ELECTRICAL ENGINEERING

A BIOLOGICALLY INSPIRED ACTIVE VISION GAZE CONTROLLER

ATIT SRIKAEW

Dissertation under the direction of Associate Professor Richard Alan Peters, II, Ph.D.

An active vision system (AVS) has been created as a general purpose vision system for a robot. The AVS enables the selective, localized capture of imagery for visual analysis. With it, the robot can direct its gaze around the environment to track an object or to perform a visual search. The ability to react to, or probe the environment simpli¯es a number the robot's interactions with the world.

This work presents both the design and the implementation of a camera head controller for the AVS of the humanoid, ISAC. The camera head comprises two color cameras mounted on a four degree-of-freedom head (pan, tilt, left verge, and right verge). The camera head controls produce camera movements that are analogous to those of human eyes. Human eye-motion can be classi¯ed into three voluntary movements: saccades, smooth pursuit, and vergence, and two involuntary movements: vestibulo-ocular re°ex and opto-kinetic re°ex.

Control methods have been designed and tested through experimentation. The Five basic human-like eye movements have been robustly attained and have resulted in demonstrable improvements in overall visual performance performance. Using a distributed network of personal computers which are tightly integrated through a

unique software architecture, the AVS performs with more than adequate speed and accuracy. Furthermore, because of the simple structure of its hardware, this AVS is conveniently portable to other robots. The results of using the the described camera head motions are faster performance, and smoother, more stable, and more robust control of the camera head than was possible with more traditional control methods. This AVS is implemented as an integral part of the humanoid robot ISAC in Intelligent Robotic Laboratory.

Approved_____ Date_____

A BIOLOGICALLY INSPIRED ACTIVE VISION GAZE CONTROLLER

By

Atit Srikaew

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

August, 2000

Nashville, Tennessee

Approved:	Date:
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

ACKNOWLEDGEMENTS

Sincerely thanks to

Dr. Alan Peter, Dr. Mitchell Wilkes, Dr. Kazuhiko Kawamura, Dr. Dan Gaines,

and Dr. Je® Schall

for all of their most valuable supports in this work.

And for those who have been giving so much of all meaningful time to me, you know

I really thank you all,

...knowing that without one of you, I would not have been me today.

My parents, for the best of their unending source of love and encouragement to me...

Especially, my lovely god mother, who really built me as I am today,

...my teachers for their most valuable time,

...my friends for the best of their time,

...my school for the greatest experience,

...my country for the best supports,

...it's nearly impossible to express how grateful I am within only a few hundred

words here...

\...and for the best of my love, I always believe in you..."

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Overview

This paper describes the camera head for the Intelligent Soft Arm Control (ISAC) humanoid robot [1][2], an active vision system with capabilities similar to that of the human visual system. It consists of two color cameras mounted on a head with four degrees of freedom (pan, tilt, left verge and right verge). The camera controls are designed to mimic the movements of the human eyes. These movements can be classi¯ed into three voluntary and two involuntary movements [3][4]. Voluntary movements include saccades, smooth pursuit and vergence. Involuntary movements include the vestibulo-ocular re°ex (VOR) and the opto-kinetic re°ex (OKR).

- Saccades are the ballistic movements of the eyes when they jump from one ¯xation point in space to another. Saccades can be intentional or re°exive and bring the image of a new visual target onto the fovea.
- Smooth-pursuit movement maintains a ¯xation point of a target moving at moderate speeds on the fovea. Multiple clues from the target, such as color and motion, are utilized for robust tracking.
- Vergence movement adjusts the eyes so that the optical axes keep intersecting on the same target while depth varies. It ensures that both eyes ¯xate on the same target. Disparity clues play an important role in the vergence system.

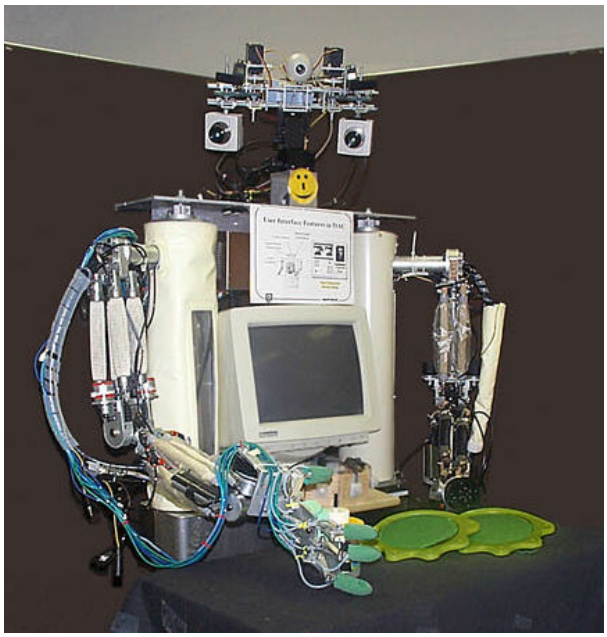


Figure 1: ISAC humanoid robot

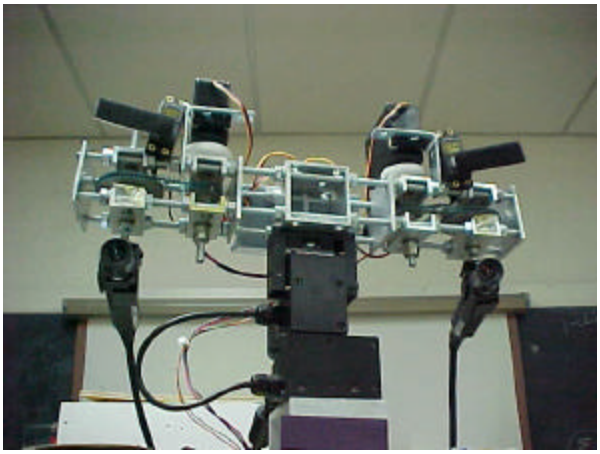


Figure 2: ISAC camera head

- The vestibulo-ocular re°ex (VOR) and the opto-kinetic re°ex (OKR) are mechanisms for stabilizing the image of the target during head movements. VOR uses information about head velocity received from organs in the head, while OKR utilizes image slip from the entire visual ¯eld to compensate for the head motion.

Purpose of Study

The purpose of this study is to design and implement the camera head control schemes for the ISAC camera head to achieve these ¯ve basic human-like eye movements. These basic controls provide human-like camera head motion. The system is integrated as part of an attentional system for ISAC, resulting in a smoother and more stable, robust visual system compared to traditional camera head control methods. To achieve this purpose, the following goals will be accomplished:

- Robust attainment of ¯ve basic human-like eye movements, shown to improve overall performance.

This goal can be examined with respect to both design and implementation of each eye movement control. For saccades, the system is able, with only one command set, to turn the camera head at maximum speed to center at the target. By using a back-propagation neural network and mapping between the camera image plane and motor positions, the saccade module can move with su±ciently high accuracy to bring the target onto the fovea (the optical center of the image).

The smooth-pursuit module then continues to track the object at moderate speed and keep it on the fovea until the target is lost from its visual ¯eld, either because

the target moves too fast and leaves the fovea area or because a new target position is detected and located. A new saccade command is then ¯red. The smooth-pursuit system utilizes both target position and velocity for tracking.

The vergence system ensures that both left and right cameras ¯xate on the same object by keeping disparity between the left and right image to a minimum. It depends on a disparity estimate module.

The VOR behavior uses information about head motor velocity to counter-rotate left and right eyes to keep the camera away from its mechanical limits (it always turns the head toward the target). The OKR acts as backup system for the VOR system. It compensates the head movement using the entire image motion ¯eld to stabilize the left and right eyes while the head is turning.

All ¯ve human-like eye movement controls run simultaneously. The system can track the moving target continuously until the target is out of the visual ¯eld.

- Performance of all ¯ve basic human-like eye movement controls with desirable speed and accuracy, but without additional special hardware

The ISAC head was built as a basic active-vision platform with only four degrees of freedom (left, right, pan and tilt). Two regular color charge-coupled device (CCD) cameras are mounted on the head. The pan-tilt unit was commercially manufactured and provides fairly high speed and accuracy for controlling pan and tilt motors. The verge unit was built in-house using hobby motors. Therefore, highly accurate left-right motor control is not expected. The system uses a PC-based platform running Windows NT without any special hardware such as a high-performance digital signal processing (DSP) board for high-volume computation. The system consists of two

PCs. A standard PC controls the camera head via regular serial ports, while a high-end dual Pentium™ III-based PC performs all image processing, including acquisition and display of left and right color images. Such limited resources requires the system to be carefully designed. All the human eye movement characteristics must be achieved with desirable speed and accuracy, using only simple, straightforward image-processing technology.

- Convenient portability of the prototype system to other binocular vision platforms

The standard PC basis of this system, combined with its relatively simple design and implementation, makes it possible to port this AVS control system to other binocular vision platforms with minimum e®ort. The system is implemented on the ISAC head, which has a simple structure. Additionally, the use of simple image-processing routines can make it far less complicated to design robust software for controlling the system.

Scope of Study

This research will focus on the design and implementation of binocular camera head controls to perform four basic human-like eye movements, using very limited hardware and software resources, while maintaining computational simplicity. The scope of the study can be summarized in the following sections:

1. Target detection and segmentation

   One objective of ISAC camera head control is to imitate the human ability to interpret visual scenes and then track the attended object. This ability enables

ISAC to visually discern objects from a complex scene.

If objects move, the AVS can distinguish and track them. Motion provides information for human-like perception, for example that an object is a separate entity that moves in relation to other objects. Like a person, ISAC can easily discern moving objects.

Conversely, if there is no moving object in the ¯eld of view, knowledge of the object structure or surface characteristics is required to di®erentiate the object from irrelevant information. In this system, the target can be located by using its color information [5]. Applying color segmentation reduces irrelevant information when the target is part of a cluttered scene. Any color model can be built and stored in a color database, for example human skin-tone color. Skin-tone segmented images yield information that can be used to identify a human body part such as a face or hand. Because ISAC is designed to interact with people, visual signals from human users are then attended to. Color information can be retrieved at any time, regardless of movement of the target.

It is problematic to detect a person, however, when non-human skin-tone objects appear in the environment. This problem is overcome by extracting target motion along with skin-tone color information. By using both types of information, any skin-tone object that is not part of a human body is ignored.

The target motion can be computed from optical °ow [6][7]. Typically, the optical °ow estimate is computationally expensive. Therefore, this study estimates optical °ow through the use of algorithms to achieve accuracy at real-time speed. This makes the essential information about the target - that concerning

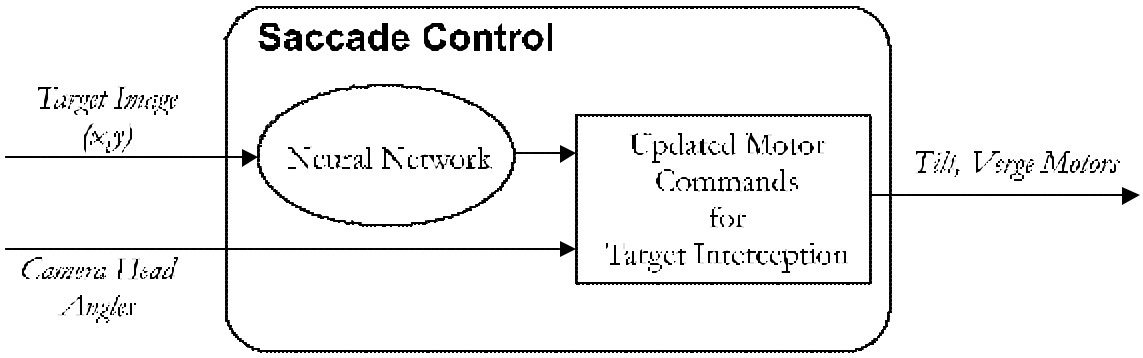


Figure 3: Saccade control

its position and velocity - available to the rest of the system.

2. Saccades

Saccades can be considered as a mapping from target coordinates in the image plane to motor commands [8][9]. A neural network is proposed as a mapping function for this purpose. The main advantages of using a neural network include: (i) it can handle a non-linear problem without the expense of finding a closed-form solution of the plant, (ii) accurate camera calibration is not necessary, and (iii) the system is adaptive through the training process, which can be either off- or on-line. This study examines a neural network design that deals with the types and the parameters of network selections. The neural network module uses target image position (x; y) to calculate how much the tilt and verge motors need to move, potentially at maximum speed, to intercept the target - see Figure 3.

3. Smooth Pursuit

Smooth pursuit is responsible for keeping the target on the fovea. It is a slower

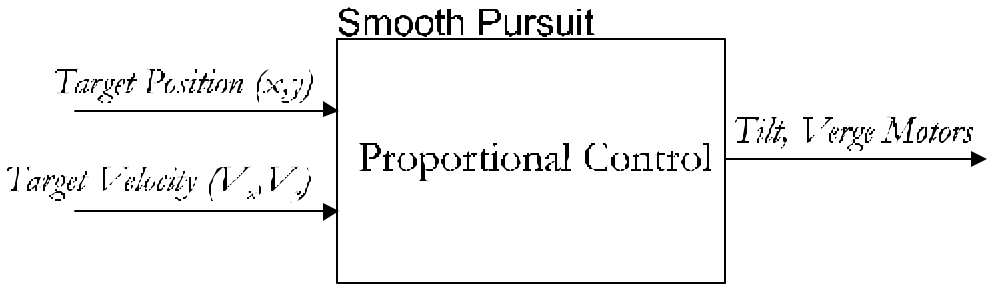


Figure 4: Smooth pursuit control

process than saccades. A smooth-pursuit system can utilize target position and velocity [10][11] [12]. If target position is known, a simple proportional control is sufficient to allow the AVS to track the moving target in the small area of the fovea. The target's velocity can be used to predict its position, so that the camera head can be smoothly controlled to maintain the moving target on the fovea. Since the target's velocity is already available from the motion detection module, the system does not have an additional workload of calculating velocity information. A smooth-pursuit control is depicted in Figure 48.

4. Vergence

Vergence ensures that the left and right eyes fixate on the same target. Vergence control is therefore defined as minimizing disparity between left and right target images from the camera head [13] [14] [15]. This study, hence, examines methods to estimate disparity. Disparity estimation techniques can be categorized into two main groups: correlation- and phase-based. Phase-based disparity estimation has been proven to give better results [15], but it is computationally expensive. The disparity estimation algorithm used in this system utilizes a

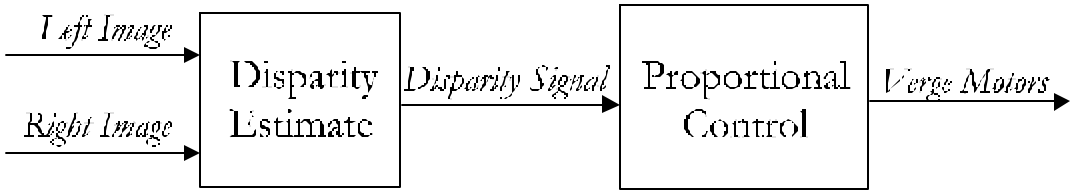


Figure 5: Vergence control

simple correlational technique to reduce computation time, but still yields adequate estimates. This disparity estimate is used as an input signal for vergence control to adjust the camera head in the direction that minimizes disparity between the left and right cameras. The vergence control applies only to disparity on the x-axis, since the left and right cameras are both mounted on the same y-axis. The vergence control diagram can be seen in Figure 5.

5. Vestibulo-Ocular Re°ex

The VOR is involved in stabilizing the eye when head position changes. This re°ex system keeps the eye looking in the same direction as before the head movement. In humans, head motion information is detected by motion sensors in the head and used to generate signals for stabilizing the eyes. A simple idea for designing such a system is to employ special hardware for measuring head velocity and acceleration [8][16]. These devices are known as inertial sensor devices and include gyroscopes and accelerometers. For designing this re°ex, the problem can be stated as: while turning the head (pan motor), keep the eyes (verge motors) looking in the same direction. The proposed system derives the camera head kinematics computationally to properly counter-rotate the eyes while moving the head. Computation of kinematics, however, is not a trivial

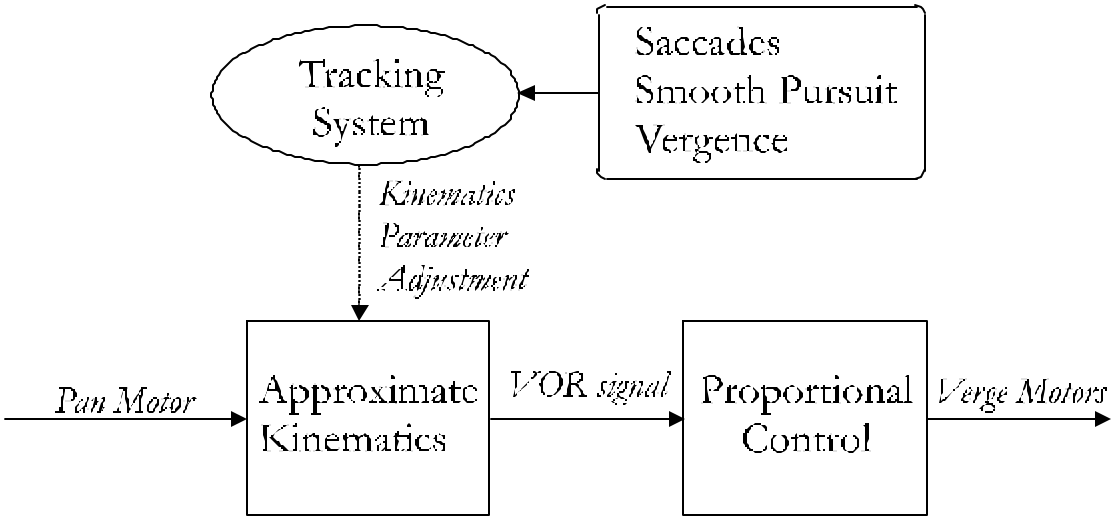


Figure 6: VOR-like control

problem. It requires accurate camera head calibration, which is not within the scope of this study. Instead of directly solving head kinematics, the AVS obtains camera head kinematics with the help of the available visual system: with some modi¯cations in setting up the tracking system along with saccades, smooth pursuit, and vergence control, an equivalent VOR-like movement for compensating verge motors against movement of the pan motor can be achieved. The system diagram is shown in Figure 6.

6. Opto-kinetic Re°ex

The OKR is considered a backup system for the VOR. It functions the same way as the VOR does i.e. stabilizing the eyes while the head is moving. The OKR system, however, uses only visual information to perform such a task. Using an image slip as a clue to generate OKR signal for the head compensation is a key idea of the OKR system [8][16]. A modi¯ed 1-D correlation technique from the disparity estimation module of the vergence system is proposed in order

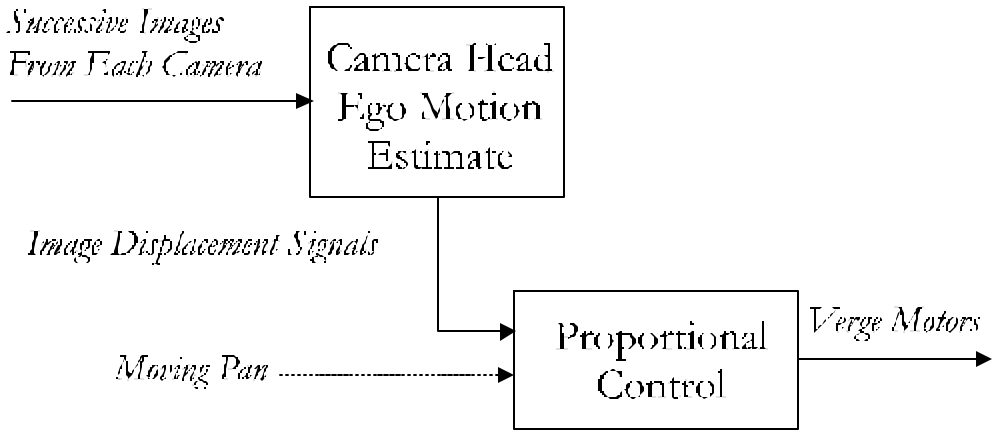


Figure 7: OKR control

to obtain the image slip between successive images from each camera. The displacement signals from both left and right cameras represent the camera ego motion. The OKR controller then utilizes these signals to keep relative velocities of the head and the eyes to zero. The OKR system diagram is depicted in Figure 7.

There are original contributions in the work proposed here. Firstly, without a sophisticated computational hardware binocular head system, a basic four-degree-of-freedom head can improve overall performance with the ¯ve human-like eye movement controls. These control methods are designed at no extra cost, as they do not use special hardware such as high-performance DSP devices for intensive computations, nor gyroscopes and accelerometers for head-motion measures. In addition, the software control system is fully con¯gurable. It allows for creating, editing, inserting or deleting di®erent system con¯gurations. Each component runs independently. This allows the system to perform tasks in parallel as a distributed system.

Secondly, each human-like eye movement control method is unique. Each utilizes visual computations only to reach the desired goal. The unique control methods can be summarized as follows:

For the implementation of saccades, the method used was most closely related to the work of the COG project (see [8]). Their camera head, however, is much more complex than the ISAC head. Moreover, they train a mapping between head-eye coordinates for saccades action using image correlation. In the proposed study, color information is utilized to assist the neural network training process (COG does not use color); this provides faster and more accurate target location than the use of image-patch correlation, such as COG uses.

