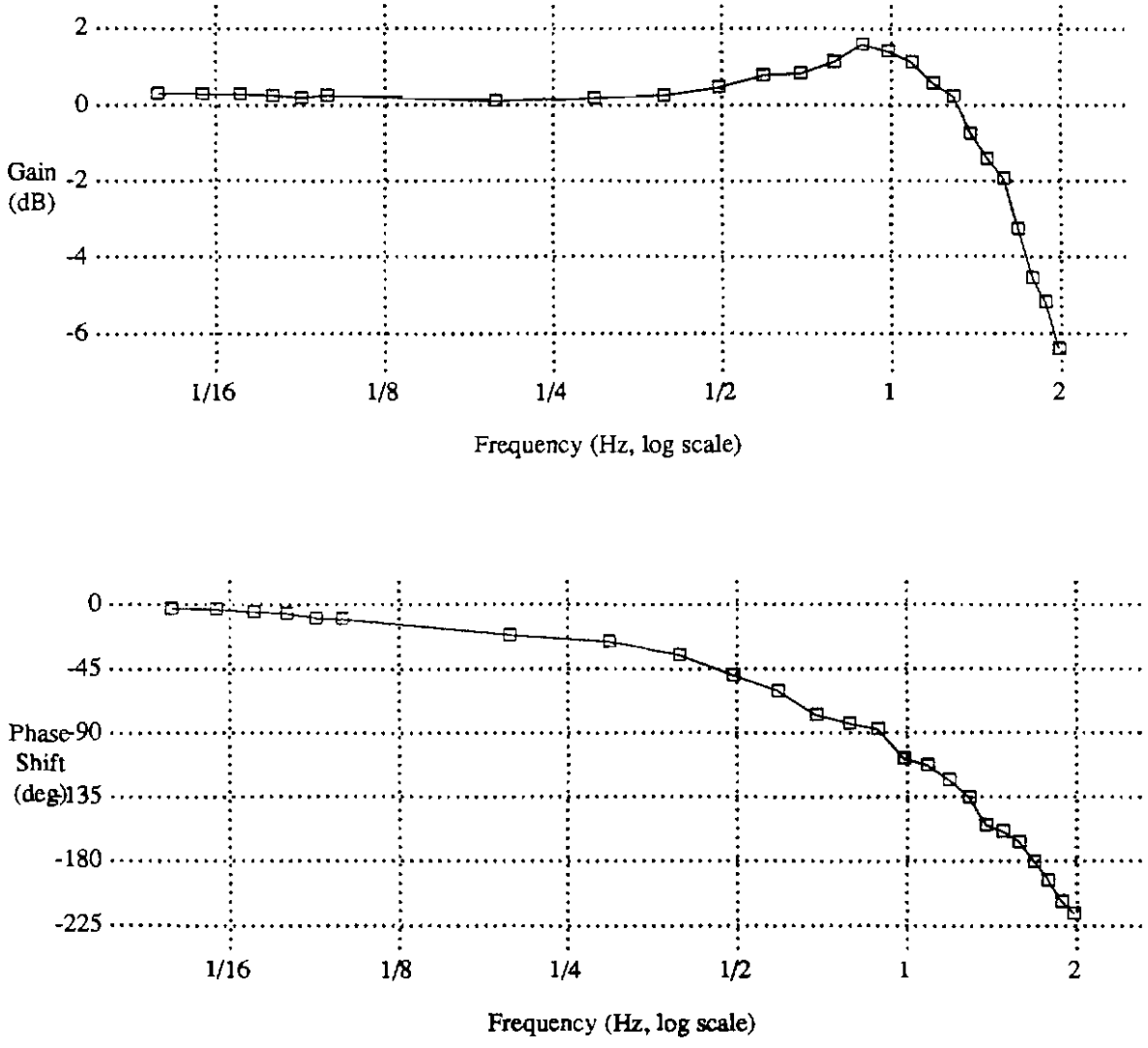


Figure 17: Bode plot: The gain (dB) and phase shift (degrees) of the vergence responses are shown for sinusoidal stimuli of frequencies ranging from 0.05 to 2 Hz. Gain (dB) = $20 * \log_{10}(\frac{\text{response amplitude}}{\text{stimulus amplitude}})$ and the phase shift is the difference in the phase angle of the two signals.