The smooth-pursuit control method employs target velocity from optical °ow estimation, which is essentially the same concept as that used in much other work (see [13],[9], and [12]). However, the traditional motion estimation and camera-motion compensation di®er from the work proposed here. By using color information, the proposed motion segmentation method can be robustly implemented while maintaining computational simplicity.

Work by [13], [11], and [14] describes vergence-tracking systems using disparity clues. Phase-based disparity estimation is required to obtain disparity clues. 2D methods are, however, nearly impossible to implement on any system without a special hardware module, since they are computationally expensive. Applying color segmentation with a 1-D correlation-based disparity estimate gives a su±ciently accurate disparity estimate for driving vergence control. This method works su±ciently well to implement and to achieve real-time performance on a regular PC-based machine.

Finally, equivalent VOR and OKR control using only visual information has been proposed. Many researchers have presented accurate stabilization models and implemented them with inertial sensor devices (see [8] and [16]). Without any of these special devices, the proposed work utilizes currently available visual systems to achieve VOR-like and OKN motion for head movements. Moreover, the human-assisted learning of VOR parameters for the camera head is uniquely designed and presented.

Summary

The purpose of this study is to design and implement camera head control schemes for the ISAC camera head to achieve ¯ve basic human-like eye movements: saccades, smooth pursuit, vergence, VOR, and OKR. These basic controls would provide human-like camera head motion. The system is integrated as part of the attentional system of the humanoid robot, ISAC, resulting in a smoother, more stable, robust visual system than traditional camera head control methods. All four basic human-like eye movements can be performed with desirable speed and accuracy without any additional special computational hardware. The prototype system is also conveniently portable to other binocular vision platforms.

CHAPTER II

BACKGROUND

Introduction

Active vision is obviously a discipline that concerns itself with the study of human-like vision. Using binocular vision platform with controllable computer vision hardware can improve visual interpretation and performance in interacting with the world. This chapter reviews work related to area of study. These include biological human eye movements, stereo camera head control, and stereo camera head system. The biological human eye movements introduce characteristics of ¯ve human eye movements: saccades, smooth pursuit, vergence, opto-kinetic re°ex, and vestibulo-ocular re°ex. Various researches on camera head control with these human-like eyes movements are discussed. Finally, stereo camera head systems that other researchers have been using are described and compared.

Human Eye Movements

Human eye movements have been studied in several aspects. The study of human eye movements is important in order to understand the characteristics of such movements and build the system that is capable of performing those movements. The detail of human eye system is discussed in [17], [18], and [4].

A three-dimensional model is used for the eye model. These three imaginary axes that intersect in the center of eyeball are vertical, horizontal, and torsional axes (see Figure 8). There are three types of eyeball rotations described as follows.

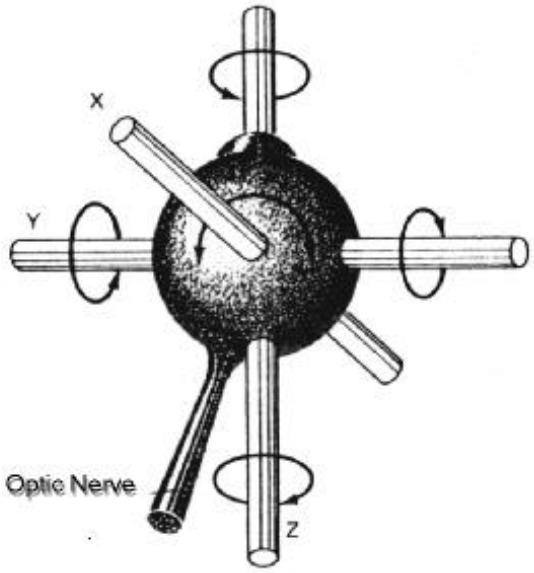


Figure 8: The three principal axes of rotation (shown for the right eye) are the vertical axis (X), which is also the center of motion, the transverse axis (Y), and the anterior-posterior axis (Z) (reprinted from [4]).

- The eyeball rotates around the vertical axis (X) from side to side in toward (adducting) or away from (abducting) movements.
- The eyeball rotates around the horizontal axis (Y) for upward (elevation) or downward (depression) directions.
- The eyeball rotates around an anterior-posterior axis (Z) for either clockwise (intorsion) or counterclockwise (extorsion).

Most eye movements are falling in either conjugate (both eyes move together in the same direction) or disjunctive (both eyes move in opposite directions) movements.

The parts of the brain that control eye movement are called the oculomotor system. The main purpose of the oculomotor system is to keep images centered on the retinal region of greatest visual acuity. Although objects can be observered over a

large visual angle (about 200±), the best visual ¯eld is relatively small arc within 5±. If an object goes o® the fovea, the motor system can correct the slippage by moving the head, body, or eyes. The oculomotor system keeps the fovea on target by means of ¯ve separate neural control systems which are saccadic, smooth pursuit, vergence, vestibulo-oculomotor, and the optokinetic movements. The details of each eye movement are described in the following sections.

Saccadic Movement

Saccades are the ballistic movements of the eyes that jump from one ¯xation point in space to another. They can be intentional, as when asked to \look to the left." or re°exive, as when look at unexpected object that pops into view out of the corner of the eye. A conjugate and ballistic movement of the eyes is generated by the saccadic movement system to bring the fovea onto a target at speed up to 600±-700± /sec. There is a delay of about 0.2 sec between spotting the target and initiating the saccade. After that, the completion of movement takes about 0.05 second. Once the saccadic process has been initiated, the system is unable to make another saccade until 0.2 second later, regardless of target behavior. Saccades are considered to be the voluntary movement and can be made in the dark or with closed eyes.

The saccadic eye movement system depends on both retinal position and eye position. The requirement of movement is that the location of a target in visual space is detected by the retinal (retinal position). The movement also depends on the initial position of the eyes when the object is spotted (eye position). Both retinal and eye position are taken into account before the command for a saccade is initiated. The brain must continuously monitor the position of the eye so that an appropriate

command for saccade can be issued. This command includes information about the direction and amplitude of the saccade.

Smooth Pursuit Movement

The smooth pursuit eye movement system is responsible for keeping the fovea on a target once that target has been located. This system uses di®erent processes for tracking stationary as well as moving objects. Normally, if both eyes and the target are stationary, ¯xation (foveation) can be retained by conscious e®ort, presumably by suppressing any conscious saccades. However, there are continuous, unconscious small movements of the eyes during that steady ¯xation on a target. These movements can be characterized as slow drifts and quick °icks. The drifts move the fovea o® a target, while the °icks are small saccades that bring the fovea to a target after a drift takes it too far away. Because of the °icks, the total net displacement of the target becomes zero.

Consider the viewer remains stationary while a target moves, the brain must calculate the direction and velocity of the image on the retina in order to keep the target on the fovea. The smooth pursuit system can operate only with a target on the retina. It is categorized in the voluntary movement and does not operate in the dark. The di®erences in properties of the smooth pursuit and saccadic eye movements are summarized in Table 1.

Vergence Movement

When the eyes see an object moving toward or away from the head, each eye must move di®erently (disjunctively) to keep the image of the object precisely aligned on

Table 1: Properties of saccadic versus smooth pursuit movements of the eye (reprinted from [4])

Property	Saccadic	Smooth pursuit
Visual Acuity during movement	Poor	Excellent
Target required	No	Yes
Maximum velocity	700±=sec	100±=sec
Velocity under voluntary control	No	No function of target velocity
Stimulus to elicit a movement	Target displacement	Target velocity
Latency	0.2 sec	0.13 sec
Barbiturate sensitivity	Least	Most
Control system	Discrete	Continuous

both foveas. If the object moves closer, the eyes must converge; if it moves away, they must diverge. This operation is performed by the vergence system. The stimulus for this re°ex leads to stereopsis, which allows us to perceive a three-dimensional object in spatial depth. This eye system works together with the pupil- and len-controlling systems in the accommodation re°ex.

This accommodation is another ocular re°ex that allows the eye to focus on objects closer than the far point. When objects move closer than the far point, they go out of focus and shift position relative to the fovea of each eye. In order to recover focus and retain stereoscopic vision, three separate processes take place:

1. The curvature of the lens is increased.

2. The pupil constricts toward 2 mm.

3. The eyes converge.

Optokinetic Movement

The optokinetic re°ex is a backup system for the vestibulo-oculomotor re°ex. The vestibulo-oculomotor re°ex stabilizes the eyes well against the head movements for

duration about 0.5 second. Once the head movement continues for 20-30 seconds longer, the vestibulo-oculomotor re°ex adjusts and can no longer compensate for the head movement. The opto-kinetic re°ex then takes over. This re°ex uses information from retinal signal rather than a signal from the labyrinth to sense head movement. For this kind of re°ex, the eyes automatically track a target and keep the relative velocity of the retinal image at zero. The eye velocity is then equal and opposite to head velocity. When the object being tracked moves out of the visual ¯eld, the eye makes a rapid saccade toward a new target and keep on tracking the object.

Vestibulo-Oculomotor Movement

The vestibular eye movement system is involved with stabilizing the eye against changes in head position. This re°ex system keeps the eye looking at the same direction as it did before the head movement. The process of this re°ex occurs in the membranous labyrinth of the inner ear, rather than within the visual system. This membranous labyrinth detects movements (velocity) of the head along the three axes of space (see Figure 8).

The angular acceleration of the head around the di®erent axis is sensed by each of the three semicircular canals in the membranous labyrinth. These semicircular canals then transmit corresponding signals to neurons in the vestibular nuclei. Higher acceleration (i.e., greater head velocities) produces greater discharge rates along the nerves. The change in head position is accessed by neurons in the vestibular nuclei which integrates information about velocity coming from each canal, and a suitable correction signal is sent to the oculomotor nuclei to stabilize the eyes. These signals persist at the end of a brief acceleration because the velocity is constant. Once the

acceleration becomes zero, the velocity signals fade away in 10-20 seconds. This is, however, much longer than the duration of most head movements. Consequently, information about the velocity and direction of the movement is always available, when the head moves, to enable the brain to determine head position and to generate appropriate compensatory eye movements. The gain of the vestibulo-oculomotor re°ex can be altered by environmental changes: changes in the relationship between corresponding hand, eye, or retinal movements.

Binocular Head: Human-like Control

In the following sections, camera head controls that have capability of human-like eye movements are reviewed. These can be summarized in ¯ve categories: saccades, smooth pursuit, vergence, opto-kinetic re°ex, and vestibulo-ocular re°ex.

Saccade Camera Head Control

Saccade is one of the most important eye movements because human eyes always perform saccades. Saccades are initiated after the target is ¯rst spotted in the visual sensory system. The saccade command contains both direction and amplitude for quickly bringing the target onto the fovea by using the information about a target location and an initial position of the eyes. Saccades are basic behaviors mostly implemented on a binocular system along with smooth pursuit. The obvious main task for saccade is to be used for moving the gaze of the camera head to the target popped up in the visual ¯eld. Many researchers have implemented the saccade mechanisms [19],[20], [8],[21],[22],[10], and [9]. Most of the time, not only saccades are employed in the system, other eyes movement mechanisms are also developed along with saccades

for better performance. This section mainly captures on various kinds of saccade techniques. A general concept of implementation for saccades is ¯rst described. Detail of each technique is then examined.

Wessler [19] implemented a saccade on \Reubens" robot head, originally intended to be a part of the visual system of Cog [8]. He built the 15 £ 13-saccade table which each cell is 32 £ 16 pixels covering the entire image plane. The values in each cell contain the motor command for the pan and tilt motors required to bring the target image onto the fovea. Once target position is located, the corresponding values of the pan and tilt o®sets is determined by using bilinear interpolation of the four surrounding data points from the table. The velocity from the target is also included to predict the actual position of the target in case the saccade is performed. Consequently, the current values from the saccade table, which includes the velocity information of the target, are employed. In this work, the saccade table was initially set up by moving the target window around and recording the servos that bring the target window back to where it started. A correction of the saccade table was also implemented which performed during the normal tracking while there is no any moving target in the visual ¯eld. The system selects a random location and crops an image portion at that location to use as a model. After the saccade has been ¯red, this model is then utilized to match the image in the centered window. If an incorrect saccade has performed, an error which is the distance between this model and the center of the image generates a small correction applied to the corresponding cell and its vicinity in the saccade table.

The previous work later has been developed in [20] and [8]. The saccade function was implemented to produce a change in eye motor position given current eye motor

position and the target location in the image plane. Using a 17 £ 17 interpolated lookup table as a saccade map, the algorithm can be summarized as follows [20][23]:

Initialize map with a linear set of values obtained from self-calibration. Then for each learning trial,

1. Randomly select a visual target.
2. Perform saccade to steer camera head to the target location using the estimate values from the current map.
3. Locate the target in the post-saccade image by using correlation.
4. Train the map using the L_2 o®set of the target and the center of the image as an error signal.

This mapping has been used for saccades to moving targets, bright color, and salient matches in the image at the average of less than 1 pixel of error in a 128 £ 128 image per saccade after 2000 trials.

In [10] and [9], processes of how saccade is performed are not described. The detail, however, was focused on target motion estimation which provide information for the saccade and smooth pursuit modules. The velocity of the target was extracted coarsely from optical °ow computation and its position was predicted using Kalman ¯lter to compensate for a vision processing latency.

Bradshow et. al use gradient-based methods to derive components of the motion ¯eld. Foreground regions are segmented using a grouping process. Finally, a constant motion ¯eld is applied to compensate for the aperture problem. By predicting the position and velocity of the target, a demand to the servo-controller is achieved ahead

of time while the visual processing during the saccade is not utilized. The demand to the servo-controller is computed by extrapolation of these position and velocity. The extrapolation provides a smoother transition for the next state for a smooth pursuit module. The end of a saccade is acknowledged by using the feedback distance from the head/eye platform. The criterion for the end of a saccade is when the di®erence between the actual head position and the command from the saccade module is below a threshold.

Smooth Pursuit Camera Head Control

Smooth pursuit process is responsible for keeping a target on the fovea after the target has been located. It is considered the slower process than the saccade and works on such a narrower ¯eld of the fovea (higher resolution). The key idea of smooth pursuit can be roughly concluded as by extracting the target position and velocity with additional help from a prediction module. The smooth pursuit then uses these position and velocity to steer camera motors. Most of the time the target acceleration is considered constant. A model of the smooth pursuit system proposed in [24] and [25] suggested that the smooth pursuit should response to the retinal slip signal of the target. Even though, there has never been any study of biological visual systems on what kind of retinal slip signal should be responded by the smooth pursuit system [26]. The smooth pursuit system, hence, mainly deals with extracting the moving target information out of a complex scene and controlling the cameras using this information. The following sections discuss the computer vision literature of smooth pursuit systems. All works presenting here can be categorized into three groups which are based on image correlation, binocular disparity and motion analysis

approaches.

Smooth Pursuit Control Based on Correlation Method

Using correlation methods allows system to locate the target portion in the next image frame. This requires a priori knowledge of the target, typically called a model, to perform matching operation of this model within a search window. The matching operation gives a location of the image portion in the new image that is best match with the model. Normally, normalized image intensity is a quantity for the matching operation. This method, however, su®ers from problems of target orientation, and scaling. Note that using the correlation method, content of the target image is not extracted or segmented out of the background.

Brooks et. al. [8] have implemented smooth pursuit system on COG using correlation method to locate the target. The target is ¯rst located from where the motion occurs. That target is then used as a model to locate its position from the best correlation value in the next frame. They used 7£7 model image to search in 44£44 window within 64£64 image. The model is updated in every frame. The distance of the target from the center in the new image gives the motion vector which is used as a velocity command to control the motors for the smooth pursuit control.

Smooth Pursuit Control Based on Binocular Disparity

Disparity is very useful clue in visual system. The recent researches on using binocular disparities have shown a signi¯cant improvement of the performance of stereo active vision systems [27][28] [21][22][15][14]. For the binocular smooth pursuit system, zero disparity ¯lter techniques have been used to locate portions of the

images that have zero stereo disparity i.e. objects are ¯xated. For this reason, using binocular cues (disparity) enables the system to track the moving target without a priori knowledge about the target.

Coombs and Brown ([11],[29],[27], and [26]) have addressed a goal of smooth pursuit behavior as maintaining image of the visual target to be steady and centered in the camera image. Consequently, the system must be able to resolve the retinotopic position and retinal slip of the target. Their system requires only initial position of the target without any knowledge of recognizing the target. They employ disparity ¯ltering to extract the target location. Once the target is located, the PID controller is used to drive the camera head to match between the target and camera head position and velocity. The ® ¡ ¯ ¡ ° ¯lter has been utilized to predict the target position under assumption that target has a constant acceleration.

Rougeaux et.al. [28][30][31] proposed a novel control strategy based on the computation of virtual horopters to track the moving target. The horopter is a circle which has the set of zero disparity points projected from left and right images on it (see Figure 9). If the target lies on this horopter, it then can be segmented out from other objects or background by suppressing contents of non-zero disparity (which stay o® the horopter). This segmentation technique was implemented in [32]. The virtual horopter is estimated to locate the object moving across the horopter by using zero disparity ¯lter. The zero disparity ¯lter is obtained by applying the logical AND operation between a blurred and binarized version of left and right vertical-edge images. The remaining pixels in the images have zero disparity. The virtual horopter is then generated by horizontally shifting the right image by a certain amount of pixel. Direction of the shift (left and right) indicates that the target is inside or outside the

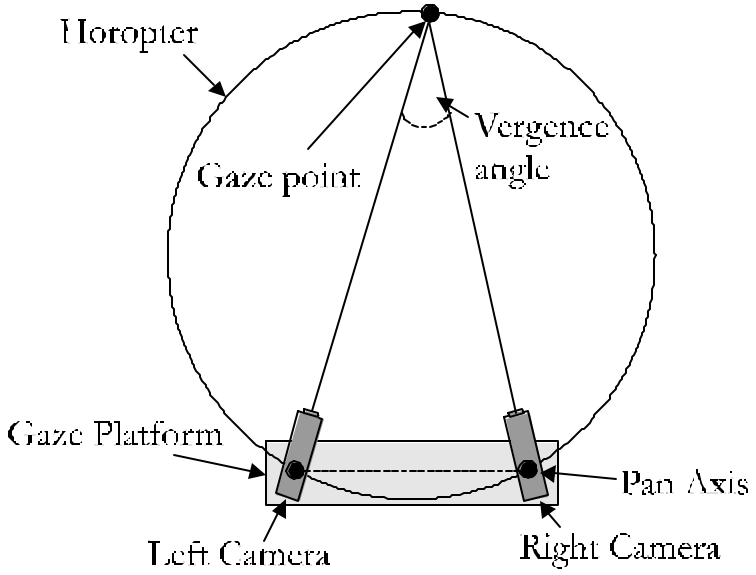


Figure 9: Horopter tracking (reprinted from [30])

horopter. The amount of shifting is then used to control the camera head to track the target.

Smooth Pursuit Control Based on Motion Analysis

The most popular approach for a smooth pursuit is to utilize the information of the target from optical °ow. A °ow image provides motion components of the moving objects under observation. This relates to many problems in computer vision ¯eld of study. For a smooth pursuit camera head control, a problem is said to be using motion information available from optical °ow of the moving observer to track the moving target.

There are two main approaches for computing optical °ow: matching-based (correspondence) and gradient-based approaches [7]. Using optical °ow from moving observer becomes more challenge since the camera movements cause the entire scene

appear to move. Consequently, the smooth pursuit system must be capable of computing optical °ow patterns from camera movement and egomotion. This, however, can be quite complicated.

Murray et.al. ([10] and [9]) apply optical °ow to extract the target position and motion and track moving objects. Information from optical °ow provides velocity of the target, which is utilized for making predictions of the target. They also manipulate the target velocity for a control process. The °ow image is calculated in both coarse and ¯ne resolution. For the smooth pursuit, the °ow image is computed in the periphery with high resolution of the fovea (¯ne resolution). This gives both velocity and position of the target to the camera controller move to stabilize against the image motion, i.e. to null the motion observed in the image. Their optical °ow is based on works from [33] and [6]. The camera head motion is compensated using a known camera head velocity. The detail, however, was not reported.

In [12][34][35], smooth pursuit works, which are similar to the previous work by Murray, has been proposed. The di®erences were the camera head structure and optical °ow computation. Their camera head [36] is much more complex than the one used in Murray's work. Also, the optical °ow was calculated using a multi-resolution structure. In this work, the target is only required not to be far from the ¯xation point of the head. During the pursuit process, velocity control of the camera head is used instead of position control. The smooth-pursuit controller maintains the target on ¯xation point by assuming that it always located on the horopter. Two Kalman ¯lters were used for the estimated image motion velocities. The background image motion is estimated using the information of lens' focal length and velocity induced by each joint of the camera head. This can be computed using the following

equations:

$$v_u = \frac{v_x \cdot f_x}{z} - \frac{v_z \cdot f_x \cdot x}{z^2} \quad v_v = \frac{v_y \cdot f_y}{z} - \frac{v_z \cdot f_y \cdot y}{z^2}$$

where $(f_x; f_y)$ represents the focal length of the lens in pixels and $V = [v_x\ v_y\ v_z]^T$ represents the velocity of the point $P = [x\ y\ z]^T$ in the camera coordinate system due to the egomotion of the camera head. The velocity V is measured from joints of the camera head.

Another similar work has been done by Rougeaux et.al. [13] [22]. They utilized the velocity from optical °ow for smooth tracking the moving target. Using their special design space-variant lenses, the motion of in the image can be modeled and used in computing °ow image. The simple segmentation of the target motion in the image plane is based on a gradient-based optical °ow compared to the joint velocities from the camera head for egomotion compensation. The Kalman ¯lter was also implemented along with target motion estimation. Similarly, Manzotti et.al. [37] also utilized space variant device (lenses) to calculate for °ow image and use this velocity information to perform smooth pursuit.

Vergence Camera Head Control

Vergence keeps both eyes ¯xated on the same object. On the other words, it points the optical axes of the two cameras to a selected ¯xation point. A straight-forward and simple way to do this is to select a ¯xation point in di®erent cameras and control each camera to have the ¯xation point projected onto its optical axes. The remaining problem is how to ensure that the two cameras select the same world point of the target. The most useful clue utilized as the input to vergence control is disparity. It generates the error signal of vergence system. There are many researchs

investigated methods to estimate disparity. These methods can mainly be categorized into two types: matching- or correlation-based, and phase-based disparity estimation. The matching- or correlation-based disparity estimation has some drawbacks in several aspects. Firstly, it lacks of robustness for tracking objects moving over highly complex backgrounds. Secondly, the error can occur when there is the repeated pattern. Thirdly, if the target in one camera is occluded, no corresponding matches would be found and lead to unstable state. Finally, it is computationally expensive. The phase-based disparity estimation, in contrast, overcomes these drawbacks of the matching/correlation-based algorithms. Better performance could then be achieved from this type of disparity estimation.

In this section, the overview of the disparity estimation is described. The algorithms to estimate disparity are then examined, including matching/correlation-based and phase-based disparity estimation. Other methods for computing disparity are also studied. The final section then investigates the applications of vergence control.

Matching/Correlation-Based Disparity Estimation

Matching/Correlation-based disparity estimation can be measured under the assumption that the left and right images are simply shifted versions of each other [38]. By measuring the shift between two regions in the image, the corresponding correlation of these regions is maximum when they are shifted the correct amount with respect to each other. The disparity at point x can be achieved by:

$$Max_{disp} \int_{Z} W(z)L(x + disp + z)R(x + z)dz$$

where W is a window function used to localized the disparity measure, L and R are the left and right images, respectively. An example of designing the window function W is to provides a blurring e®ect. The performance of the algorithm for noisy images could then be signi¯cantly improved.

The major advantage of correlation techniques is its high density of responses. This correlation-based disparity estimation, however, has been shown to have a number of problems. These can be summarized as follows:

- Perspective projection - this can cause an non-ideal shifted version of the images with respect to each other.
- Interocular illumination di®erences - the same surface patch appears with different brightness from two di®erent angles of view.
- Disparity gradients - an imperfectly shifted version of the left and right images seen by the left and right eyes.
- Too little or too much detail in an image - the correlation is proven to be highly sensitive to either too little or too much structure in an image.

There are other techniques have been investigated in order to improve the correlation-based disparity estimation. For example, the normalized cross-correlation which is the correlation of zero-mean signals and normalized by their vaiances [39]. In this work, Witkin et al. employ normalized cross correlation to various blurred versions of the image in order to overcome basic problem. By providing pre-processing e.g. blurring window functions, some problems may be resolved. However, it could lose a genuine characteristic of the correlation operators.

In the next section, a correspondence algorithm is described for achieving a better performance of disparity estimation.

Correspondence-Based Disparity Estimation

There are three steps for measuring disparity based on correspondence as follows [38]:

1. Monocular features extracted from a stereo image pair are performed separately.
2. These features in one image are matched with corresponding features in the other image.
3. The incorrect matches (false target) from step 2 are removed.

One common option of feature is a zero crossing in a band-pass version of the image. The matching process does not always produce only the correct matches. Consequently, in the design of a traditional correspondence-based algorithm, there are three main concerns: (i) the choice of image features, (ii) the choice of matching criteria, and (iii) the process to eliminate the incorrect matches. The features, with respect to each main issue, can be summarized as: (i) correspond to surface properties, (ii) produce correct matches that are fairly dense, and (iii) distribute over an image such that false matches can be relatively eliminated. The con°ict, however, occurs between the density and the false target detection. The more density, the more di±cult for the false matching detection.

The simplest feature for the algorithm is the image intensity itself. Other features such as high contrast points, corner points and edges have been employed. The fundamental problem, however, still remains e.g. the feature may not exists or detectable

in one image at all.

Both correlation- and correspondence-based algorithms have suffered in many aspects. Correlation responses may have a high density, but they are highly sensitive to either too little or too much structure in an image. Correspondence responses are more robust, but they are sparse, and some post-processing must be required in order to "fill in" the missing responses. This shows a complement relationship between these two algorithms as Jenkin noted:

> Correlation produces a dense set of responses but has difficulty with constant or rapidly changing structure and with interocular image differences, while correspondence avoids some of these problems by considering the image at different scales but then fails to obtain a dense set of responses by matching only sparse tokens.

Other disparity estimation algorithms have been proposed to overcome problems that both correlation and correspondence have encountered. One such a well-known method is phase-based disparity estimation which is examined in the next section.

Phase-Based Disparity Estimation

Phase-based disparity estimation has been proven to be the most robust method for measuring disparity. Many researchers have studied and proposed several algorithms to achieve a better performance [11],[13],[38],[40],[41], and [42]. In this section, a general concept of the phase-based disparity estimation is briefly described. Then the reviews of well-know methods are provided. Examples of the algorithms are also displayed.

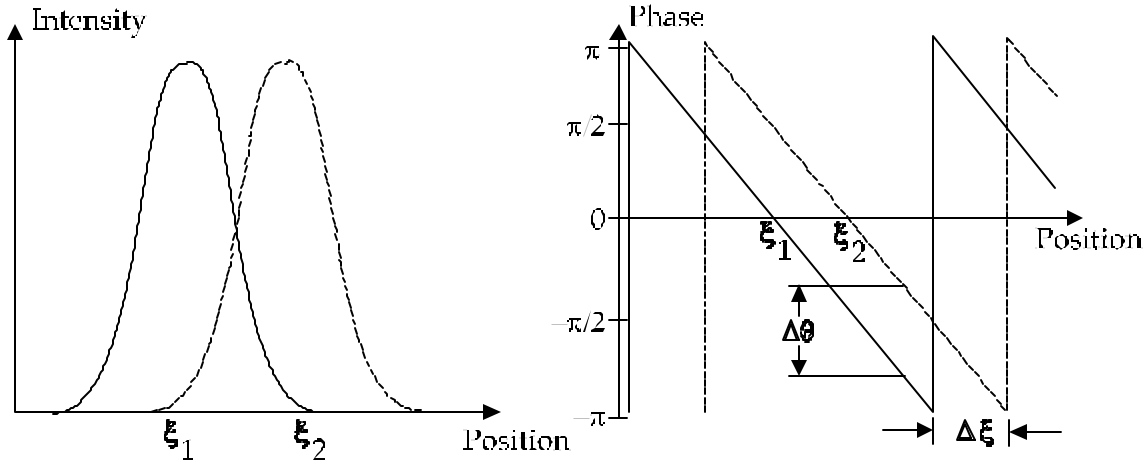


Figure 10: Relationship between position and phase di®erence (adapted from [15])

The concept of the phase-based disparity estimation is reviewed from [15]. Local displacement of two images can be estimated using the relationship between phase and position. Figure 10 shows an example of the estimation of the line displacement using phase di®erences. The left and right images contain a line located at $»_1$ and $»_2$, respectively. From the phase curve corresponding to the two images, the relationship of the displacement ¢» can be estimated by computing the phase di®erence ¢µ and the slope of the phase curve dµ=d». The subpixel displacement produces a phase shift providing phase di®erences with subpixel accuracy. Some techniques are utilized to simplify and stabilize the algorithm. For examples, a multi-resolution representation of the images is used to reduce the computational time. Edge images are employed along with intensity images in order to reduce the in°uence of singular points. Disparity measurements from both intensity and edge images can be weighted together to obtain the disparity measure between the shifted image. Also, the e®ect of a DC component in the disparity ¯lter is decreased using the edge images.

The processes of disparity estimation begin with shifting the input images by half

the distance each. This procedure reduces the disparity since the left and right images are shifted toward each other as de¯ne by:

$$x_{sL}(»_1; »_2) = x_L(»_1 + 0:5¢»; »_2)$$

$$x_{sR}(»_1; »_2) = x_R(»_1 ¡ 0:5¢»; »_2)$$

In order to obtain the phase, the images are convolved with a complex ¯lter. Different kind of ¯lter yields di®erent result due to the di®erent characteristic of the ¯lter. The well-known disparity estimation method using Gabor ¯lters is presented in [43]. The responses from the ¯lter can be represented by a complex number. A con¯dence measure of the ¯lter response is measured by the magnitude of the ¯lter. The argument of the complex number is the phase in the signal. Let the complex numbers z_L and z_R represented the responses from the phase ¯lter for the left and right image respectively. The ¯lters are normalized such that $0 \cdot jjz_{L,R}jj \cdot 1$: By calculating $d = z_L z_R^¤$ yields a phase di®erence measure,

$$jjdjj = jjz_L z_R jj; \qquad 0 \cdot jjdjj \cdot 1$$

$$\arg(d) = \arg(z_L) ¡ \arg(z_R); \quad ¡¼ \cdot \arg(d) \cdot ¼$$

where ¤ denotes complex conjugate.

The magnitude jjdjj indicates a reliability of the phase di®erence. It is large if both ¯lter magnitudes are large. If the images have a very little content (e.g. white plane), the magnitude will be zero and its argument is unde¯ned. The con¯dence values are then crucial for calculating a stable phase di®erence. One of the con¯dence values can be computed as follow:

$$C_1 = \left(\frac{2jjz_L z_R jj}{jjz_L jj^2 + jjz_R jj^2}\right)^{¶◦}$$

This function couples both filter magnitudes and the absolute value which represent both the similarity and the strength of the signal. If the magnitudes are almost the same ($||z_L|| \approx ||z_R||$), the confidence value is also the same ($C_1 = \sqrt{||z_L z_R||}$). It is likely that the wrap-around problem may be occur if the phase difference is very large. This will indicate a wrong sign of the disparity. A lower confidence value is then introduced for the very large phase differences as defined by:

$$C_2 = C_1 \cos^2\left(\frac{\arg(d)}{2}\right)$$

In order to achieve the phase difference correctly, the phase derivative should be estimated locally. If $z(x)$ is a phase estimate at position x; the local frequency can be approximated using the phase difference to the left and right of the current position:

$$f_{L-} = z_L(x-1)z_L^*(x)$$

$$f_{L+} = z_L(x)z_L^*(x+1)$$

$$f_{R-} = z_R(x-1)z_R^*(x)$$

$$f_{R+} = z_R(x)z_R^*(x+1)$$

The local frequency can then be estimated by $\omega = \arg(f_{L-} + f_{L+} + f_{R-} + f_{R+})$: Using this local frequency, the disparity in pixels can be calculated by:

$$\phi x = \frac{\arg(d)}{\omega}$$

The phase difference is said to be unreliable if the local frequency is zero or negative which the confidence value is then set to zero. The confidence value is updated by:

$$C_3 = \begin{cases} C_2 \|f\|^{\frac{1}{4}} \sum \frac{f_i}{\|f_i\|}\|^4 & \text{if } \omega > 0 \\ 0 & \text{if } \omega \le 0 \end{cases} \qquad \text{where } i \in \{L-; L+; R-; R+\}$$

If both edge and intensity images are used, the disparity and con¯dence values are calculated separately and then combined to give the total disparity and its con¯dence value.

$$¢» = \frac{C_{g3}¢»_g + C_{e3}¢»_e}{C_{g3} + C_{e3}}$$

$$C_{total} = jjC_{g3}e^{i\frac{arg(d_g)}{2}} + C_{e3}e^{i\frac{arg(d_e)}{2}}jj$$

Note that $arg(d_{e;g})$ is divided by two in order to ensure that C_{total} is large only for $arg(d_g) ¼ arg(d_e)$ and not for $arg(d_g) ¼ arg(d_e) § 2¼$: Subscript g and e denote gray level and edge image values respectively.

Once the disparity estimate and its con¯dence value are computed in each shift, the disparity accumulator is updated. The value to be added to the disparity accumulator is an adjustment toward the ¯nal disparity. The accumulator can then be computed as:

$$¢»_{new} = ¢»_{old} + ¢»$$

and the con¯dence value:

$$C_{new} = \left(\frac{\sqrt{C_{old}} + \sqrt{C_{total}}}{2}\right)^2$$

In [40], a Fourier phase-based disparity extraction was purposed. The vergence system was designed for real-time applications. Consequently, an algorithm that estimates a single disparity in a small ¯xed amount of time is required. Their algorithm is motivated by the property of the Fourier transform that a translation in spatial domain will proportionally result in a translation in frequency domain. The calculation of the phase di®erence of the two images then provides the translation of the object in the two images and the actual disparities can be determined.

The images used in their system are binary images. This reduces many problems introduced by similar methods, such as, intensity di®erence and local occlusion. The advantages of this approach are summarized as follows:

1. The images are employed as binary images which the ideal assumption of the shifted version for both images from left and right camera can be obtained.

2. The exact position of objects in the two images is not required in prior in order to determine the disparities. By keeping the object within the region of interest (in the window), accurate disparity estimation can be achieved.

3. The resulting disparity is totally a function of the image properties. The disadvantages of search and peak-¯nding [38] can be avoided.

4. The method provides a robust estimation of overall disparity. The problems of local occlusion and local intensity changes are not a®ected.

5. The method is simple which makes it computationally inexpensive. This approach is then suitable for many real-time applications.

In this method, once an object is centered in a single camera, that means the phase angle is controlled to be ¼: Also if an object is completely inside the image window, the linear relationship between the object displacement and the corresponding phase di®erence of fundamental frequency can be achieved. The position change and the phase shift of fundamental frequency is satis¯ed the following condition:

$$¢position = \frac{window\_size}{2¼} ¢phase$$

This equation is derived directly from the translation property of the Fourier transform as follow:

$$f(x - x_0; y - y_0) \quad (\quad) \quad F(u;v)\exp[-j2\pi(ux_0 + vy_0)/N]$$

where the only fundamental frequency is considered ($u = v = 1$) and N is the window size.

Thus, if the right image $R(x;y)$ is considered to be a "shifted version" of the left image $L(x;y)$, by calculating the fundamental frequency phase change in these two "consecutive" images, the disparities x_d and y_d can be determined. The Fourier phase in conjunction with the projection method is applied in order to achieve faster processing (1 dimension) and to separate disparities along each dimension (x_d and y_d). The projection of $F(x;y)$ along y-direction and x-direction are defined by:

$$F_y(x) = \int F(x;y)dy \quad \text{and} \quad F_x(y) = \int F(x;y)dx$$

The vergence disparity extraction can then be summarized as follows:

- The windows are selected such that both projections of the target are within in both left and right image.
- The object of interest is segmented from background to obtain binary images.
- Each of binary image is projected onto x- and y-axis to obtain four signal sequences, i.e. $L(i_L); L(j_L); R(i_R)$; and $R(j_R)$:
- The phase angles of Fourier transform of $L(i_L); L(j_L); R(i_R)$;and $R(j_R)$ are calculated and denoted as $\mu_L^i; \mu_L^j; \mu_R^i$; and μ_R^j; respectively. Phase differences between the corresponding pairs are calculated as follows:

$$\Delta\mu^i = \mu_R^i - \mu_L^i$$

$$\phi\mu^i = \mu_R^j - \mu_L^j$$

The corresponding spatial difference of the target object in the two images is then can be calculated from:

$$x_d = \frac{h}{2\tan\frac{1}{4}}\phi\mu^i$$

$$y_d = \frac{h}{2\tan\frac{1}{4}}\phi\mu^j$$

In this research, the influence of the location and size of the window, the segmentation technique, and the influence of the illumination from the environment are also examined.

Rougeaux and Kuniyoshi [13] use phase difference estimation in their zero-disparity filter. The estimator is calculated from the Fourier domain in conjunction with a correlation measurement in the spatial domain to extract features with small horizontal disparity. The algorithm can be concluded as follows:

- The left ($l_{x;y}$) and right ($r_{x;y}$) images are convolved with a complex filter to produce two signals $L_{x;y}$ and $R_{x;y}$. These signals are related to each other by:

$$L_{x;y} = e^{j\omega_{x;y}D_{x;y}}R_{x;y}$$

where the instantaneous frequency $\omega_{x;y}$ is defined by:

$$\omega_{x;y} = \frac{d(\arg(L_{x;y}))}{dx}$$

- The disparity $D_{x;y}$ can be calculated using the equation:

$$D_{x;y} = \frac{\arg(L_{x;y}R^*_{x;y})}{\omega_{x;y}}$$

which $R^*_{x;y}$ is the complex conjugate of $R_{x;y}$.

- A confidence value associated with the disparity signal can be computed to overcome the effects of the aperture problem by:

$$C_{x;y} = |L_{x;y}||R_{x;y}|$$

- A confidence weighted disparity map is calculated by convolving locally the product of the disparity and the confidence value with a Gaussian filter $G_{x;y;\sigma_1}$:

$$Dc_{x;y} = \frac{\sum_{x;y} C_{x;y} G_{x;y;\sigma_1} D_{x;y}}{\sum_{x;y} G_{x;y;\sigma_1} D_{x;y}}$$

- The output of the zero-disparity filter along with a correlation measure in the spatial domain is defined as:

$$ZDF_{x;y} = \frac{C_{x;y} e^{-\frac{Dc_{x;y}}{2\sigma_2^2}}}{\sum_{x;y} (I_{x;y} - r_{x;y})^2}$$

where σ_2 is reflecting the tolerance in the disparity measurement error.

- The target disparity d_t and position $(x_t; y_t)$ in the image plane can be computed as:

$$d_t = \frac{\sum_{x;y} ZDF_{x;y} Dc_{x;y}}{\sum_{x;y} ZDF_{x;y}}$$

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \frac{\sum_{x;y} ZDF_{x;y} \begin{pmatrix} x \\ y \end{pmatrix}}{\sum_{x;y} ZDF_{x;y}}$$

The position $(x_t; y_t)$ and the disparity d_t are employed to estimate the target motion along with the horopter (vergence) for a real-time binocular tracking of the ESCHeR head (see stereo camera head tracking system).

Another example of phase-based parity detection was studied in [41]. This algorithm was also applied in [44] and [14]. The outline of the method can be summarized as follow.

From the basic concept of the phase-based algorithm, the left and right stereo images are convolved with a complex filter, such as Gabor filter. The filter output can then be retrieved from the complex phase difference. The local shift between the two images is linearly proportional to the local phase difference. The disparity, thus, is obtained at each point with estimation. In this algorithm, the disparity estimation is considered on 1-dimentional (horizontal). Consequently, the relationship between the left and right images can be displayed as:

$$l(x;y) = r(x + \phi x; y)$$

where $l(x;y)$ and $r(x;y)$ are defined the left and right image, respectively. ϕx is a constant horizontal disparity (shifted amount in x-axis).

At any point $(x_0;y_0)$ and a particular complex filter $f(x)$ (presented here are Gabor filter and Fast filter) the convolution of $l(x;y)$ and $r(x;y)$ with $f(x)$ can be expressed as:

$$\begin{aligned} conv_r(x_0;y_0) &= \int r(x;y_0) f(x - x_0) dx \\ conv_l(x_0;y_0) &= \int l(x;y_0) f(x - x_0) dx \\ &= \int r(x + \phi x; y_0) f(x - x_0) dx \end{aligned}$$

The relation in frequency shift between $conv_r(x_0;y_0)$ and $conv_l(x_0;y_0)$ is:

$$conv_l(x_0;y_0) = e^{j \omega_0 \phi x} conv_r(x_0;y_0)$$

The disparity ¢x can be approximated by:

$$¢x ¼ ¢©=!_0$$

$$¢© ´ \arg[conv_l] ¡ \arg[conv_r]$$

This is, however, satis¯ed only for ¯lters of in¯nitesimal bandwidth.

A con¯dence threshold value is also de¯ned in order to observe the reliability of the disparity estimation. In this work, the con¯dence value is de¯ned as follows:

$$conf = \frac{mag[conv_l] £ mag[conv_r]}{mag[conv_l] + mag[conv_r]}$$

The Gabor ¯lter and Fast ¯lter were employed as the complex ¯lter f(x). One-dimensional Gabor ¯lters which minimize the product of spatial width and bandwidth have the functional form as follows (note that it is crucial that both the spatial width of the ¯lters and the spatial frequency bandwidth are small in the phase-based algorithm):

$$g(x ¡ x_0) = e^{\frac{¡(x ¡ x_0)^2}{2¾^2}} ¢ e^{j!_0(x ¡ x_0)}$$

$$G(! ¡ !_0) = e^{\frac{¡(! ¡ !_0)^2}{2¿^2}} ¢ e^{jx_0(! ¡ !_0)}$$

x_0 : The spatial location of the ¯lter.

$!_0$: The central frequency of the power spectrum.

¾ : The spatial half-width of the ¯lter.

¿ : The half-width in frequency domain (¾¿ = 1).