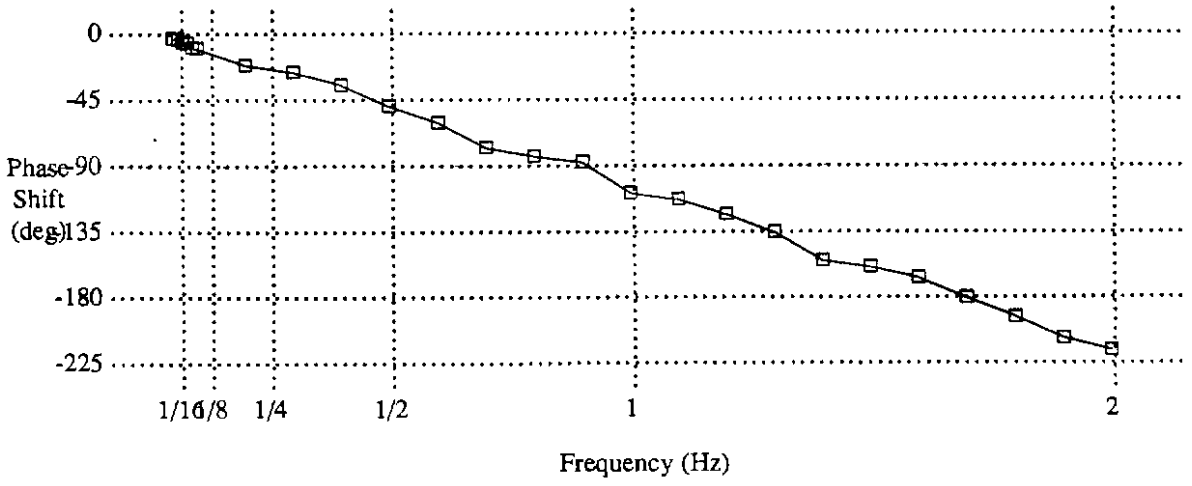


Figure 18: The phase shift (degrees) of the vergence responses is shown plotted against frequency on a linearly scaled abscissa, in contrast with the logarithmic abscissa of the Bode plot, to show that the phase shift seems to be linear. This character of the phase shift arises from a constant time delay, since it contributes proportionately more to phase shift at higher frequencies.

7 Conclusion

We have argued that vergence is important for active vision systems, and have discussed general issues in the design of vergence control systems. We have also described in detail the application of these ideas to develop a real-time vergence control system for the Rochester Robot.

The error estimator for the vergence system is a variant of the cepstral disparity estimator of Yeshurun and Schwartz [Yeshurun and Schwartz, 1989]. The estimator has been shown to be capable of remarkable accuracy, in the best case achieving an RMS error of a small fraction of a pixel. It is simple enough that with a small investment in special hardware it can be computed at speeds comparable to the video frame rate. The cepstral method of disparity estimation can be shown to be equivalent to autocorrelation of images that have been adaptively enhanced to sharpen their autocorrelation functions. It is thus closely related to phase correlation.

The demonstration system uses a position controller that generates smooth vergence camera movements in response to smooth changes in the desired vergence angle. This system also responds reasonably (but suboptimally) to step changes in target vergence angle. Optimal response to step stimuli could be achieved by saccadic vergence movements.

Vergence responses have yet to be integrated with other gaze controls. Future work in the area of gaze control will concentrate on the questions: What gaze controls are prim-

itive, what cues serve them best, and how do visually-mediated controls interact? More sophisticated controls can arise with complex inputs such as optic flow.

Concentrating on visual cues alone is an interesting constraint, but we have not lost sight of the fact that they probably are, in successful systems, combined into a model of target motion which may be used to control gaze [Brown, 1990a; Coombs, 1989a]. There are also non-visual cues such as head accelerations that can inform gaze stabilization systems. Using such non-visual cues calls for models of the observer's physical plant so that the proper compensating movements may be made. Furthermore, the appropriate camera movements to compensate for head motion depend on the gaze location, and so a representation of the location of objects in three dimensions (or at least depth) is needed. There is increasing evidence that human and other primate gaze control systems do indeed make use of such cues [Paige, 1990; Snyder *et al.*, 1990].

Interaction of controls can be simple (say by preemption), or more complex (with controls aware of and cooperating with the actions of other controls [Brown, 1990b; Brown, 1990a; Brown, 1990c]). The former approach requires breaking down the controls into either orthogonal, non-interacting primitives or being content to have one control acting at a time. The latter approach requires more sophisticated modeling of the effects of interaction.