Because Gabor ¯lter is computationally expensive especially for the calculation of the 2-dimensional Fourier transform, the fast ¯lter was then introduced. The fast

filter is formed as a pixel-wise complex filter $p(n)[n : 0; 1; 2; :::]$:

$$Im[p(n - n_0)] = \begin{cases} -t; & \text{for } n = n_0 - 1 \\ t; & \text{for } n = n_0 + 1 \\ 0; & \text{for } n \neq n_0 \pm 1 \end{cases}$$

$$Re[p(n - n_0)] = \begin{cases} -t; & \text{for } n = n_0 \pm 2 \\ 2t; & \text{for } n = n_0 \\ 0; & \text{for } n \neq n_0 \pm 2; n_0 \end{cases}$$

n_0 : The spatial location

t : The amplitude scale

The frequency of this filter is $\omega_0 = \frac{\pi}{2}$ [radian/pixel].

The disparity estimation was applied to a vergence control on the KTH head-eye system (see stereo camera head tracking system).

Additionally, Maki and Uhlin [44] have utilized a disparity selection method along with this disparity estimation in order to select the disparity corresponding to the target in binocular pursuit from multiple disparity environment. The basic concept is to slice the scene using the disparity histogram. Only the target, thus, remains. The slice is chosen around a peak in the histogram using prediction of the target disparity and target location achieved by back projection. The choosing process is done along a useful information obtained from a denser disparity map. This map is created by a confidence-weighted disparity applied in a coarse-to-fine strategy.

Coombs developed a method based on Cepstral filtering for disparity error measurement [11]. The objective of using the Cepstral filtering is to obtain phase of the images. This process, however, requires a heavy computation which is not practical for a real-time feature.

Other Methods for Disparity Estimation and Vergence Control Applications

In [31] and [28], a zero disparity ¯lter was developed to cope with vergence error. The zero disparity ¯lter is extended with a simple algorithm based on the computation of virtual horopters. Objects which stay on the horopter are said to have a zero disparity. Consequently, these objects can be picked up easily while other objects are suppressed with their features of non-zero disparity. The object target is simply isolated by the following processes:

- Vertical edges within the image windows are extracted from both left and right images.
- These edge image windows are then blurred and binalized by a prede¯ned threshold.
- A logical AND operation is then performed between left and right binary images.

The position estimation of the target can be achieved by virtual horopters. This virtual horopter is the horopter generated by shifting horizontally the right image by a certain amount of pixel. A small shift (s pixels) of the right image are almost equivalent to a small virtual rotation ($\phi\mu_r$) of the right camera. $\phi\mu_r$ can be estimated by:

$$\phi\mu_r = \tan^{i\,1}(\frac{s}{f_x}) \text{and} f_x = \frac{h_{pixel}}{h_{width}}$$

where f is the focal length, h_{pixel} is the pixel width of the images, and h_{width} is the horizontal length of the image planes.

The target position is estimated using the center of gravity of zero disparity ¯lter outputs. By counting the total number \N" of pixel and using the average width of

all edges in the zero disparity filter output, a pixel shift "s" can be obtained by:

$$s = w \times (1 - \frac{N}{N_{max}})$$

where N_{max} is the pixel number in the ideal case, (target is on the horopter).

This pixel shift s is used to control the camera position for a gaze holding task which involves maintaining a gaze point on a moving visual target from a moving gaze platform. The position estimation, however, could suffer in the following reasons: (i) there are dense vertical edges in the background, (ii) another object with a fairly large number of vertical edges comes into the 3D window area, and (iii) a target object is significantly occluded.

In active vision tracking control applications, after the disparity is measured as the input to the system, the difference between it and the output reference is calculated. The output is then controlled by an incremental amount. This process is repeated until the measured difference is zero or within some tolerance. This traditional strategy is referred to as a static look and move control strategy. Applications of disparity detection for vergence control are introduced in [45] and [29]. The general goal of vergence control is to compensate the disparity between the cameras to reduce the vergence disparity between them. The input for the control is the vergence error extracted from the disparity while the output is the parameters of the degrees of freedom for the cameras.

In [45], the only non-dominant camera (slave camera) is controlled. The techniques in fixation point selection and the phase-based vergence disparity extraction algorithm are presented. They used a feedback information prediction and dynamic vision-based self-tuning control strategy to achieve vergence control. A dual sampling rate system,

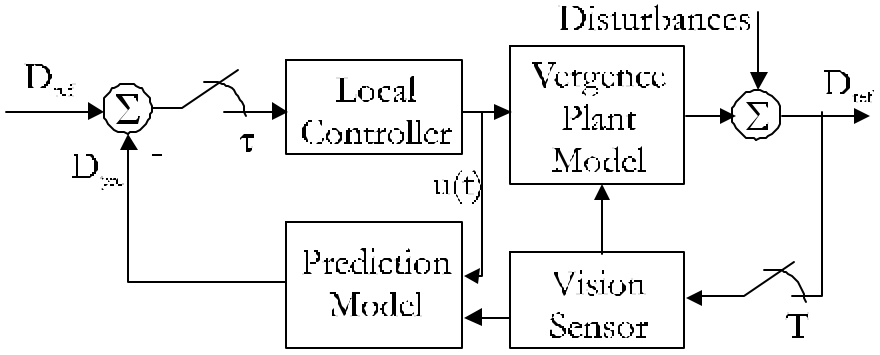


Figure 11: Dual sampling rate vergence server system (reprinted from [45]).

i.e. the control and image processing sample rates, was employed in stead of a time-delay strategy. The dual sampling rate vergence system can be depicted as in Figure 11.

D_{ref} is the estimated vergence disparity to be compensated. It is also the reference input to the system to be compare with the system output, D_{pre}. A prediction model is developed to predict the system output due to the system output cannot be obtained during image processing. A recursive least square technique is employed to identify parameters of both the prediction model and the real system. A minimum variance regulator approach is utilized for the control scheme. The control signal u(t) is synthesized and applied to the motor based on the reference input and the prediction output to minimize input/output variance.

Coombs and Brown [29] apply the cepstral ¯lter and phase correlation technique to a binocular gaze holding system (see [11] for details of a disparity estimator). They use a straight forward proportional derivative (PD) controller in cascade with the eye motor for a feedback loop as can be seen in Figure 12 where μ_t is target vergence angle, μ is vergence angle, μ_e is vergence error, and $\dot{\mu}_c$ is motor velocity control signal.

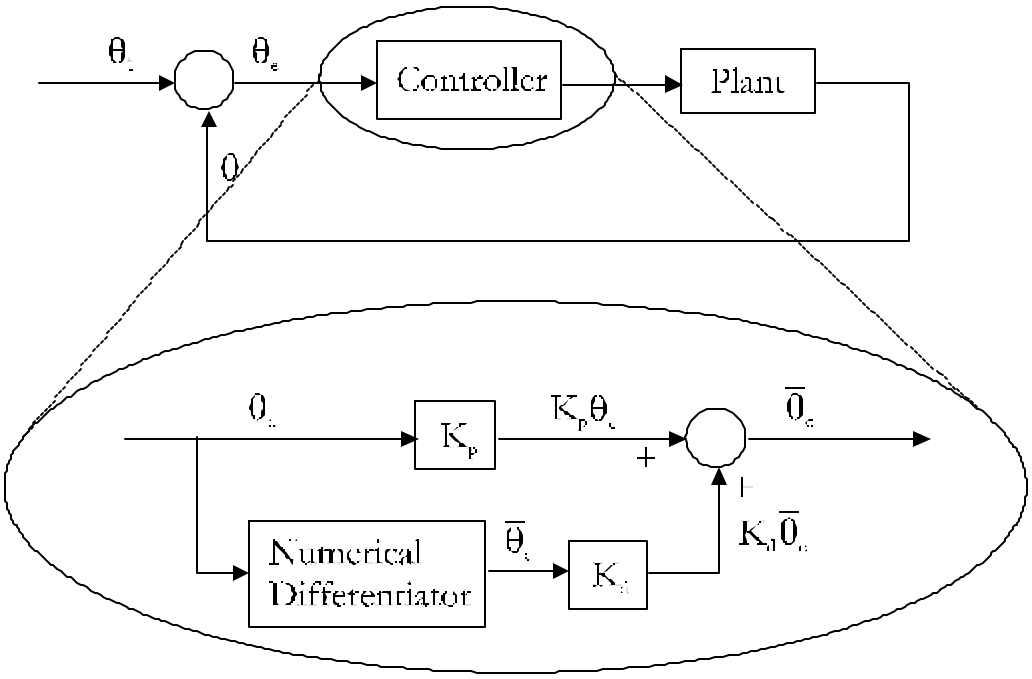


Figure 12: Block diagram of the vergence system (reprinted from [29]).

Junji [21] applied correlation technique to estimate disparity between two cameras. This sum-of-absolute-di®erence based vergence module controls only the slave camera in order to minimize the disparity between both cameras. The images are ¯ltered with Laplacian of Gaussian before performing matching in order to reduce e®ect from intensity di®erences of between image frames. The matching operation is basically to minimize sum of absolute di®erence between two image patches from master and slave camera. The disparity map $D(x;y)$ is then calculated by:

$$D(x;y) = \arg\min_{x'} \sum_{¢x¢y2patch} |A - B|$$

where

A = LoG(Targ($x + ¢x; y + ¢y$)

B = LoG(S($x + x_0 + x^0 + ¢x; y + ¢y + y_{offset}$))

LoG = Laplacian of Gaussian

Targ($x; y$) = Target image

S($x; y$) = Search image

x_0 = Value to control slave motor

x^0 = O®set at zero disparity

y_{offset} = O®set in y-axis between both cameras

patch = 5£5 image patch

This disparity map is then converted to a weight map to control the camera.

Correlation technique was also employed in [37]. Their system, however, utilizes the space-variant sensors which has its unique characteristic in hardware point of view, hence, not mentioned here.

Opto-Kinetic Re°ex Camera Head Control

The optokinetic re°ex is a backup system for the vestibulo-oculomotor re°ex. The vestibulo-oculomotor re°ex stablizes the eyes well against the head movements for a duration about 0.5 second. Once the head movement continues for 20-30 seconds longer, the vestibulo-oculomotor re°ex adjusts and can no longer compensate for the head movement. The opto-kinetic re°ex then takes over. This re°ex uses visual information rather than any signal measured from head motion.

There are few researches have incooperated this type of re°ex for the camera head control. The well-known work has been presented in [8]. The OKR was implemented to assist their VOR system for a more stable and robust eyes stabilization system. In

this work, OKR utilizes only a rough approximation of the optical °ow of the entire image at relatively slow speeds. This optical °ow computes the background motion between successive image frames. Using this optic °ow estimate, a displacement motion of the camera head can be compensated.

Panerai et.al. [46] employ visual information with inertial sensor to stabilize the eyes. Even they have not addressed of using any OKR in their system, their approach can be considered as utilizing OKR. Detail of this research has been discussed in the previous section.

Vestibulo-Ocular Re°ex Camera Head Control

The vestibulo-ocular re°ex involves with stabilizing the eye against changes in head position. This re°ex system keeps the eye looking at the same direction as it did before the head movement. In human, this head involuntary compensation utilizes the head velocity and orientation obtained by three semicircular canals and otolith organs to compensate the eyes on the target while head is moving. Current approaches for arti¯cial vestibulo-ocular re°ex can be summarized into three categories as follows:

1. Using only visual information.

   While the head is moving, the eyes stabilize on the target by tracking the target on the image which is currently ¯xated. This can be done using only visual information (i.e. locate the target in the image and maintain the eyes on it).

2. Using copies of motor commands.

   Copies of motor commands provide information about the head motion. With

e±ciently precise kinematics of the camera head, these motor commands can be used to generate a compensatory motion of the eyes.

3. Using inertial sensing devices.

   Inertial sensing devices, e.g. gyroscope and accelerometer, can be implemented to measure the rotational velocity of the head and used to generate for head motion compensation. This information is not relying on any vision and/or internal motor commands.

Though, little e®ort has been done particularly for arti¯cial vestibulo-ocular re°ex camera head movement [47][46] [16][8]. These works utilize both visual and inertial information to stabilize the eyes against the head motion.

Brooks et.al. [8] used three rate gyroscopes mounted on orthogonal axis and two linear accelerometers. These devices are equivalent to semicircular canals and otolith organs in the human VOR system. Using only a velocity signal from these devices, however, yields undesirable implementation and results. They, then, employ a visual feedback to assist the VOR system. This visual feedback module is one of the human eye movements called opto-kinetic re°ex (more detail in next section). Combining VOR with OKN provides a more stable and robust system.

More detail of integrating visual and inertial information for arti¯cial vestibulo-ocular re°ex has been presented in [47][46] [16]. Their current VOR system is limited to the vertical rotational axis. The eye velocity required to stabilize the image of a stationary object has been derived as (see Figure 13):

$$\underline{\omega}_e = \left[\frac{d^2 - d(\frac{b}{2}\sin\omega_h + a\cos w_h)}{\frac{b^2}{4} + a^2 + d^2 - bd\sin w_h - 2ad\cos w_h} \right] \underline{\omega}_h$$

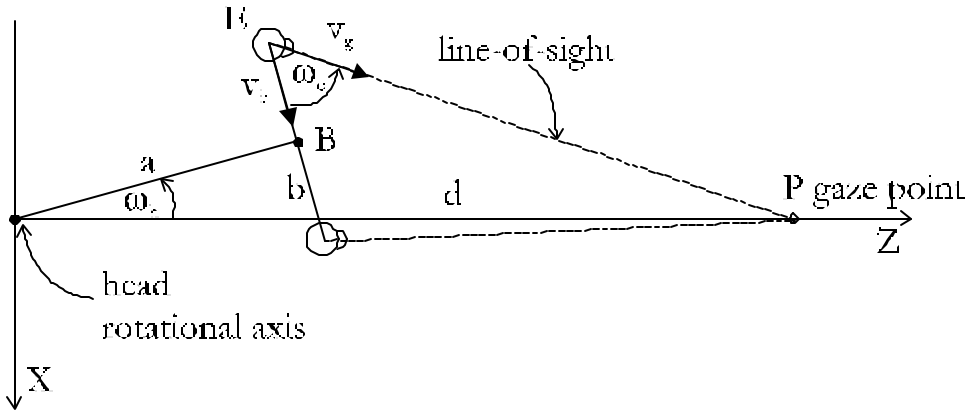


Figure 13: Geometry of the eye-head system showing the parameters relevant to inertial and visual measures (reprinted from [47])

E = eye position

P = gaze point

b = baseline distance

a = distance between the rotational axis and the baseline

where

v_g = normalized vector connecting between E and P

v_b = normalized vector connecting between E and mid-baseline point B

$!_h$ = head angle

$!_e$ = eye angle

The equation shows the relationship between eye velocity $\dot{!}_e$ and geometrical parameters of the eye-head system. This equation also presents the relationship between $\dot{!}_e$ and a distance d of the ¯xation point P for any given head velocity $\dot{!}_h$. From the equation, beside a and b; d is obviously an important factor in computing the eye velocity, hence, the visual information is concerned.

Their structure of a visuo-inertial image stabilization system is shown in Figure 14. The gain of VOR (G_{vor}) is used for tuning the inertial sensor. If the gaze point

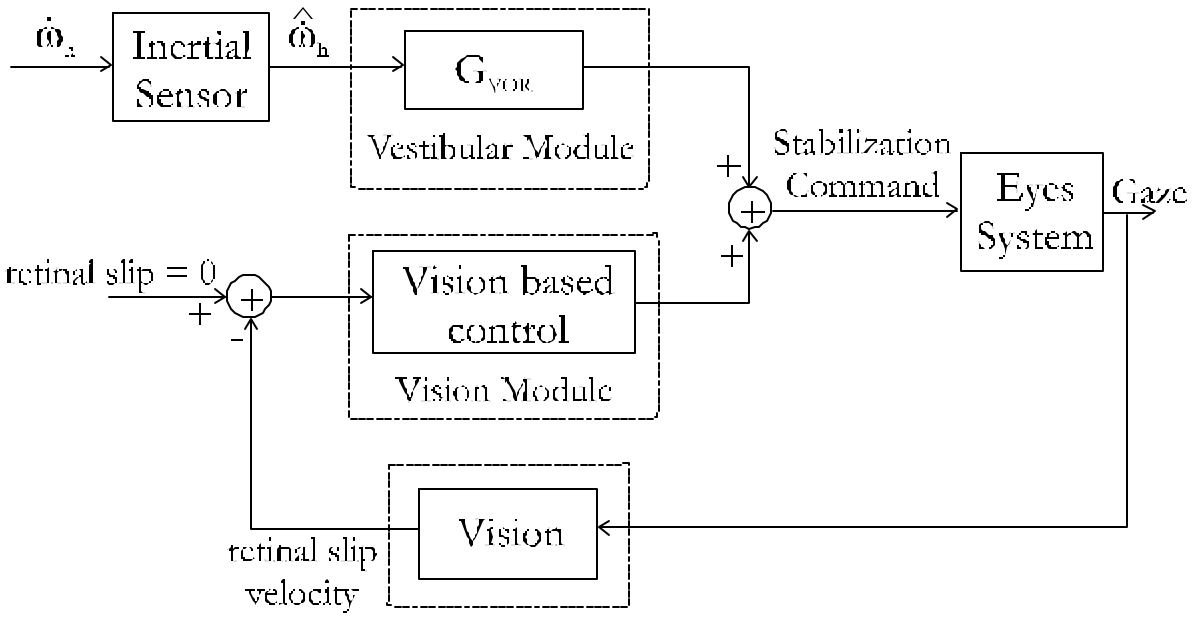


Figure 14: Block diagram for the control system (reprinted from [47])

is at in¯nity, G_{vor} is equal to 1 and head velocity is equal to zero (head motion is not sensitive to the gaze point movement). If the gaze point is closer to the head, G_{vor} is lesser and total gain for stabilizing the image is compensated from visual measures.

Stereo Camera Head System

Stereo camera head or binocular head is essential for active vision task. Most methods and algorithms developed for achieving human-like eyes movement characteristics have to demonstrate on the camera head. This implies that those methods and algorithms must be designed based on their camera head structure. In this section, literature reviews of stereo camera head system in research area of human-like eye movements are discussed. Characteristics and features of each camera head system are brie°y described. All camera head system and its human-like eye controls are summarized in Table 2.

Table 2: Summary of human-like movement stereo camera heads

Name	DOF	Saccade	Smooth Pursuit	Vergence	VOR	OKR
LIRA	4				±	±
VARMA	18	±	±	±		
MARVIN	4	±	±	±		
KTH	13		±	±		
COG	5	±	±	±	±	±
Medusa	4	±	±	±		
ESCHeR	4	±	±	±		
Hadaly	7		±			
Koala	8		±			
AUC	4		±	±		
Rochester	4		±	±		
Yorick	4	±	±			
PennEyes	4		±			

Table 3: Summary of human-like movement stereo camera heads (continued)

Name	Feature Hardware
LIRA	Space-variant sensors Inertial sensors
VARMA	Three-DOF neck Adjustable baseline Motorized zoom lenses
MARVIN	Mounted on mobile robot
KTH	Two-DOF neck Separated eye module for pan, tilt, and cyclotorsion
COG	Inertial sensors High resolution and wide angle lenses for each eye
Medusa	Space-variant sensors
ESCHeR	Space-varient sensors High performance motors
Hadaly	Two-DOF Neck Eyelids with eyelashes
Koala	-
AUC	-
Rochester	-
Yorick	High acceleration motors
PennEyes	Camera head on a robot arm

Table 4: Summary of human-like movement stereo camera heads (continued)

Name	Core Control Methods
LIRA	1st-order approximation of optic °ow Head-eye kinematics model with inertial devices
VARMA	1st-order optical °ow for image velocity Image motion disparity
MARVIN	-
KTH	Correlation-based disparity
COG	Template matching Saccade map Eye stabilization using inertial devices Optical °ow for background image velocity
Medusa	Optical °ow for image velocity Correlation-based disparity from log-polar image
ESCHeR	Gradient-based optical °ow for target velocity Phase-based disparity estimate
Hadaly	Adjustable eyelashes to detect brightness Touch sensing for eyelids
Koala	-
AUC	Template matching Kalman ¯lters
Rochester	Cepstral disparity ¯lter
Yorick	Coarse resolution optical °ow Fine resolution optical °ow
PennEyes	-

LIRA-Lab Head

The LIRA-lab heads [48][49] were built and developed at University of Genova, Italy. Figure 15 shows the LIRA-Lab heads. The oldest version of the head which is still in use is shown in Figure 15 -(a). It has four degrees of freedom, three for the eyes and one for the neck. Its control system is simply implemented on PC-based machine. The tilt and vergence movements are designed to have a human visual system capabilities. Space variant sensing technologies are exploited for autonomous operations and trackings. Figure 15 -(b) displays the smaller version of the binocular head. It has only two degrees of freedom with two color micro-cameras mounted on it. Currently, it has been using in the babybot project [49]. The lastest version of LIRA-Lab head is shown in Figure 15 -(c). It has ¯ve degrees of freedom: left, right, pan and tilt. The eyes are on the common tilt while the neck is decoupled.

The LIRA-Lab heads have been developed to achieve the human-like eye movements. Their main study is focused on visuo-inertial integration which is the relationship betwen visual and inertial information for binocular gaze stabilization Theire approach is based on human physiological data and inertial measures.