Another area for future exploration concerns the use of camera systems that offer vergence and fixation as reliable primitives. One obvious application is the support of stereo systems with limited fusional ranges [Olson, 1990]. More generally, systems that fixate must choose appropriate targets for the task they are performing. Thus gaze control at the highest level can be viewed as a resource management problem, in which limited sensory and computing hardware must be allocated so as to maximize the usefulness of the recovered information [Rimey and Brown, 1990].

Acknowledgments

Robert Potter was the first to close the vergence loop, and the first to use the cepstral filter for vergence. Dana Ballard and Chris Brown provided leadership and advice, and read several drafts of this paper. Randal Nelson offered many suggestions for improving this work and its presentation. Mike King and Larry Snyder shared their expertise in biological eye movement systems and control systems. The comments of Ray Rimey, Mike Swain, and Lambert Wixson improved the presentation. Dave Tilley and Ray Rimey enhanced the MaxVideo™programming environment with Zebra and Zed [Tilley, 1990] and Tim Becker helped make the PUMA™software usable. Hal Moroff of DataCube, Inc., helped us untangle the mysteries of the EUCLID digital signal processor. Josh Diamond wrote an early version of the cepstral disparity estimator.

A Understanding the cepstral filter

Our experience with the cepstral disparity estimator confirms Yeshurun and Schwartz' observation that the method is remarkably robust. The standard analysis (summarized in Section 5.1) explains what the algorithm does, but does not yield much insight as to why it works so well. We feel that the algorithm is better understood by exploring its relation to autocorrelation; this view also suggests an alternative algorithm. The argument is as follows:

The cepstrum of a signal is computed by forming the power spectrum, taking the logarithm of each pixel, and Fourier transforming the result. Note that the power spectrum is just the Fourier transform of the autocorrelation function of the signal, and (like the autocorrelation function) is both real-valued and even symmetric. The forward and inverse Fourier transforms are equivalent for even, real-valued input functions. Therefore, the second Fourier transform in the cepstrum is equivalent to an *inverse* transform. The cepstrum, then, is the inverse transform of the log of the forward transform of the autocorrelation function. Without the logarithm step, therefore, the algorithm would simply compute the autocorrelation function.

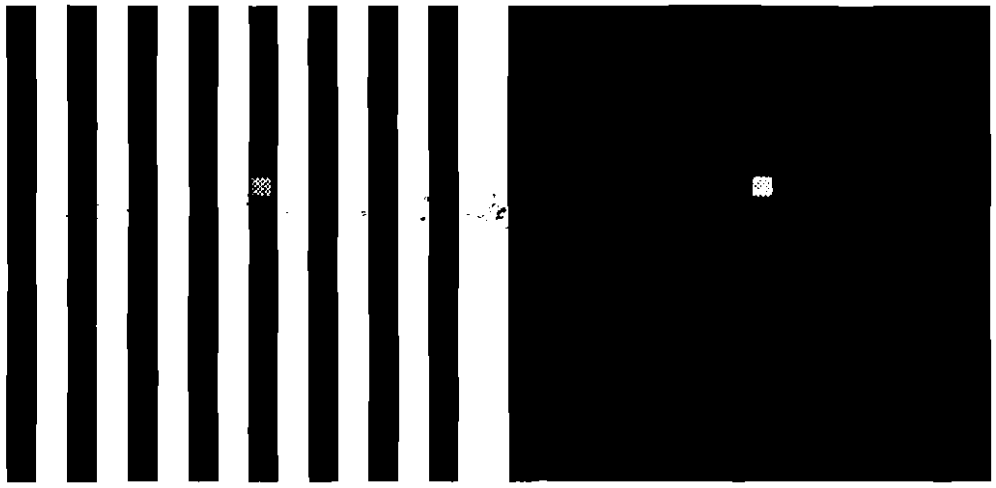
The effect of taking the logarithm before the inverse Fourier transform can be seen by rewriting the log power spectrum as

$$\log|F|^2 = \frac{\log|F|^2}{|F|^2} F F^* = \left| \frac{\sqrt{\log|F|^2}}{|F|} F \right|^2$$

(where F^* is the complex conjugate of F). The right-hand side of this equation can be recognized as the power spectrum (*i.e.* the Fourier transform of the autocorrelation) of a filtered version of the original function. In other words, the cepstrum can be thought of as autocorrelation with an adaptive (non-linear) prefilter. The prefilter is compressive in the frequency domain—it tends to make the power spectrum more nearly uniform, reducing the contribution of narrowband signals while leaving broadband signals relatively unaltered. Narrowband signals include such things as periodic patterns and large smooth blobs, both of which are poor correlation targets. By suppressing narrowband signals, therefore, the prefilter makes the input a better, less ambiguous correlation target. The effect can be seen by applying the appropriate prefilter to images that contain both good and bad autocorrelation targets, as shown in Figure 19. As can be seen, the periodic part of the signal has been largely suppressed, while parts of the image that have unique matches have been enhanced.

This view of the cepstrum suggests that any non-linear compressive function applied to the power spectrum should have a similar sharpening effect. Informal experiments suggest that this is indeed the case. For example, replacing the log step in the cepstral algorithm with a fourth root or arc tangent produces results that do not differ greatly from the standard cepstrum.

The ultimate compressive operator would be one that takes all input values to a constant. The Fourier transform of a constant is an impulse at $(0,0)$, so this operator would provide the unhelpful information that the image matches itself perfectly at a disparity of zero. In



a) b)

Figure 19: a) an image $f(x, y)$ containing both good and bad correlation targets. Both types of targets have adequate high frequency content, but the periodic grid is subject to false matches both horizontally and vertically. b) The same image after application of the cepstral-equivalent filter. This image is a much clearer correlation target.

order to get useful disparity information by this method one must find a way to preserve the phase information that is normally destroyed by formation of the power spectrum. For example, one might compute the transformed cross-correlation of the left and right images by multiplying the transform of one times the conjugate of the transform of the other, and then rescale so that all entries in the resulting complex array have the same magnitude. This intuitively derived algorithm can be rigorously justified as a type of deconvolution, as follows:

The cepstral disparity estimator depends on the assumption that the right and left images differ only by a shift of d_h horizontally and d_v vertically. Given this assumption, however, a more direct approach is possible. The stated assumption is equivalent to the formula

$$R(x, y) = L(x, y) * \delta(x - d_h, y - d_v)$$

where $*$ represents convolution. Fourier transforming and solving for the disparity term gives

$$e^{-j2\pi(ud_h + vd_v)} = \frac{F_R(u, v)}{F_L(u, v)}$$

or

$$\delta(x - d_h, y - d_v) = \mathcal{F}^{-1}\left(\frac{F_R(u, v)F_L^*(u, v)}{|F_L(u, v)|^2}\right). \tag{1}$$

By hypothesis, however, F_L and F_R have identical magnitude spectra—they differ only in phase, because the left and right images differ only by a shift. Thus the division can be rewritten as

$$\frac{F_R(u, v)F_L^*(u, v)}{|F_R(u, v)F_L^*(u, v)|}, \tag{2}$$

which is the Fourier transform of the crosscorrelation of the right and left images, rescaled so that all entries have magnitude one. That is, it is exactly the procedure suggested above by intuition. What was described there as a peak-sharpening operation turns out to undo the convolution of $L(x, y)$ with the disparity delta function.

Like the cepstrum, the deconvolution disparity estimator can be understood as correlation with an adaptive prefilter. In this case the effect of the prefilter is to obliterate the magnitude spectra of each image, so that the images differ only in phase. Deconvolution is thus equivalent to correlating phase images, and the technique is well known under the name of *phase correlation*. Kuglin and Hines [Kuglin and Hines, 1975] first described the algorithm and showed that the height of the correlation peak and the distribution of the background values can be used to estimate the extent to which the two images do in fact differ by a shift. Pearson *et al.* [Pearson *et al.*, 1977] describe a cleverly optimized hardware implementation of the algorithm that transforms 128×128 sample windows at 30 frames per second.