VARMA Head

VARMA is the camera head under development at the Institute of Systems and Robotics in Portugal. The ¯rst generataion of VARMA head [50] (see Figure 16) was

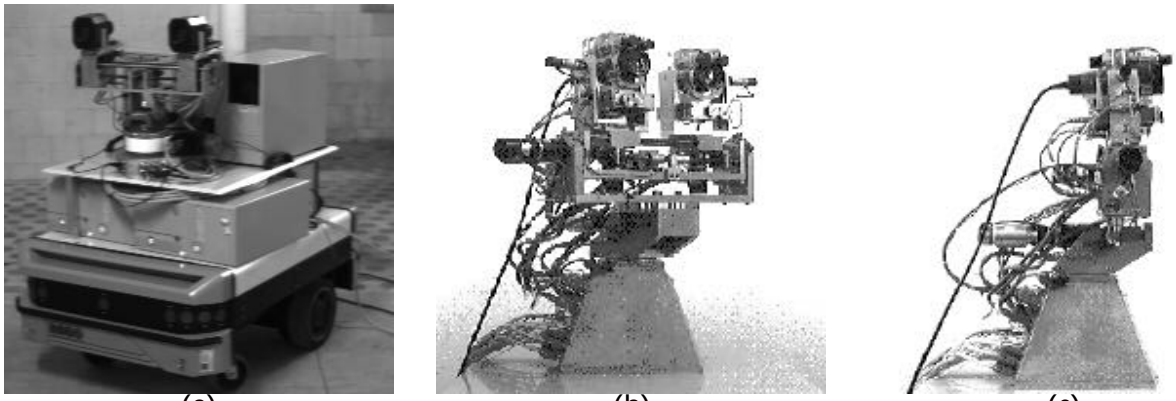
(a) (b) (c)

Figure 15: LIRA lab head: (a) oldest version of the head (b) small version binocular head (c) lastest version of robot head

built using step motors. Each eye has four motorized degrees of freedom (pan for rotation and focus, zoom and aperture for accommodation). Its neck has two degrees of freedom for pan and tilt. There is one degree of freedom for changing the baseline. The second generation of VARMA has 18 degrees of freedom listed as follows:

- Each eye has three mechanical degrees of freedom (tilt, pan and ciclotorsion) for rotation, three optical degrees of freedom (focus, zoom and aperture) for accomodation and optical center adjustment.
- The neck has three degrees of freedom (swing, tilt and pan).
- One degree of freedom for changing the baseline.

Research on VARMA is focused on tracking and smooth pursuit.



(a) (b) (c)

Figure 16: VARMA head of Institute of Systems and Robotics, university of Coimbra, Protugal: (a) ¯rst generation of Head-Eye system - Step Motors Version (b) second generation of Head-Eye system - Harmonic Drive DC Motors Version (c) another side view

MARVIN Robot

MARVIN is a mobile robot equiped with a double binocular camera system developed at Ruhr University Bochum, Germany under the project NAMOS (Navigation of Autonomous MObile Systems) [51]. The camera head system has independently controllable cameras and adjustable basis. Two types of cameras are mounted on each side for foveal and peripheral image processing.

MARVINs vision system utilizes only visual sensors to accomplish desire tasks as follows:

- perform sensor-driven and expectation-driven saccades to explore or recognize a scene.
- perform vergence movements and estimation distance and local shape of 3-dimensional objects.
- track and segment moving objects.
- employ motion-parallax to percieve 3-dimensional scenes.

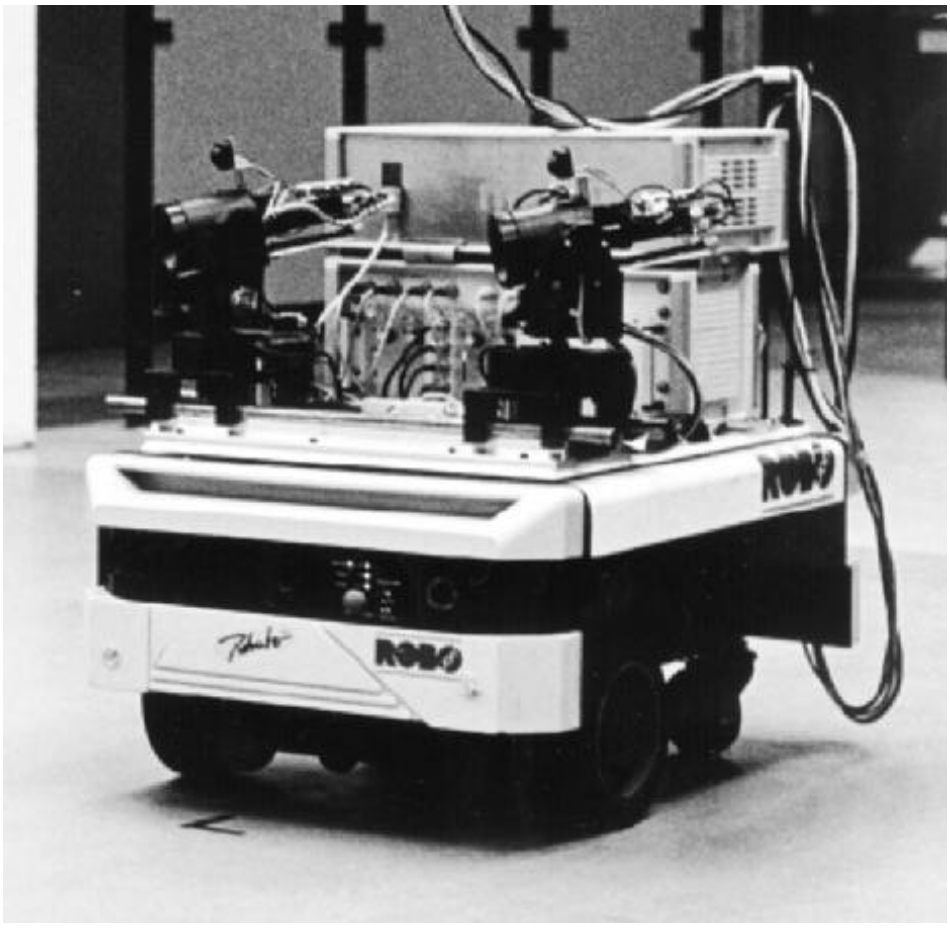


Figure 17: MARVIN mobile robot and its binocular head.

Figure 17 shows MARVIN and its camera head.

KTH Head

KTH head has been developed by the Computational Vision and Active Perception Laboratory (CVAP) , department of Numerical Analysis and Computing Science (NADA) at KTH in Stockholm, Sweden. KTH head has accommodation facility, separate eye modules, foveal simulation capability (e.g. zooming) and neck joints. This allows for synchronous control of eye modules, accommodation, iris control and image mangi¯cation. Its location of the lens axially is also adjustable for optical center

to be displaced to different places. Main research developed on KTH head includes dynamic fixation and active perception.

COG Head

COG [8] is an upper-torso humanoid robot which has twenty-one degrees of freedom and a variety of sensory systems (see Figure 18). Its visual system is designed to have capabilities of the human visual system, including binocularity and space-variant sensing. Each eye can rotate independently on vertical axis (pan). Both eyes are coupled on horizontal axis (tilt). There are two grayscale cameras per eye: one for wide-angle len for periphery field of view and another one for narrow-angle len for foveal area field of view. Saccades, smooth pursuit tracking and binocular vergence have been implemented on Cog's visual system. Cog also has a vestibular system integrated with an extra hardware. Three rate gyroscopes and two linear accelerometers are mounted on the head, providing a measurement of head velocity. This is then used to counter-rotate the eyes and maintain the direction of the gaze while the head is moving. OKN is also operated on Cog using a rough approximation of the optic flow on the background image to compensate for the head movement. More detail on building COG can be found in [?].

Medusa Stereo Head

Medusa stereo head [52][53] is designed for research on active vision applications by Computer Vision Lab at the Institute of Systems and Robotics, Instituto Superior Tecnico, Lisbon, Portugal (see Figure 19). It has four degrees of freedom: tow independent vergences, common tilt and common pan. Its baseline is manually adjustable

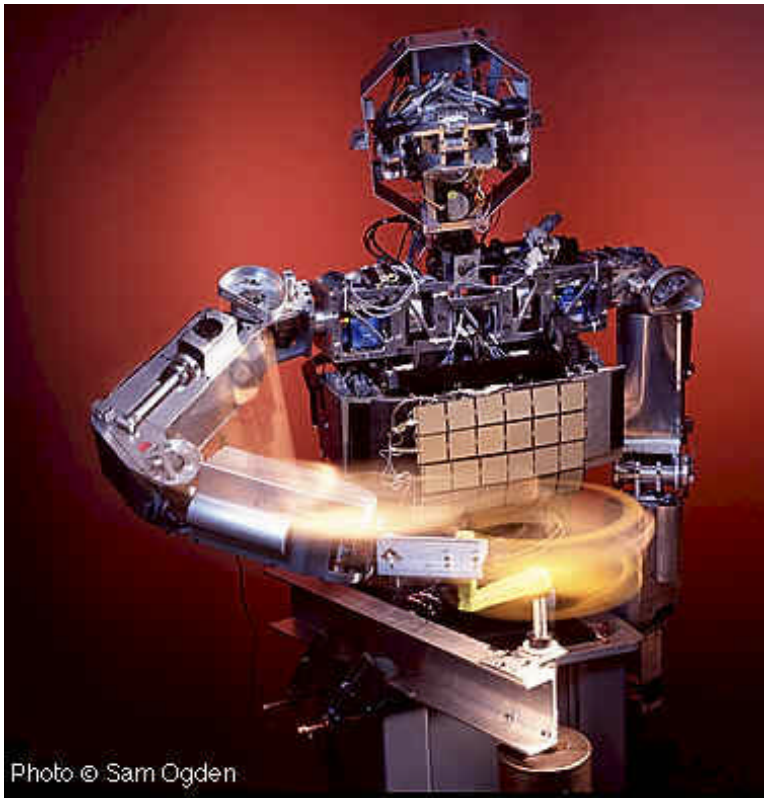


Figure 18: COG - humanoid robot

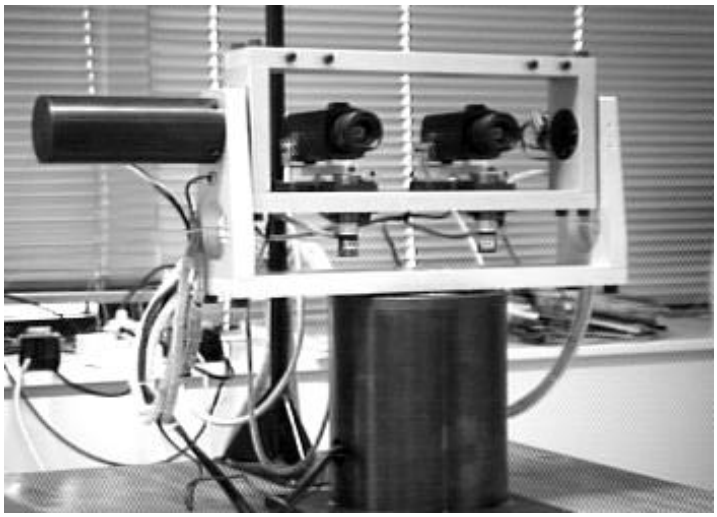


Figure 19: Medusa stereo head

for applications with di®erent ranges of depth. Main ocular movements similar to the human oculomotor system have been implemented. These include saccades, smooth pursuit, and vergence.

ESCHeR Head

ESCHeR [22][54][55][56] [57] (Etl Stereo Compact Head For Robot Vision) is a high performance stereo-head developed by the Humanoid Interaction Laboratory, Electrotechnical Laboratory, Japan. It is equipped with foveated wide-angle lenses mounted on four degrees of freedom camera head: independent left and right vergence with a common tilt supported by a common pan (see Figure 20). Velocity and disparity cues are utilized to provide robust tracking. The velocity of the target is determined by a fast gradient-based optical °ow segmentation algorithm. The position of the target and its vergence correction on left/right camera are computed using zero-disparity ¯lter. These capabilities allow ESCHeR to drive saccade and pursuit

Figure 20: ESCHeR head

the target within a complex background. The Kalman ¯lter is also integrated using information from both optical °ow and zero-disparity ¯lter to produce a smoother tracking. This system has performed tracking in real-time and proven capable of tracking at frame rate on human body parts such as hands and heads moving with speed up to 80 degree/sec over cluttered backgrounds.

Hadaly Head

WE-3R [58] is an anthropomorphic head-eye system for a humanoid robot \Hadaly-2" (see Figure 21). WE-3R consists of three parts: eyeball, neck, and eyelid. It is capable of tracking objects in 3 dimensional space. WE-3R is also designed to adjust to brightness by having eyelids controlled by hardware and retina (adaptation) and iris (adjustment of the pupil diameter) controlled by software. The tracking system, however, can only perform on the light bulb as the targeted object. VOR is implemented for head motion but no information has been provided.

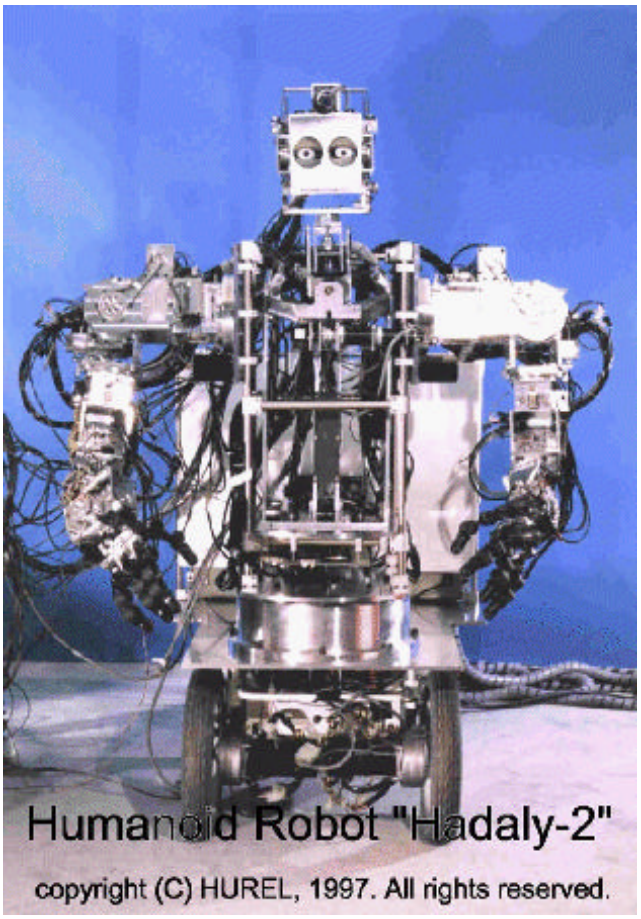
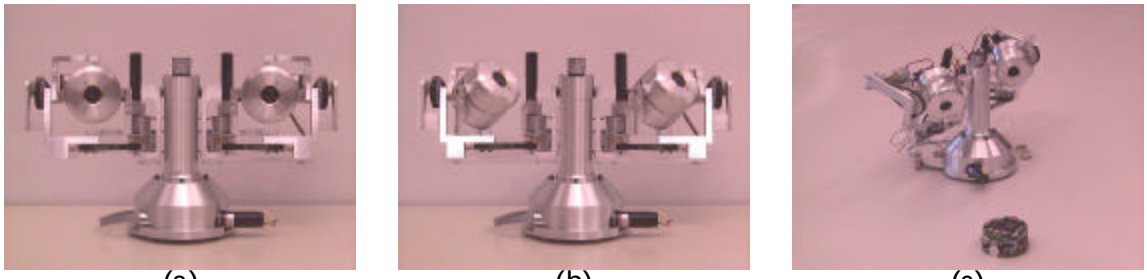


Figure 21: Hadaly-2 robot



(a) (b) (c)

Figure 22: Koala head

Koala Head

Koala head is a binocular designed by Sebastien Menot [59] at Microprocessor and Interface Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. It has eight degrees of freedom which each eye has three DOF (pan, tilt, and cyclotorsion). Left and right eyes mounted separately on 2 DOF neck (pan and tile). The human ocular movements have been studied with this camera head including saccades, smooth pursuit, and vestibular ocular reflex.

AUC Robot Camera Head

AUC head has been developed at Laboratory of Image Analysis, department of Medical Informatics and Image Analysis, Institute of Electronic Systems, Aalborg University, Denmark [60]. It has 12 degrees of freedom including two controlable zoom lenses for control of zoom, aperture, and focus. Each camera has independent pan (left and right). Neck movements are emulated by combination of pan and tilt. Detail in design and implementation of the AUC head can be obtained from [61].

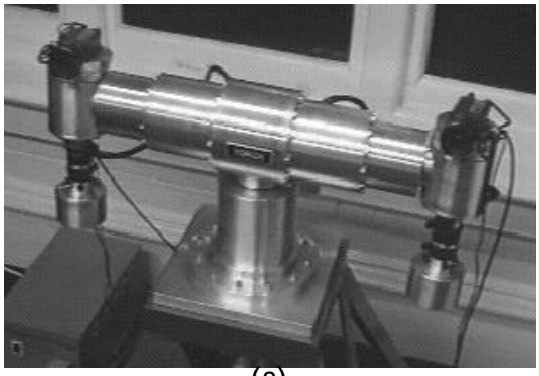
Figure 23: AUC robot camera head

Yorick Stereo Camera Platforms

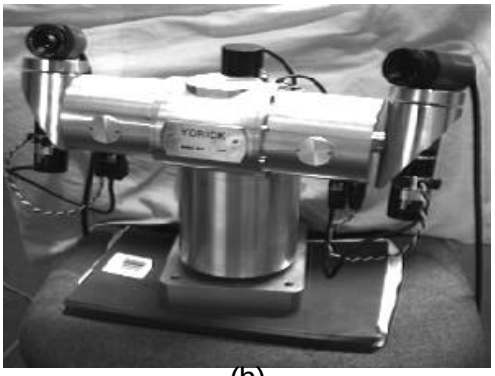
Yorick is a name of a series of high performance binocular head/eye platforms built by Active Vision Laboratory, University of Oxford, England [62]. Figure 24 shows series of Yorick head: 11-14, 8-11, and 5-5c. They are di®erent in size and speed of each motor. Yorick head has basic pan, tilt, and verge movements. Implementation of human ocular movements have been done on Yorick head including saccades and smooth pursuit [63].

PennEyes

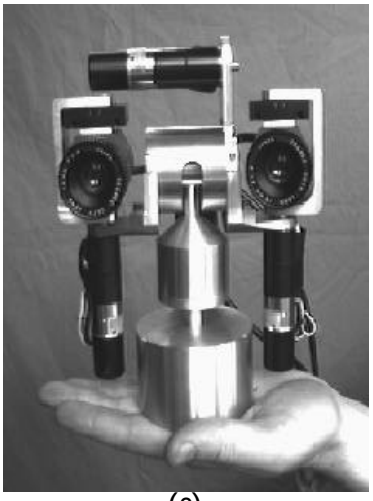
PennEyes [64] (GRASP Laboratory at the Department of Computer and Information Science, University of Pennsylvania [65]) is a binocular active visition system that was designed to be a positionable vision system (see Figure 25-(a)). This work mostly describes the hardware architecture to build the system from commercially



(a)



(b)



(c)

Figure 24: Yorick head: (a) Yorick 11-14 (b) Yorick 8-11 (c) Yorick 5-5c

(a) (b)

Figure 25: PennEyes: (a) camera head (b) camera head mounted on puma robot arm

available components. It is composed of two CCD cameras with motorized lenses (for zoom, focus, and aperture) mounted on 4-axis camera head. The camera head is mounted on the 6 degree of freedom Puma robotica arm. This system was used to actively explore an arbitrary scene with a responsive binocular platform and obtain better representation of objects of interest. Possibility of performance is discussed in hardware point of view. Current system on PennEyes is mounted on a puma robot arm. The study of this system is focused on tracking in full 3-dimensional.

Summary

This chapter reviews all related works in area of human-like eyes movements for binocular head controls. A brief biological overview of eyes movement in human has been presented. Binocular head control methods with human-like eye movements have been discussed. Mapping between motors and image coordinates have been successfully used for saccades. Inverse kinematics approach was also presented but obtaining an accurate kinematics is not trivial. Extracting target information such as position and velocity for smoot pursuit systems have been discussed in using both

optical °ow and binocular disparity techniques. Vergence control work has been focused on disparity estimate. Matching- and phase-based disparity estimation have been reviewed. Works on arti¯cial vestibulo-ocular and opto-kinetic re°exes have been discussed using both inertial sensing device and visual measures. Brief overviews on stereo camera head systems that have been used in researches around the world are summarized in ¯nal section. Theoretical analysis of the proposed system in this study is examined in the following chapter.

CHAPTER III

THEORETICAL ANALYSIS

Introduction

This chapter describes the main components of the proposed system from a theoretical point of view. Two of three main parts are discussed: the visual attention network and the eyes motion center. The main diagram of the system can be seen in Figure 26. The visual attention network tells the eyes motion center where to look. The eyes motion center then utilizes both the retinal motion signal and motor information to generate motor commands for the camera head to track the target. The camera head controller directly controls the camera head in the hardware layer. Details of these components are described in the following sections.

Visual Attention/Searching Network

The visual attention network analyzes scenes from a sequence of images and provides useful information to the rest of the system. Images are acquired from two color CCD cameras mounted on a robot camera head. The network performs a visual search at two levels: global motion detection and object-level/disparity detection (see Figure 27). The global motion detection narrows down the scene to moving targets only while the object-level/disparity detection examines these moving targets in ¯ner detail.