In theory, phase correlation should be somewhat faster than the Yeshurun and Schwartz cepstral disparity estimator for a given sample window size. This is because the cepstral estimator is based on Fourier transforms of windows of size $h \times 2w$, while phase correlation replaces those with fewer than twice as many transforms of windows of size $h \times w$. Since the running time of the Fast Fourier transform (FFT) rises more than linearly with increasing sample window size, converting to phase correlation should reduce the time needed to estimate the disparity. However, this neglects the problem of wraparound. Like all Fourier-based approaches to discrete correlation, phase correlation (and the cepstrum) compute wrapped correlations, which can lead to ambiguities in the sign of the disparity. In the case of the cepstrum, the ambiguity can be resolved without padding by the strategy described in Appendix B. For phase correlation, however, this strategy fails. The padding required to prevent wraparound overwhelms the apparent speed advantage of the phase correlation estimator.

B Efficient Computation of the Cepstral Filter

The main body of the cepstral algorithm consists of a 2-D FFT, a point transform (the log of the power spectrum), and a second 2-D FFT. In this respect it resembles standard linear filtering, so standard optimizations apply. The implementation used on the Rochester Robot transforms first the columns and then the rows using a one dimensional decimation-in-frequency (DIF) FFT that expects normally ordered input and produces bit-reverse ordered output. After the point transform, a DIT FFT is used to transform first the rows and then the columns, undoing the bit-reversal at the same time. The point transform approximates $\log_2 n$ by counting the number of bits in the (real-valued) product of each pixel and its complex conjugate. Although the ADSP-2100 lacks floating point instructions, it does have a barrel shifter that provides single-cycle normalization. Therefore, counting the bits requires only two or three instructions per point.

The most important optimization treats wraparound in the transform domain. A discrete Fourier transform (DFT) of size $h \times 2w$ can only handle frequencies in the range $-h/2$ to $h/2$ vertically and $-w$ to w horizontally. In the continuous case the cepstral transform

produces peaks at $(\pm(w + d_h), \pm d_v)$, so positive horizontal disparities will produce peaks falling outside the range of the DFT. Since DFTs are circular, positive disparities will wrap around to the opposite ends of the horizontal frequency axis. The result is that disparities of (d_h, d_v) and $(-d_h, -d_v)$ will be indistinguishable.

The standard solution to this problem is to widen the input (the spliced image) by padding with zeros. However, doing so substantially increases the cost of the algorithm in both time and space. For applications like vergence and stereopsis a more economical solution is possible. In these cases the vertical disparity is known to be small—specifically, it can be assumed to be in the interval $\pm h/4$. The disparity peaks can be "tagged" by introducing an artificial vertical disparity of $+h/4$, *i.e.* by "rolling" the right-hand sample window upward by 1/4 of its height. This is achieved by transforming the right-hand sample window by moving row m to row $(m + h/4) \bmod h$. After the cepstral transform, peaks are located and interpolated as usual. Since the vertical disparity should be positive, a negative vertical disparity indicates that the horizontal disparity has wrapped around and hence has incorrect sign.

A final optimization exploits the fact that in the second 2-D FFT the final set of column transforms only needs to be done in the region that will be searched for peaks. This region consists of the columns representing horizontal frequencies between $w/2$ and w. This makes it possible to eliminate 3/4 of the column transforms.

All of the optimizations described in the previous section are incorporated into the EUCLID implementation of the cepstral disparity estimator. Sixteen-bit fixed-point arithmetic was used throughout. The DIT and DIF FFTs are implemented as assembly language subroutines, as is the procedure that computes the power spectrum and takes its log. All other code is written in C. This version of the estimator computes a disparity in 32×32 windows in 51 milliseconds, not counting the 8 milliseconds (average) necessary to acquire the VME bus and extract the sample windows from the MaxVideo frame buffer.

References

[Abbot and Ahuja, 1988] A. Lynn Abbot and Narendra Ahuja, "Surface Reconstruction by Dynamic Integration of Focus, Camera Vergence, and Stereo," In *International Conference on Computer Vision*. IEEE, 1988.

[Aloimonos *et al.*, 1987] J. Aloimonos, I. Weiss, and A. Bandopadhay, "Active Vision," In *Proc. 1st Int'l. Conf. on Computer Vision*, pages 35–54, London, June 1987.

[Analog Devices, 1987] Analog Devices, Inc., Norwood, Massachussetts, *DSP Products Databook*, 1987.

[Bajcsy, 1986] Ruzena Bajcsy, "Passive Perception vs. Active Perception," In *Proc. IEEE Workshop on Computer Vision*, Ann Arbor, 1986.

[Ballard, 1989] Dana H. Ballard, "Reference Frames for Animate Vision," In *International Joint Conference on Artificial Intelligence*. AAAI, 1989.