The purpose of the visual attention network agent is mainly to provide input to the eyes motion center agent for where to look next. Multiple types of output from

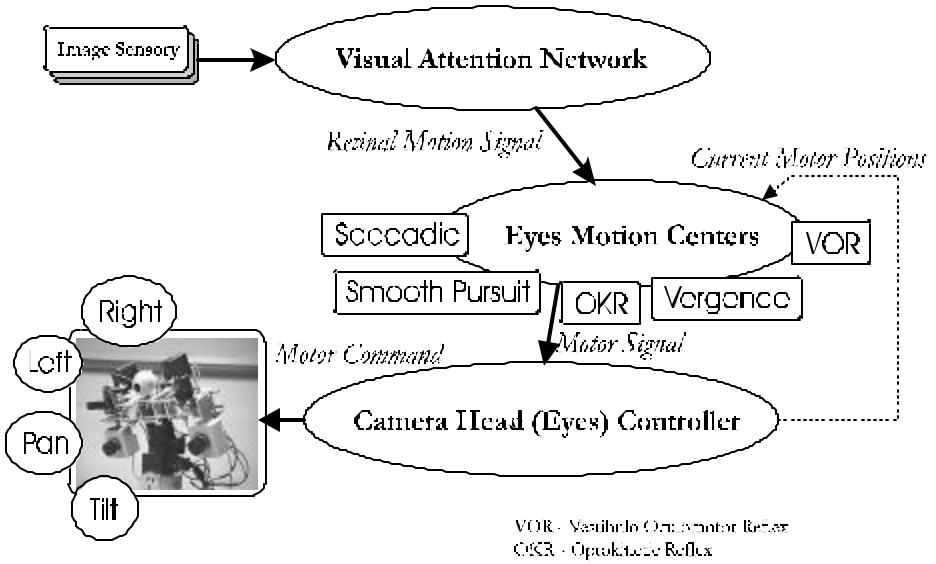


Figure 26: Main system diagram

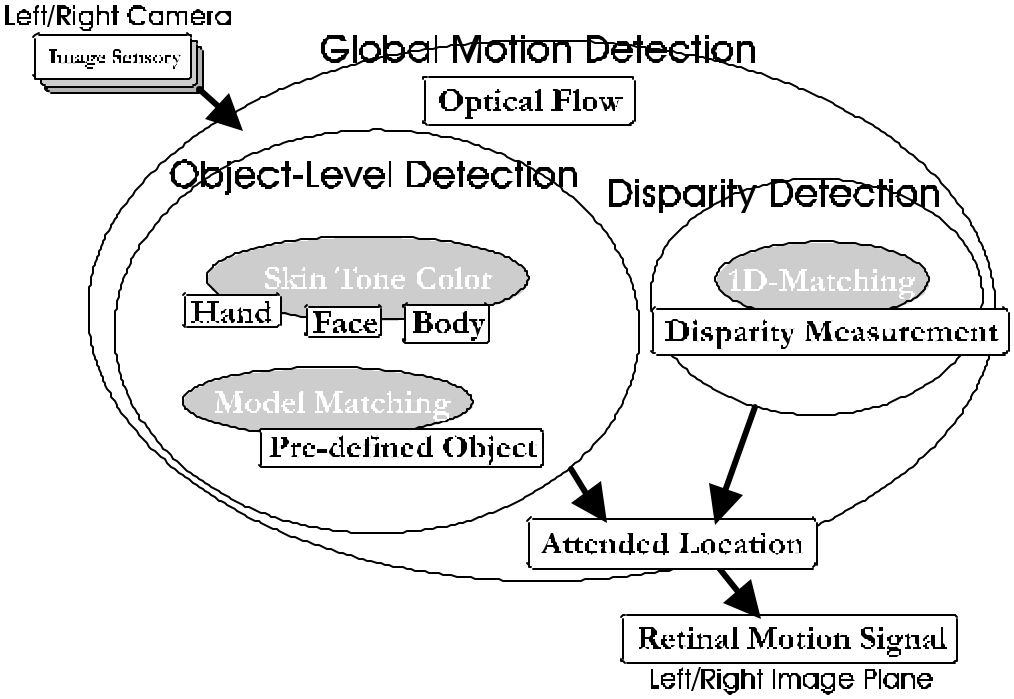


Figure 27: Visual attention network

this agent are utilized by di®erent components in the eyes motion center. Moreover, the purpose of the visual attention network can be solved by any method including feature gate [66].

Global Motion Detection

Global motion detection uses the information from optical °ow to detect a motion ¯eld within the whole image. It con¯nes the attended location to where the motion occurs. The computation of optical °ow is typically computationally expensive. Skin color-based correlation optical °ow estimation, hence, is proposed in order to achieve the performance of real-time optical °ow. The feature matching in °ow estimation is narrowed down by skin color segmentation. This considerably reduces size of search space for matching processes. The target motion also provides meaningful clues because only moving skin tone blobs are attended. Consequently, only human motion is being considered. After the motion of skin tone blobs is estimated, the camera head ego motion has to be compensated in order to obtain the absolute velocity of the target. Image-based camera stabilization is utilized to estimate the camera velocity between the image sequences. There are two main components obtaining the target velocity. Firstly, the relative velocity of the target from consecutive images needs to be extracted. Secondly, the global velocity of the entire image is calculated for the camera head motion compensation. Figure 28 presents the main diagram for the velocity estimation scheme. The following sections describe the details of the proposed system.

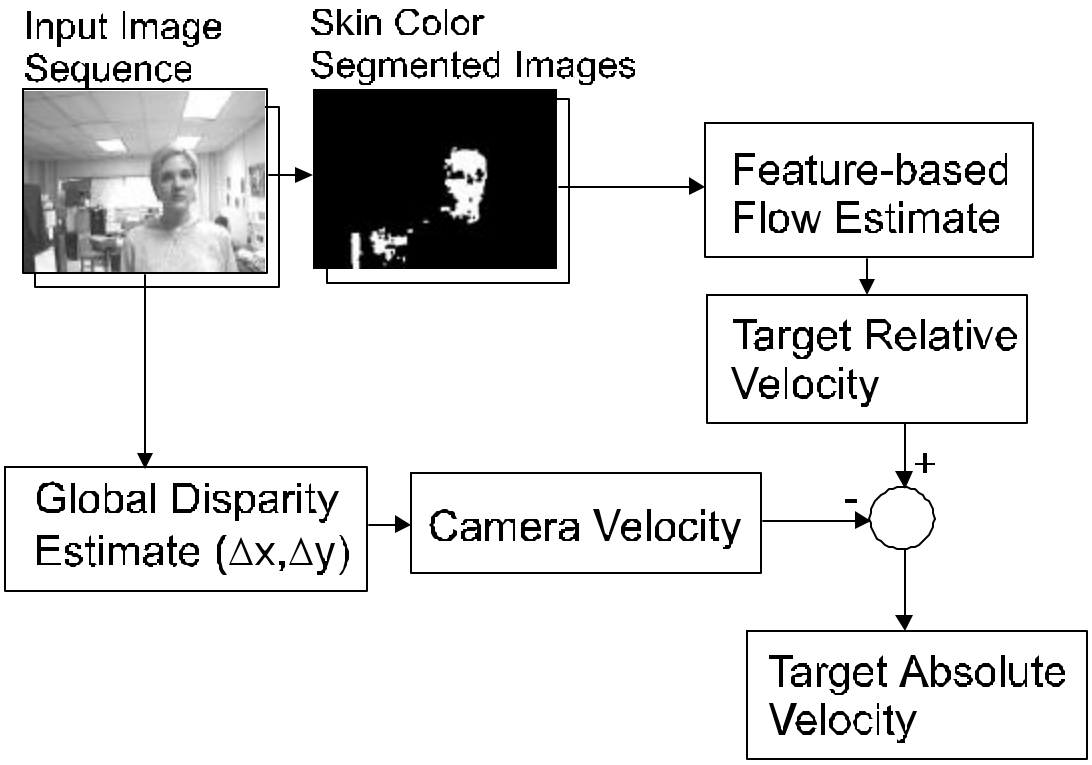


Figure 28: Target velocity estimation

Skin Color Segmentation

The output image of the skin color segmentation process contains human body candidates. This eliminates irrelevant pixels in the image that do not belong to the human body. The resulting images, then, give a smaller search space for further processes e.g. °ow estimation. Using skin color information allows for the detection of human motion. Because more than one skin-tone blob may occur in the image, a post-image processing routine is necessary to ensure that the skin-tone blob is the right target.

Feature-based Flow Estimation

The well-known block-matching technique [7] is utilized in this system as is seen in the following processes.

Let input images at time t_n and t_{n+1} be I_n ,and I_{n+1}, and segmented images be S_n, and S_{n+1}, respectively. The feature tracking process applies to all pixels of input image patches, p_i, in I_{n+1} which belong to skin-tone blob, C, to estimate motion (displacement) of selected features. In this case, the motion ¯eld estimation is to ¯nd the coordinates $(x^0; y^0)$ of the image patch center of p_i that minimize the correspondence between the image patches belonging to the image I_{n+1} and the reference image patch in I_n with coordinates $x; y$ (see Figure 29).

The maximum velocity in x and y direction are de¯ned as $V_{x\,i\,max}$ and $V_{y\,i\,max}$, respectively. The size of maximum velocity is obviously half of the search window size for each direction (in pixel). The velocity in x and y direction of each image patch (in pixel) are V_x and V_y and equal to x- and y-displacement, respectively.

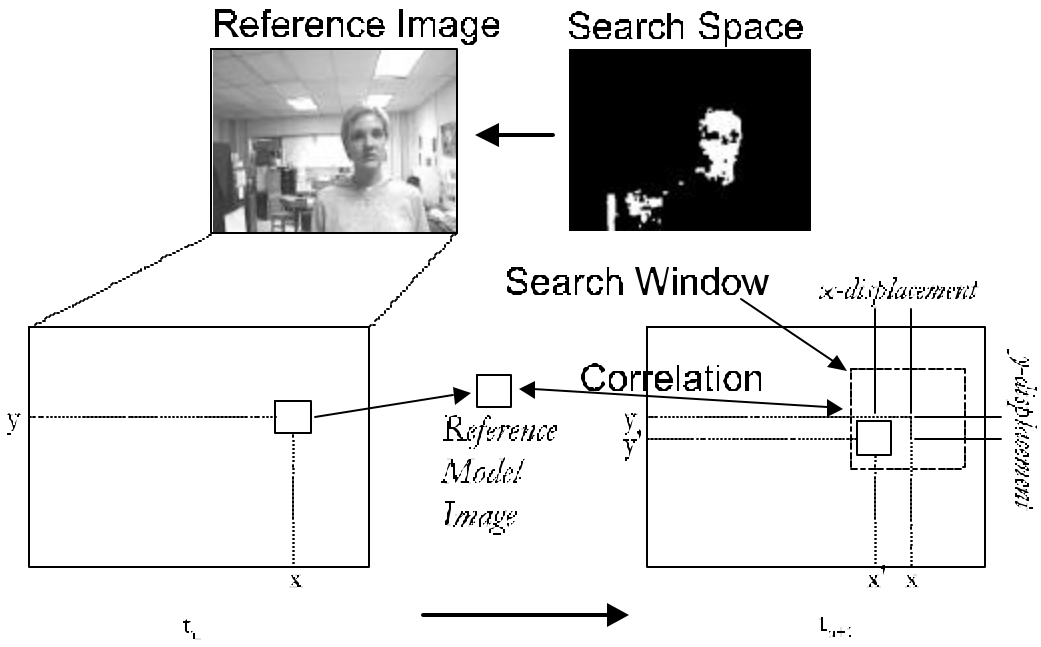


Figure 29: Block-matching technique for °ow estimation

Camera Head Motion Compensation

Disparity between consecutive images provides useful information in obtaining image slip (velocity of the entire successive images). Figure 30 depicts the diagram of computing image velocity for camera head motion compensation. The concept of camera head velocity estimation is the same as the estimate velocity of the sequence images. Disparity estimation is applied to successive images instead of left and right images. The resulting disparity in x- and y-direction are then equivalent to displacement of the entire image. Correlation-based disparity estimation is modi¯ed to obtain the estimated global disparity rather than local disparity. The x- and y-velocity is constant through out the entire image.

Once velocity of the camera head is known, the absolute target velocity can now be determined by subtracting the °ow estimation from camera displacement. The

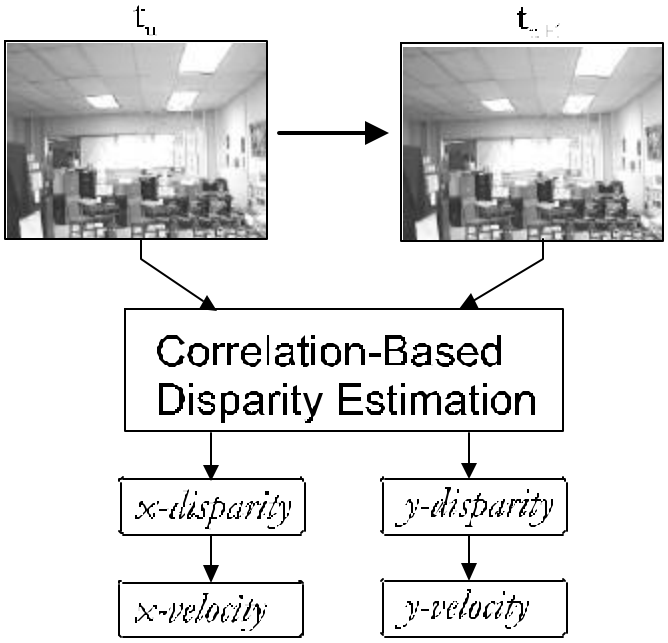


Figure 30: Image velocity estimation between consecutive images

remaining velocity °ow is utilized to extract the target motion.

Object-Level Detection

After the motion clue has been extracted, the object-level detection then can focus on this particular region of interest (where the motion occurs) in detail. The skin tone color segmenter and the model matching can be utilized to con¯rm target existence. The remaining pixels, after the skin tone color segmenter is performed, can result in a human face, hand or body image. The classical model matching can also be applied to locate the object of interest. Pre-de¯ned object models, such as certain orientations and scales of the human hand, are built in a look-up table for fast access and correlation operation. Each module produces a saliency map to represent the absence of a moving target in the scene.

Disparity Detection

Disparity clue provides more useful information in this visual system design. Not only does it ensure that the same object is ¯xated on both eyes, but it also provides some clues for depth estimation. To keep real-time performance, the disparity detection based on 1-dimensional matching is then proposed. Assuming that both cameras are mounted on the same tilt axis, the problem of ¯nding a matching point then becomes 1-dimensional matching. This can be done by a simple matching/correlation of projected left/right images along x-axis within a pre-de¯ned range in y-axis.

The object-level/disparity detection provides the attended location of the target in the image plane for both cameras. It is represented by a retinal motion vector and a disparity vector. The retinal motion vector's amplitude is the Euclidean distance between the target and the center of the image while its direction is taken from the center of the image toward the target in the image plane. The disparity vector is a 1-dimensional vector along the x-axis in the image plane (left/right). Its amplitude is computed from the center of the image plane to the location of the target detected by the disparity detection module. Both vectors, called retinal motion signal, are sent to other parts, such as the eyes motion centers and the camera head controller, in order to be interpreted and executed for the eye movements of the humanoid camera head. The disparity vector, produced by a zero disparity ¯lter component is particularly utilized by the vergence component in the eyes motion center agent.

Disparity Estimation

A stereo pair of images from left and right cameras is acquired. These images give two slightly di®erent viewpoints. There are two di®erent disparity measurements

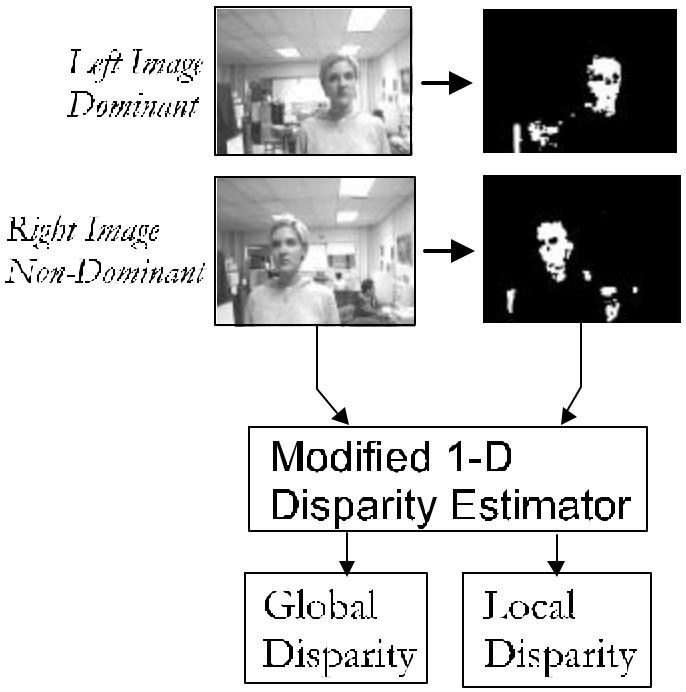


Figure 31: Disparity detection scheme for both global and local disparity

needed for this system. First is a global disparity for camera head compensation. Second is an object disparity for camera vergence control. Both measurements utilize a modi¯ed 1-D correlation-based disparity estimate algorithm to detect disparity between these two images. The algorithm can be summarized as follows (see Figure 31):

Let L(x; y) be an image taken from a dominant (left) camera and R(x; y) be an image taken from a non-dominant (right) camera. In order to extract the vergence disparity between two images from both cameras, the object of interest must be segmented from the background. This can be done by applying a skin mask image to the original image using AND operation. Consequently, the ¯nal images before

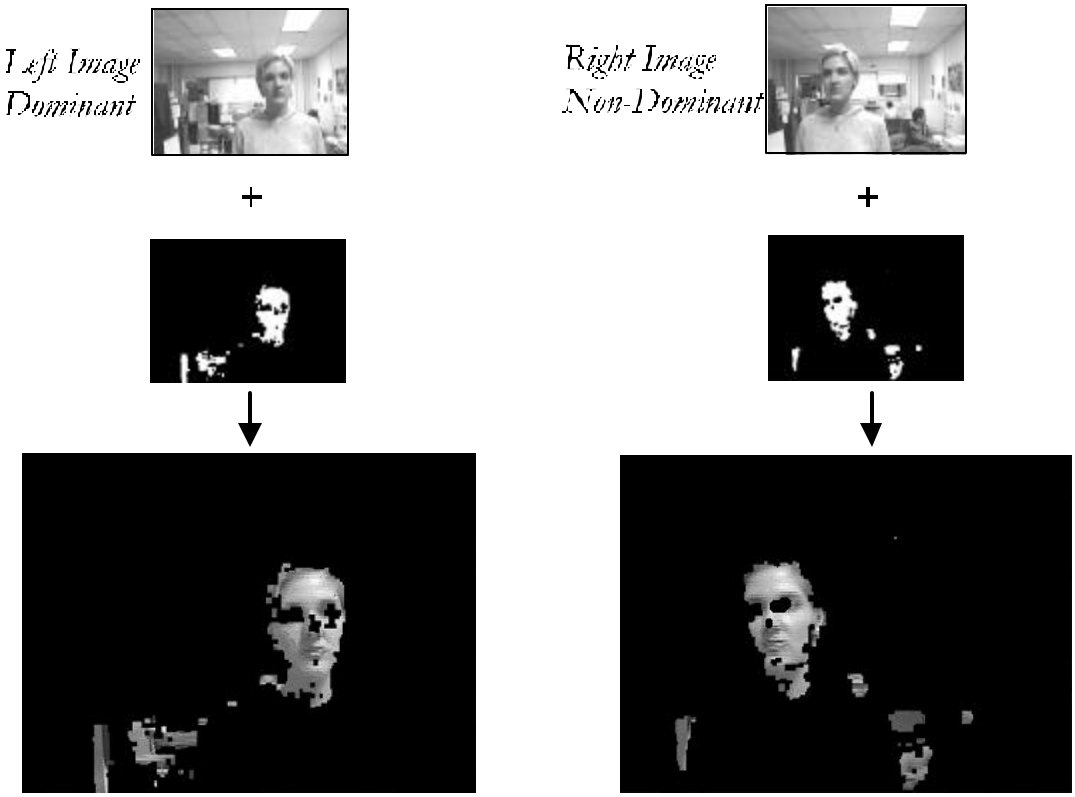


Figure 32: AND operation between input image and skin segmented image

extracting disparity are de¯ned as (see Figure 32):

$$L^{0}(x;y) = L(x;y) - M_{L}(x;y) \tag{1}$$

$$R^{0}(x;y) = R(x;y) - M_{R}(x;y) \tag{2}$$

where $M_{L}(x;y)$ and $M_{R}(x;y)$ are the skin segmented images from corresponding left and right cameras.

One way to achieve faster processing and to separate disparities along each dimension is to compute disparities from projection of the image along the x- and y-axes. The projection of $L^{0}(x;y)$ along y-direction onto the x-axis is de¯ned by:

$$L^{0}_{y}(x) = \int L^{0}(x;y)dy: \tag{3}$$

The projection of $L^0(x;y)$ along x-direction is de¯ned by:

$$L^0_x(y) = \int L^0(x;y)dx: \tag{4}$$

Similarly, the projection of $R^0(x;y)$ along y- and x-direction are de¯ned as:

$$R^0_y(x) = \int R^0(x;y)dy \tag{5}$$

and

$$R^0_x(y) = \int R^0(x;y)dx: \tag{6}$$

Disparity along each dimension can be estimated from a pair of these signal i.e. $L^0_y(x)$, $R^0_y(x)$ and $L^0_x(y)$, $R^0_x(y)$: By performing a correlation on the dominant signal, L^0, with the non-dominant signal, R^0, the output signal will yield a peak of a matching point between L^0 and R^0: The distance between the location of the peak from the center of the signal is disparity in that direction. Let x_c and y_c be the center of $L^0_y(x)$ and $L^0_x(y)$, respectively. The disparity can then be obtained by the following processes:

$$C_{y\,i\,max} = \max Corr(L^0_y(x); R^0_y(x))j_{x=x_i} \tag{7}$$

$$C_{x\,i\,max} = \max Corr(L^0_x(y); R^0_x(y))j_{y=y_i} \tag{8}$$

$$d_x = x_i \ i \ x_c \tag{9}$$

$$d_y = y_i \ i \ y_c \tag{10}$$

where d_x and d_y are disparity in x- and y-direction.