[Ballard and Ozcandarli, 1988] Dana H. Ballard and Altan Ozcandarli, "Eye Movements and Visual Cognition: Kinetic Depth," In *International Conference on Computer Vision*. IEEE, December 1988.

[Bandopadhay, 1986] Amit Bandopadhay, *A Computational Study of Rigid Motion Perception*, PhD thesis, University of Rochester, 1986, also published as Computer Science Department TR 221.

[Bar-Shalom and Fortman, 1988] Yaakov Bar-Shalom and Thomas E. Fortman, *Tracking and data association*, Academic Press, 1988.

[Barnard and Fischler, 1982] S. T. Barnard and M. A. Fischler, "Computational Stereo," *Computing Surveys*, 14:553–572, December 1982.

[Bogert *et al.*, 1963] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-Cepstrum, and Saphe Cracking," In M. Rosenblatt, editor, *Proc. Symp. Time Series Analysis*, pages 209–243. John Wiley and Sons, 1963.

[Brooks, 1986] Rodney A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, pages 14–23, April 1986.

[Brooks, 1987] Rodney A. Brooks, "Intelligence Without Representation," In *Proc. Workshop on Foundations of Artificial Intelligence*, pages 1–21, 1987.

[Brown, 1990a] C. M. Brown, "Gaze controls cooperating through prediction," *Image and Vision Computing*, 8(1):10–17, February 1990.

[Brown, 1990b] C. M. Brown, "Gaze controls with interactions and delays," *IEEE Transactions on Systems, Man, and Cybernetics, in press*, IEEE-TSMC20(3), May 1990.

[Brown, 1990c] C. M. Brown, "Prediction and cooperation in gaze control," *Biological Cybernetics, in press*, May 1990.

[Brown *et al.*, 1988] Christopher Brown, Dana Ballard, Timothy Becker, Roger Gans, Nathaniel Martin, Thomas Olson, Robert Potter, Raymond Rimey, David Tilley, and Steven Whitehead, "The Rochester Robot," Technical Report 257, University of Rochester, Computer Science Department, 1988, Christopher M. Brown (Editor).

[Burt and Adelson, 1983] Peter J. Burt and Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[Clark and Ferrier, 1988] James Clark and Nicola Ferrier, "Modal Control of an Attentive Vision System," In *International Conference on Computer Vision*. IEEE, 1988.

[Coombs, 1989a] David J. Coombs, "Gaze Stabilization for Animate Vision," Thesis Proposal, December 1989.

[Coombs, 1989b] David J. Coombs, "Tracking Objects with Eye Movements," In *Topical Meeting on Image Understanding and Machine Vision*. Optical Society of America, 1989.

[Dorf, 1980] Richard C. Dorf, *Modern control systems*, Addison-Wesley, 3rd edition, 1980.

[Erkelens *et al.*, 1989a] C. Erkelens, J. Van der Steen, R. Steinman, and H. Collewijn, "Ocular Vergence Under Natural Conditions I: Continuous Changes of Target Distance Along the Median Plane.," *Proceedings of the Royal Society of London*, 1989.

[Erkelens *et al.*, 1989b] C. Erkelens, R. Steinman, and H. Collewijn, "Ocular Vergence Under Natural Conditions II: Gaze Shifts Between Real Targets Differing in Distance and Direction.," *Proceedings of the Royal Society of London*, 1989.

[Erkelens and Regan, 1984] C. J. Erkelens and D. Regan, "Human Ocular Vergence Movements Induced by Changing Size and Disparity," *Journal of Physiology*, 379:pp. 145–169, 1984.

[Fleet *et al.*, 1989] David J. Fleet, Allan D. Jepson, and Michael R. M. Jenkin, "Phase-based Disparity measurement," Research in Biological and Computational Vision RBCV-TR-89-29, Department of Computer Science, University of Toronto, November 1989.

[Horn, 1986] Berthold K. P. Horn, *Robot Vision*, The MIT Press, Cambridge, Massachusetts, 1986.

[Howard and Simpson, 1989] Ian Howard and W. Simpson, "Human Optokinetic Nystagmus Is Linked to the Stereoscopic System," *Experimental Brain Research*, 1989.