Note that the unit of d_x and d_y is in pixel. This disparity tells how much the non-dominant image shifted from the dominant image in the corresponding direction. In this system, both left and right cameras have the common y-axis. Therefore, only d_x is needed. This d_x composes the disparity vector used for vergence control.

For global disparity, the objective is to estimate shifted pixels between consecutive images. Consequently, the dominant image will be the image from the left camera at time t and the non-dominant image will be the image from left camera at time t + 1, similarly for the right camera. Another di®erence in estimate global disparity is all images are not segmented version to preserve the representation of the entire image content. Both d_x and d_y are estimated and used as a velocity of the camera in x- and y-direction.

Eyes Motion Center

Designs of the eyes motion center can mainly be described in four categories: saccade, smooth pursuit, vergence, and vestibulo-ocular re°ex (VOR). Each eye movement mode is designed separately. Only some resources are shared among them such as gray-scale images and segmented images. The main concept of designing such system is to keep real-time performance using available simple and robust computer vision algorithms. The following table shows description of each eye movement behavior to be achieved.

Saccade

Saccade is one of the most important eye movements because human eyes always perform saccade movement [4]. The target is ¯rst spotted in the visual sensory

Table 5: Summary of eye movement mode for designing the system

Mode	Description	Speed	Input	Motors
Saccades	Gaze shift to the target at one motor command	Fast	Target position	Verge Tilt
Smooth Pursuit	Maintain target on the center of the image	Moderate	Target position Target velocity	Verge Tilt
Vergence	Keep both cameras on the same target	Moderate	Disparity	Verge
VOR	Stabilize the eyes while moving camera head	Moderate	Head positions Head speeds	Verge Pan
OKR	Stabilize the eyes while moving camera head	Slow	Displacement between successive images	Verge Pan

system and the saccade is then initiated. A saccade command is both a direction and amplitude for quickly bringing the target onto the fovea by using the information about a target location and an initial position of the eyes. Saccade is a basic behavior mostly implemented on a binocular system. The main task for saccade is to be used in moving the gaze of the camera head to the target appearing in the visual ¯eld and out of the fovea area. A general concept of implementation for saccades is ¯rst described. Detail of the design is then examined.

Design Concept for Saccade Implementation

Saccade is a mechanism that maps between a target position in an image plane to a motor command in order to move the camera to the target at once. Figure 33 shows a diagram of the general saccade mechanism. After the target position is acquired, $(\Delta x; \Delta y)$ is determined to use as an input to a saccade mapping function. This function maps image coordinate to a motor command such as tilt and verge motors. The new motor command moves cameras' direction so that the target is brought onto the center of the image. Unlike other kinds of eye movement, the camera movement is performed at once to bring the target onto the fovea. A very precise mapping

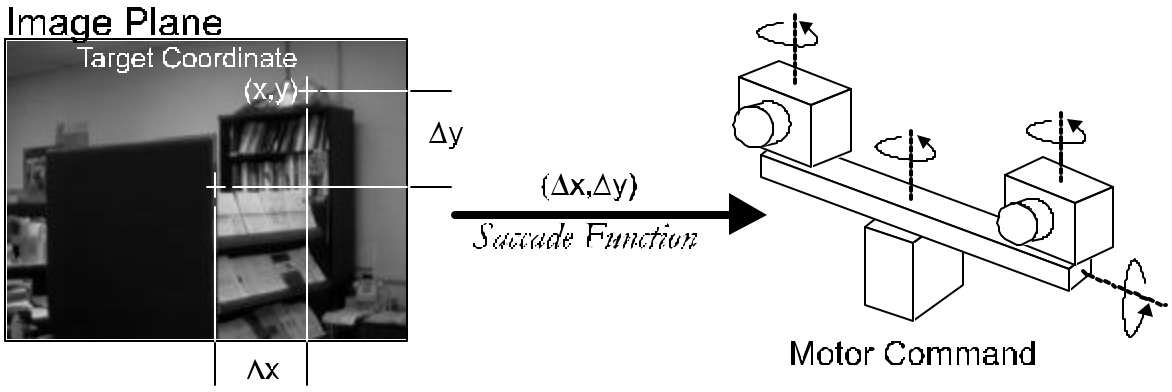


Figure 33: Saccade mechanism

between image plane and motor commands is therefore essential.

Neural Network Approach for Head-Eye Coordination Mapping

Mapping between image coordinate and camera head position is considered a non-linear problem. Using a neural network approach can overcome this non-linear problem without any camera head calibration. Basic back-propagation neural network is utilized in this system to perform a mapping. The neural-network-based saccade system is depicted in Figure 33. The neural module receives the position of the target and simulates corresponding saccade motor commands for eyes motion center to update the position of the camera head via camera head controller. Additional post-saccade target locating is designed to correct the neural weights in case there is any error in saccade process (i.e. target is not on the center of the image after saccade). This post-saccade target locating generates error vectors to feed back to the neural module for updating the weights. This process runs at o®-line (not in tracking mode). Normally, if camera head structure is not changed, the saccade module should perform e±ciently after its ¯rst training.

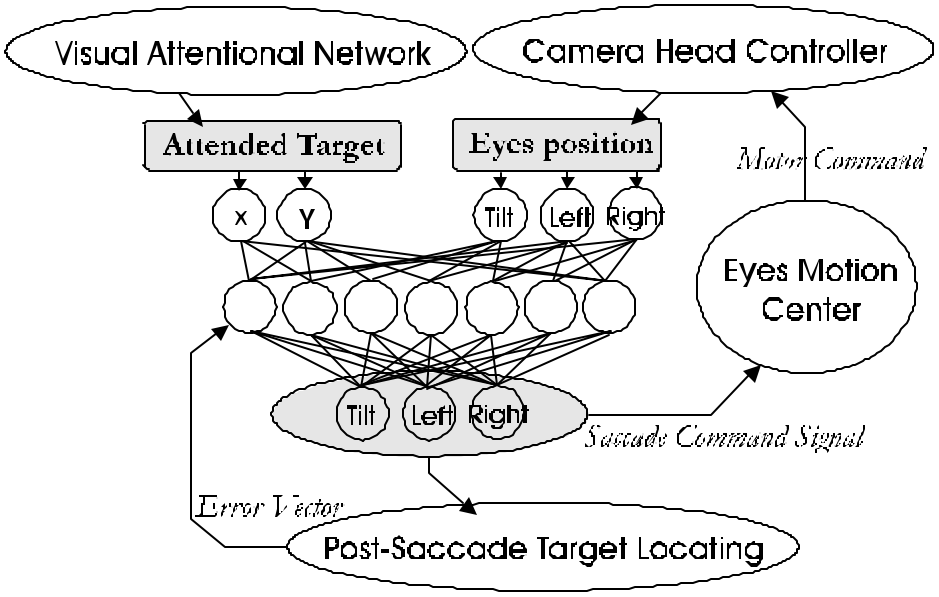


Figure 34: Neural network-based saccade system

Figure 36 shows a structure of feed-forward neural network for saccade training. There are two input nodes which take the distance of the target from the center of the image plane, ¢x and ¢y, as an input. Middle layer contains N hidden neurodes. There are two output nodes providing the output motor commands, tilt and verge motor (either left or right motor). The activation functions for the middle and output layers are h(¢) and g(¢), respectively. Both activation functions are bipolar hyperbolic tangent sigmoid (see Figure 35), because inputs (¢x and ¢y) and outputs (verge and tilt motors) can have both positive and negative value. Consequently, this extended sigmoid is appropriate for such a signal. The output of neurons in the middle layer can be calculated as:

$$y_n = h\left(\sum_{i=1}^{\infty} w_{in} x_i\right): \qquad (11)$$

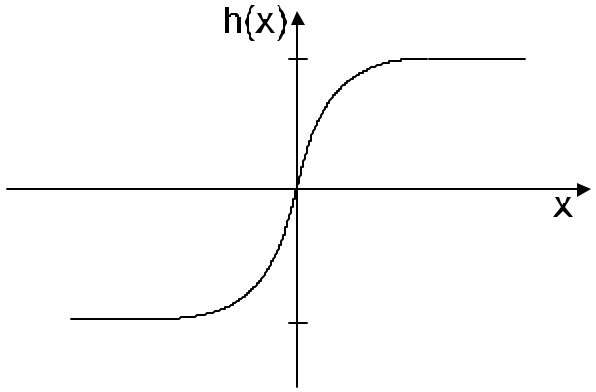


Figure 35: Hyperbolic tangent sigmoid function

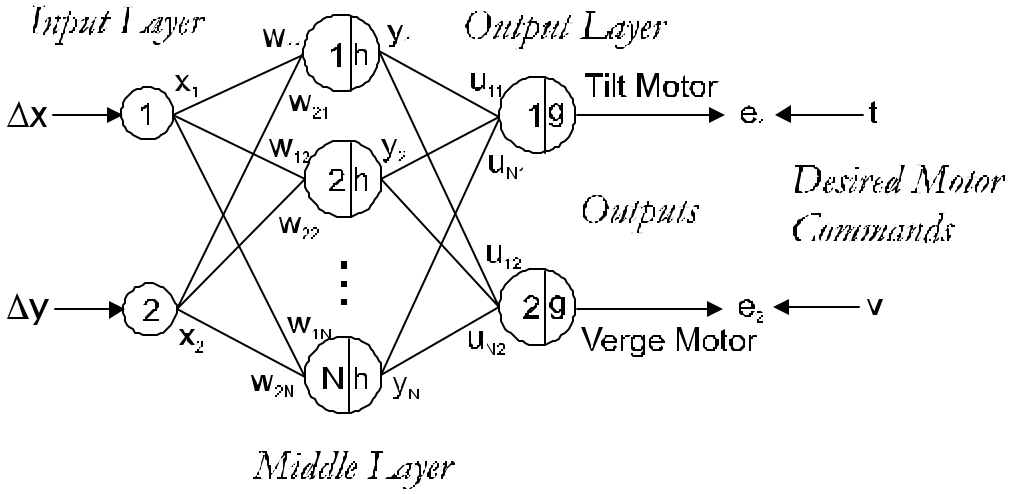


Figure 36: Feedforward neural network for saccade training

Similarly, the output of neurons in outputs layer can be computed as:

$$Tilt\ Motor\ Command = g\left(\sum_{n=1}^{N} u_{n1} y_n\right) \tag{12}$$

$$Verge\ Motor\ Command = g\left(\sum_{n=1}^{N} u_{n2} y_n\right) \tag{13}$$

The designed network only has one hidden layer which is sufficient for a small number of inputs and outputs. This reduces training and simulating time. The back

propagation learning rule is applied for training due to its simpli¯cation. Input/target pairs are collected by recording the camera motor positions corresponding to the position of the target that is appearing in the center of the image.

Post-Saccade Processing

Post-saccade processing is designed to observe the accuracy of the saccades. The procedure can be summarized as follows:

- Generate target to perform saccades.
- After the camera head saccades to the target, acquire the target position.
- Measure the distance of the target from the center of the image.
- If the measurement is greater than a pre-de¯ned threshold, compute corresponding error vector to adjust the weights of the network.
- Repeat the process again.

Typically, this process is performed o®-line, i.e. not tracking any object. Normally, after the camera head has been trained, saccades should be accurate as long as the structure of the camera head is not changed.

Smooth Pursuit

Smooth pursuit is a slow process that keeps the target on the fovea. Designing a smooth pursuit system makes use of target position and velocity. A simple proportional control utilizes the target position to track the moving target in such a small

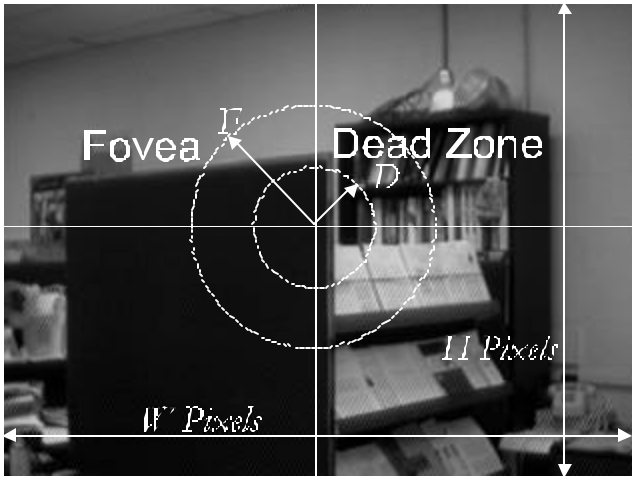


Figure 37: De¯nition of fovea and dead zone area in image plane

area of fovea. Target velocity can be used to predict its position ahead of time in order to smoothly move the camera head to maintain the moving target on the fovea.

Tracking Moving Target

Smooth pursuit begins after the target is intercepted by saccade process. De¯ne an area of the fovea in the image plane as a circle area with radius F pixels and a dead zone area as a smaller circle area with a radius of D pixels (see Figure 37). If the target is detected inside the dead zone, no camera movement is needed. Once the target is captured outside the dead zone and inside the fovea, a smooth pursuit module is activated.

Proportional Control for Tracking

The smooth pursuit uses target position and velocity information to estimate a proportional amount of necessary camera movement. There are two types of input vectors for the smooth pursuit: positional vector and velocity vector (see Figure 38).

The positional vector is composed of the distance of the target from the center of the fovea in x- and y-direction. Let $\vec{p}$ be the positional vector. This vector can be represented by:

$$\vec{p} = (\Delta x; \Delta y) \tag{14}$$

$$p_x = \Delta x \tag{15}$$

$$p_y = \Delta y \tag{16}$$

$$|\vec{p}| = \sqrt{\Delta x^2 + \Delta y^2} \tag{17}$$

Similarly, the velocity vector, $\vec{v}$, can be represented as follows:

$$\vec{v} = (v_x; v_y) \tag{18}$$

$$|\vec{v}| = \sqrt{v_x^2 + v_y^2} \tag{19}$$

where a unit of this velocity vector is in pixels/time unit.

Let m_l, m_r,and m_t be updated motor commands and K_l, K_r, and K_t be constant gains for left, right, and tilt motors, respectively. The smooth pursuit is activated when $D < |\vec{p}| < F$. Corresponding updated motor commands can then be calculated as follows:

$$m_l = K_l \times \Delta x_l \tag{20}$$

$$m_r = K_r \times \Delta x_r \tag{21}$$

$$m_t = K_t \times \frac{(\Delta y_l + \Delta y_r)}{2} \tag{22}$$

where subscript l and r refer to left and right image, respectively.

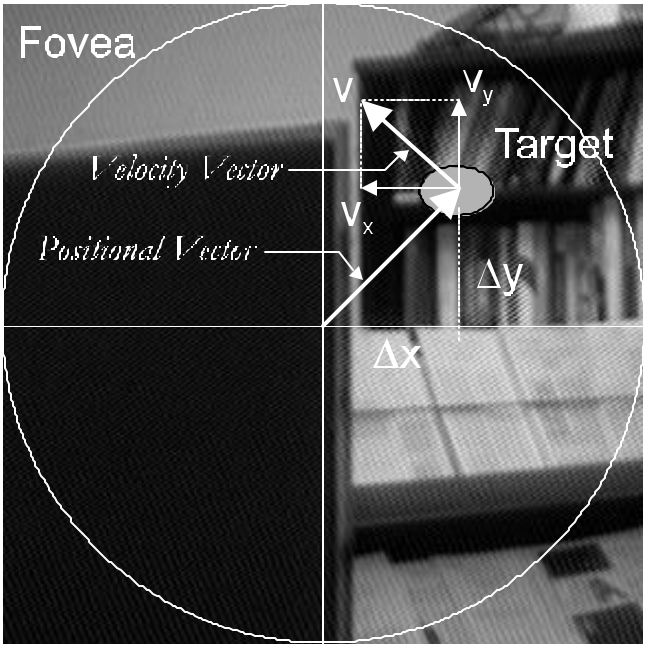


Figure 38: Positional vector and velocity vector used for smooth pursuit control

Target Position Prediction

Let ¢t be a time interval between two consecutive images being acquired by a frame grabber and ¥ be an estimated velocity of the target on a current frame at time t. The estimated position of the target on the next frame at time t + ¢t can be computed as:

$$¢x_{t+¢t} = ¢x_t + v_x £ ¢t \quad (23)$$

and

$$¢y_{t+¢t} = ¢y_t + v_y £ ¢t \quad (24)$$

By employing these estimated position of the target, the camera head can follow the moving target more smoothly. Other sophisticated position prediction methods such as Kalman ¯lter or ® ¡ ¯ tracker may be used to smooth the camera head movement.

This tracking, however, is performed at moderate speed. If the target goes out of the fovea area, it can be quickly brought back onto the fovea by the saccade process. Other positive prediction method are therefore not necessary.

Vergence

Vergence ensures that both left and right eyes ¯xate on the same target. Vergence control is then clearly de¯ned as a problem of minimizing disparity between left and right images from camera head. Disparity information is available from the visual attention network. This disparity in x-direction composes a 1-D disparity vector used to control left and right motors. (Note that disparity in y-direction is not taken into account because both left and right cameras are mounted on the same tilt axis).

Vergence Control using Disparity Signals

Let $\vec{d}_x$ be disparity vector in x-direction composed of disparity, d_x, measuring between a non-dominant camera (right) with respect to a dominant camera (left). The d_x represents shifted pixels between the image from the non-dominant camera and the dominant camera. The image from the non-dominant camera, nevertheless, is not a perfectly shifted version of the image from the dominant camera due to an o®-set distance between two cameras. This $\vec{d}_x$ controls the left and right camera to either diverge or converge. The left and right cameras move by half distance each (i.e. $\frac{d_x}{2}$). The images from the non-dominant camera are then a decreasingly shifted version of the image from the dominant camera. This means that if a disparity is estimated fairly well at a coarse resolution, the reduction of the disparity between two cameras will lead to a zero disparity ($d_x = 0$) once the target in both cameras

are ¯xated. Vergence control vectors for updating the left and right camera position are computed as:

$$x_l = x_{l\,old} + \frac{d_x}{2} \tag{25}$$

$$x_r = x_{r\,old} - \frac{d_x}{2} \tag{26}$$

where subscript l and r refers to left and right cameras.

Note that a negative sign in Equation 26 means left and right camera are always moving in di®erent directions (either divergence or convergence). The updated vectors, $\frac{d_x}{2}$ and $-\frac{d_x}{2}$, are served as inputs for a proportional control to generate proper motor commands for vergence control. Figure 39 shows the vergence control utilizing disparity vectors.

Vestibulo-Ocular Re°ex

The vestibulo-ocular re°ex involves the stabilization of the eye against changes in head position. This re°ex system keeps the eye looking at the same direction as it did before the head movement. VOR-like behaviors prevent the camera head away from its mechanical limit stops (i.e. verge motors on the ISAC head). For the ISAC head, the pan motor always turn toward the target. While the pan motor is moving, the VOR-like module keeps the verge motors on the target. The proposed system derives the camera head kinematics to properly counter-rotate the eyes while moving the head. Instead of directly solving camera head kinematics, the system can obtain camera head kinematics by help of the available tracking system. With some modi¯cations in setting up the tracking system along with saccade, smooth

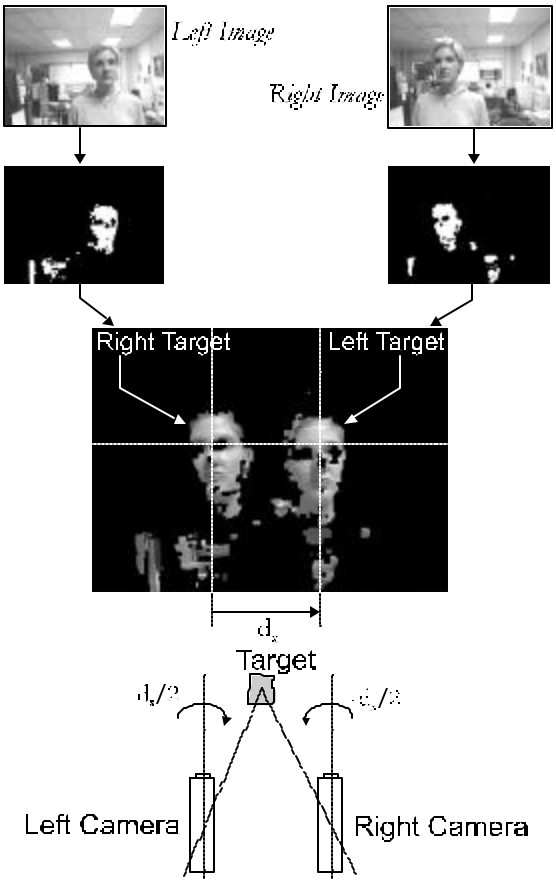


Figure 39: Vergence control using disparity vector

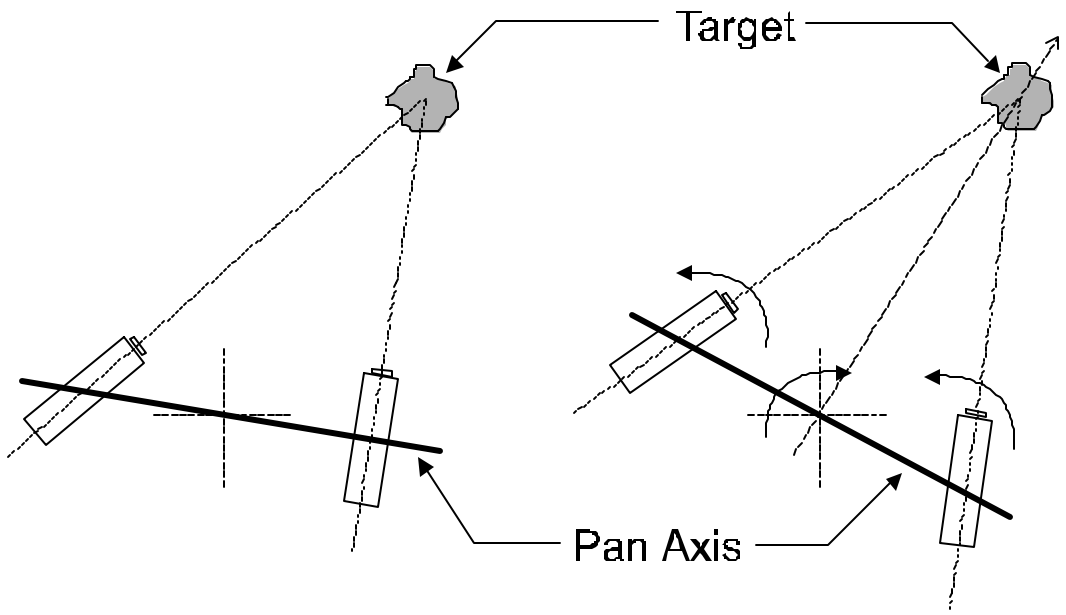


Figure 40: VOR scenario

pursuit, and vergence control, a VOR-like movement for controlling pan motor can be achieved.