[Jepson and Jenkin, 1989] Allan D. Jepson and Michael Jenkin, "The Fast Computation of Disparity from Phase Differences," In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 398–403, San Diego, June 1989.

[Krotkov, 1989] Eric Paul Krotkov, *Active computer vision by cooperative focus and stereo*, Springer-Verlag, 1989.

[Kuglin and Hines, 1975] C. D. Kuglin and D. C. Hines, "The Phase Correlation Image Alignment Method," In *Proc. IEEE Int'l Conf. on Cybernetics and Society*, pages 163–165, 1975.

[Maxwell and King, 1990] J. S. Maxwell and W. M. King, "Disjunctive Saccades contribute to vergence in Rhesus Monkeys," In *Abstracts of Society for Neuroscience.* Society for Neuroscience, 1990.

[Mayhew and Longuet-Higgins, 1982] J. E. W. Mayhew and H. C. Longuet-Higgins, "A Computational Model of Binocular Depth Perception," *Nature*, 297:376–378, 1982.

[Mead and Mahowald, 1988] Carver A. Mead and M. A. Mahowald, "A Silicon Model of Early Visual Processing," *Neural Networks*, 1:pp. 91–97, 1988.

[Miles *et al.*, 1990] F. A. Miles, U. Schwarz, and C. Busettini, "The Decoding of Optic Flow by the Primate Optokinetic System," In A. Berthoz, P.-P. Vidal, and W. Graf, editors, *The Head-Neck Sensory-Motor System.* Wiley, to appear 1990.

[Olson, 1990] Thomas J. Olson, "Stereopsis for Fixating Systems." Technical Report in preparation. University of Virginia Department of Computer Science, 1990.

[Olson and Potter, 1989] Thomas J. Olson and Robert D. Potter, "Real-Time Vergence Control," In *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, pages 404–409, San Diego, June 1989, also available as University of Rochester Computer Science Department Technical Report 246.

[Paige, 1990] G. D. Paige, "Modulation of the linear vestibulo-ocular reflex (LVOR) by vergence," *Investigative Ophthalmology and Visual Science*, 31(4):121, 1990, Annual ARVO Meeting Abstract Issue.

[Pearson *et al.*, 1977] J. J. Pearson, D. C. Hines, Jr., S. Golosman, and C. D. Kuglin, "Video-rate Image Correlation Procssor," In *SPIE v. 119, Applications of Digital Image Processing*, pages 197–205, San Diego, 1977.

[Rimey and Brown, 1990] R. D. Rimey and C. M. Brown, "Selective attention as sequential behavior: Modelling eye movements with an augmented hidden Markov model," Technical Report 327, University of Rochester, Computer Science Department, February 1990.

[Robinson, 1987] David Robinson, "Why Visuomotor Systems Don't Like Negative Feedback and How They Avoid It," In Michael Arbib and Allen Hanson, editors, *Vision, Brain and Cooperative Computation.* MIT Press, 1987.

[Snyder *et al.*, 1990] Lawrence H. Snyder, Diane M. Pickle, and W. Michael King, "Does instantaneous vergence angle modify the vestibulo-ocular reflex in monkeys?," *Investigative Ophthalmology and Visual Science*, 31(4):121, 1990, Annual ARVO Meeting Abstract Issue.

[Tilley, 1990] David G. Tilley, "Zebra for MaxVideo: an application of object oriented microprogramming to register level devices," Technical Report 315, University of Rochester, Computer Science Department, 1990.

[Tistarelli and Sandini, 1990] Massimo Tistarelli and Giulio Sandini, "On the Estimation of depth from motion using an anthropomorphic visual sensor," In *European Conference on Computer Vision*, 1990.

[Tsotsos, 1987] John K. Tsotsos, "A 'Complexity Level' Analysis of Vision," In *Proc. First International Conference on Computer Vision*, pages 825–834, London, June 1987.

[Van der Spiegel *et al.*, 1989] J. Van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Bellutti, and G. Soncini, "A Foveated Retina-Like Sensor Using CCD Technology," In C. Mead and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*. Kluwer, 1989.

[Yarbus, 1967] A. L. Yarbus, *Eye Movements and Vision*, Plenum Press, New York, 1967.

[Yeshurun and Schwartz, 1989] Yehezkel Yeshurun and Eric Schwartz, "Cepstral Filtering on a Columnar Image Architecture: A Fast Algorithm for Binocular Stereo Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), July 1989.