Stabilizing Cameras while Head is moving

Figure 40 illustrates a scenerio of the VOR-like behavior. The system is designed to keep the pan motor moving toward the target while maintaining both left and right cameras on the target as close as possible. By keeping the pan motor directed toward the target, verge motors will not reach their mechanical limit stops. In this case, a moving pan motor is considered as a moving head.

Camera Head Kinematics using Active Vision

One simple solution for this the eyes stabilization problem here is to determine a relationship between left and right motor velocities and pan motor velocity. Once the

pan motor moves, proper motor commands for left and right motors can be computed. These commands are then issued to camera head controller for the compensation of the moving pan.

Considering camera head positions are sampled every ¢t milliseconds. Let the pan motor velocity be p degree/sec. Within ¢t ms, the pan motor has moved p£¢t degree. The problem is how to compensate for this amount of pan motor movement for left and right motors. If any accurate camera head kinematics are available, the problem would have been solved by only turning the left and right motors in di®erent directions. In order to achieve camera head kinematics, the available tracking system is assistantly employed. The processes can be outlined as follows:

1. Initialize the target in front of the camera head.
2. Perform tracking using the avaliable saccades, smooth pursuit, and vergence to maintain the target on the fovea.
3. While the target is stabilized on the fovea, move the pan motor with a small incremental.
4. The camera head then tracks the target and centers it.
5. Record positions of the pan, left and right motors (in degrees).
6. Repeat again with di®erent movements for the pan motor.

This strategy can achieve estimated left and right motor movements according to the pan motor movements. Once the pan motor is commanded to move toward the target, proper motor commands for left and right motors can be issued to compensate the pan motor movement.

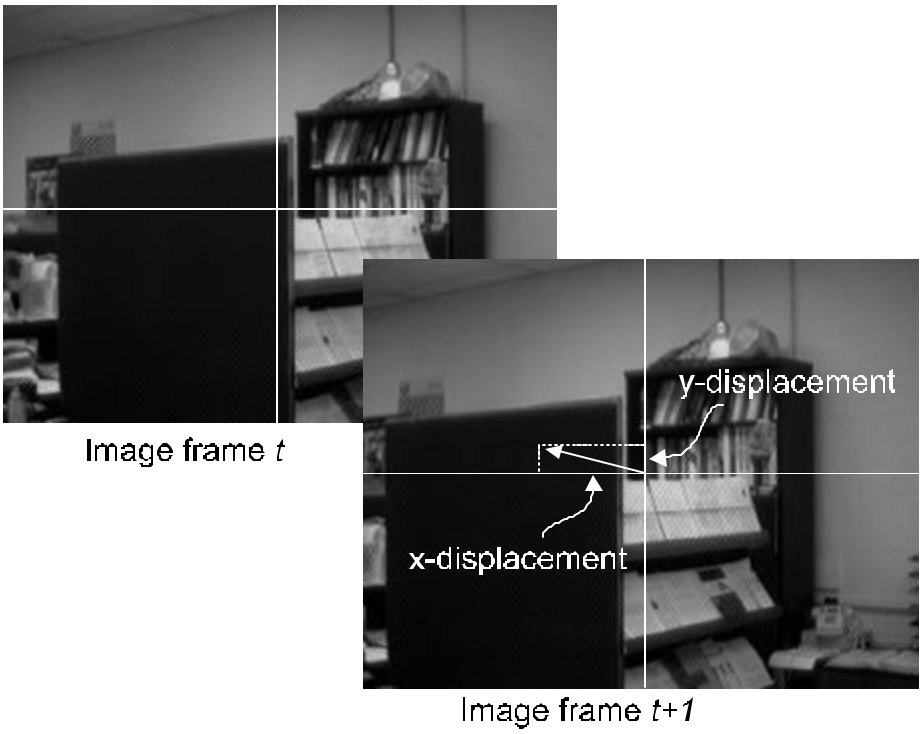


Figure 41: Image slip shows x and y displacement of successive frames

Opto-Kinetic Re°ex

The opto-kinetic re°ex is considered a backup system for the vestibulo-ocular re°ex. It stabilizes the eyes while the head is moving. The di®erences are (i) it is a slower process and (ii) only visual information is used for the head compensation. A simple OKR compensates for the head motion using information from an image slip. It requires a computation of the optical °ow on the background image in order to determine the image slip. The entire °ow image between successive frames provides an estimate of the head motion (see Figure 41). The OKR then uses x- and y-displacement to proportionally generate corresponding camera motor commands for counter-rotating the head movement.

Summary

Theoretical analysis of each main module has been described. These include elements in visual attention network and eyes motion center. Visual attention network has been de¯ned in its functionalities (i.e. providing information about a target such as position, velocity, and disparity). Eyes motion center has been designed as a core unit for controlling the camera movements: saccades, smooth pursuit, vergence, OKR, and VOR-like motions. It ties these four independent modules together to work and it also communicates with a camera head controller. In the following chapter, details in design and implementation discussed in this chapter are described, particularly the eyes motion center.

CHAPTER IV

DESIGN AND IMPLEMENTATION

Introduction

This chapter explains the design and implementation of the entire system in detail. This includes the ISAC head, camera head controller, and saccade, smooth-pursuit, vergence, OKR, and VOR-like camera head movements. The first section explains hardware specifications of the system, including the ISAC camera head and processing units. The camera head controller, which directly controls the camera head in the hardware layer and provides camera head positions to the rest of the system, is then described. Saccade motion is implemented based on a neural-network scheme for head-eye coordination mapping. Smooth pursuit uses information from a target such as position and velocity to assist in keeping the target on the fovea. Disparity between the left and right cameras is employed for vergence control. Resulting camera kinematic estimates from the available visual tracking system are utilized to achieve VOR-like behavior. Finally, system integration of all four functionalities is described. Specifications of the design and implementation code is listed in Appendix A.

ISAC Head

The ISAC head is built in-house at the Intelligent Robotic Laboratory of the Vanderbilt University. It has four degrees of freedom: left, right, pan and tilt. Two CCD color cameras are mounted on a common tilt axis. Each camera is steered by a verge motor (left and right) providing independent control from the pan motor. Figure

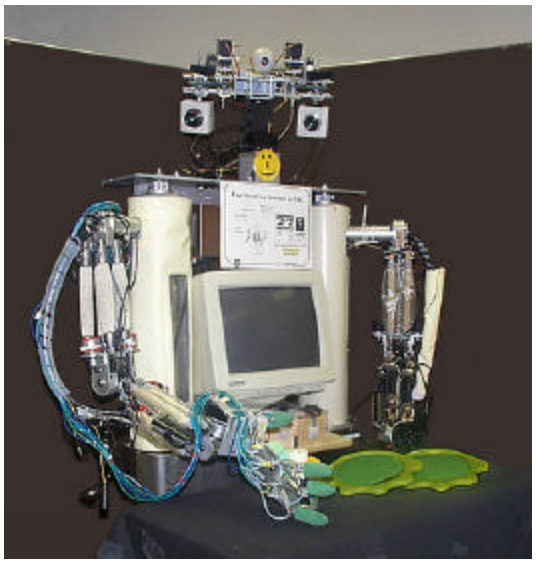


Figure 42: ISAC humanoid robot

42 shows the camera head mounted on the ISAC humanoid robot. In the following section, system hardware speci¯cations are described. Also, the ISAC camera head structure is examined in more detail.

System Hardware Speci¯cation

The system consists of two PC-based computers for controlling the visual attention network, the eye motion center and the camera head controller, as can be seen in Figure 43. The camera head is controlled by a dual PentiumPro ® 200 MHz machine via its RS-232 serial ports. This machine also does sampling on the current camera position from the camera head (left, right, pan and tilt motors). These motor positions are always available to the entire system. The second machine, a high-performance, dual Pentium III ® 500 MHz, is responsible for all image-processing routines. It acquires stereo color images (24 bits, 320£240 pixels) from the two cameras on the ISAC head using stereo color frame grabbers at a rate of 30 frames/second.

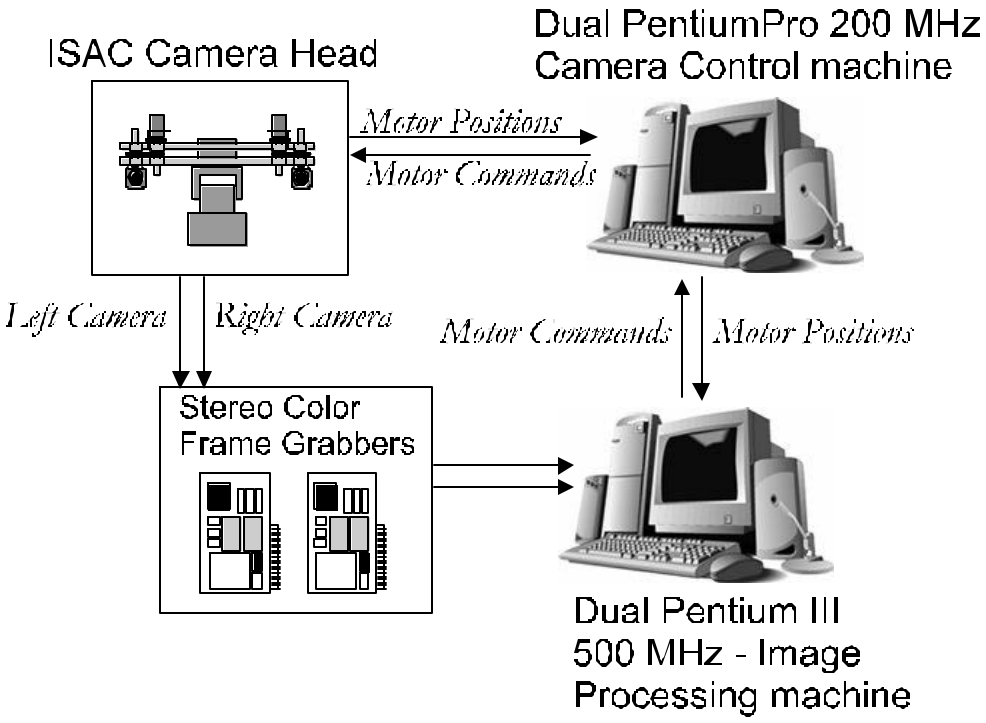


Figure 43: System hardware diagram

ISAC Camera Head Speci¯cation

Figure 44 shows the main units of the ISAC head. There are two main motor units: the pan-tilt unit and the verge unit. The pan-tilt unit is commercially made by Direct Perception [67]. Its performance is listed as follows.

- Maximum Rated Payload: over 4 lbs.
- Maximum Speed: over 300 degrees/second.
- Acceleration/Deceleration: Trapezoidal, on-the-°y speed and position changes.
- Resolution: 3.086 arc minutes.
- Tilt Range (approx): minimum 31 degrees up and 47 degrees down (78 degree range).

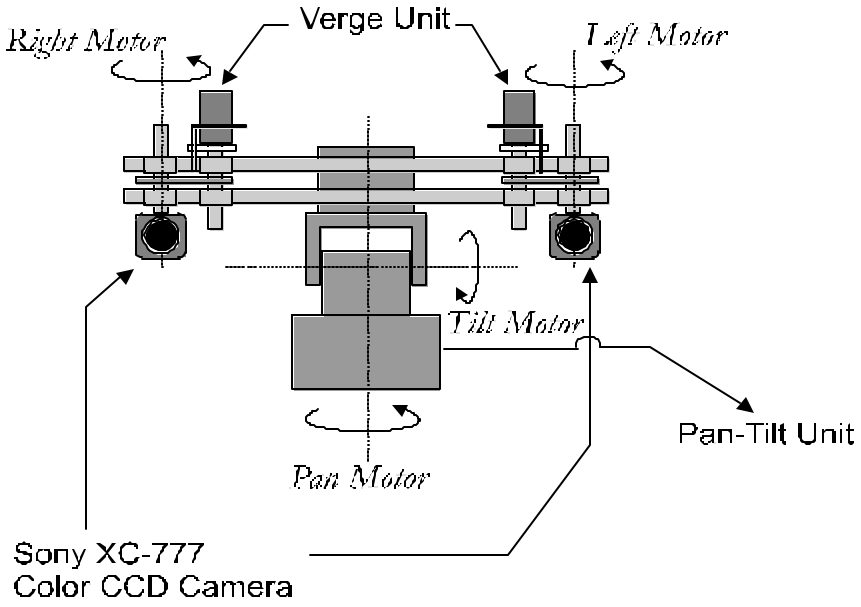


Figure 44: ISAC camera head

- Pan Range (approx): +159 degrees (318 degree range) with option of +180 degrees (360 degree range).

The verge unit is built in-house on the pan-tilt unit. It has separate motors for the left and the right camera. This provides two more degrees of freedom independently from the pan-tilt unit. The verge motors, however, are made from hobby motors. Desirable speed and accuracy are therefore not expected.

Camera Head Controller

The design of the camera head controller directly concerns hardware controls. In this system, the camera head controller should have the following properties:

- Left, right, pan and tilt positions should be accurate and available for the rest of the system to acquire on the °y.

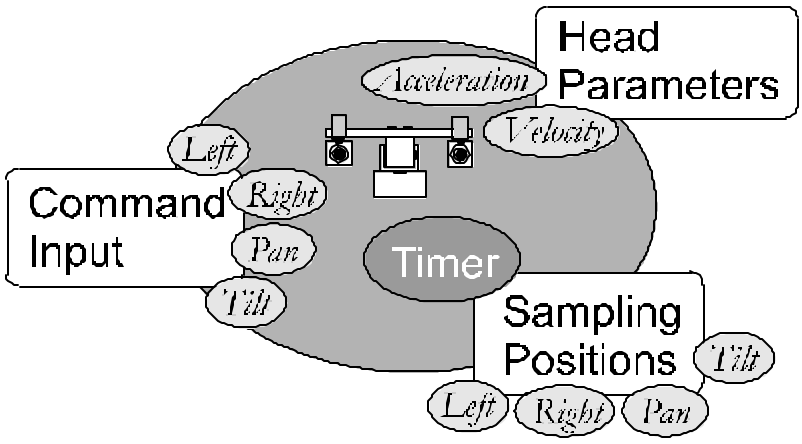


Figure 45: Camera head controller design

- Left, right, pan, and tilt motors are independently controlled.
- Velocity and acceleration are controllable.

Figure 45 shows a diagram of the current camera head controller in the system.

The current camera head system consists of a PC-based machine that communicates with the head via two standard RS-232 ports. The first port controls the left and right motors and the second the pan and tilt motors. The camera head reacts to command inputs and swiftly moves corresponding motors. The camera head positions are sampled at a rate set by a timer. A normal rate used in the system is 5 milliseconds. Only the pan-tilt unit, which is commercially made, has controllable velocity and acceleration.

Saccade

Saccade control comprises two separate modules: the saccade map trainer and the saccade command generator (see Figure 46). The saccade map trainer provides

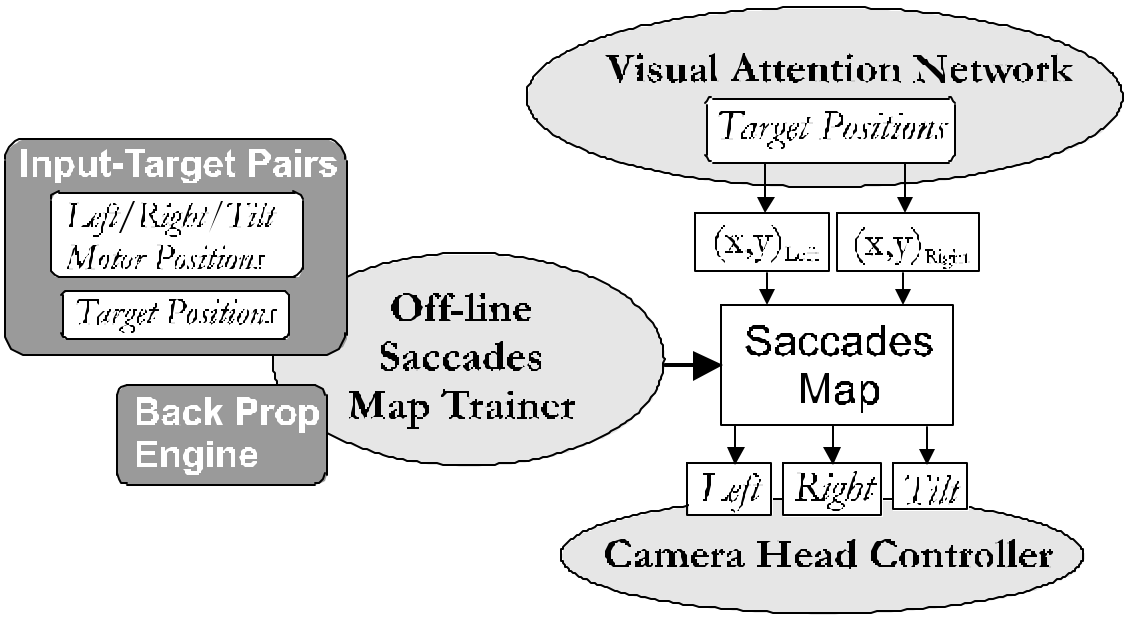
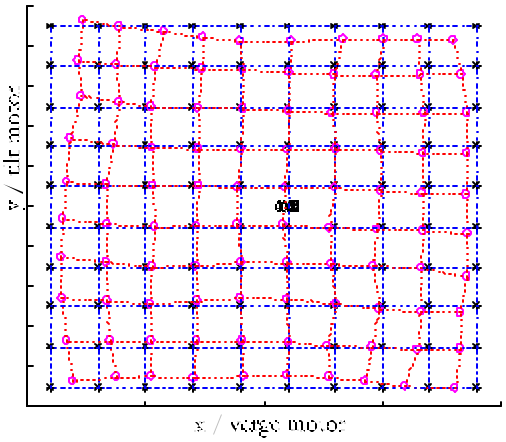


Figure 46: Implementation of saccade control

an updated saccade map for the saccade command generator. The neural-network-based trainer is utilized from an o®-line process. The saccade command generator then uses this map to generate saccade commands for the left, right and tilt motors corresponding to the target position in the image plane.

Saccade map training is performed o®-line using back-propagation learning. The left and right cameras are trained separately because they are controlled independently. Each camera is trained with two inputs, namely the x and y positions of the target, and two outputs, the corresponding left or right motor and tilt motor commands. Sample input-target pairs are shown in Figure 47-(a). The network comprises 25 neurodes in the middle layer. The training goal for a sum-square error is set to be 0.02. The resulting training map is displayed in Figure 47-(b). The result shows a non-linear mapping characteristic.

Initial Sample Points for Training Saccades



* = Image Coordinate (pixel)
o = Corresponding Motor Command (degree)

Figure 47: Saccades map training

The saccade command generator uses the saccade map to calculate motor commands corresponding to target-position inputs. It employs a standard feed-forward network to simulate outputs for left, right and tilt motors. These motor commands and others from different units are then gathered by the eye-motion center to control the camera head.

Smooth Pursuit

Smooth-pursuit movement utilizes target position and velocity as clues for maintaining the target on a fovea at a moderate speed. The following section describes the smooth-pursuit movement implementation using a simple proportional control. Additionally, a low-pass filter is implemented for smoother motor controls. This prevents overshoot, which could lead to oscillations and unstable states for camera head motors.

Smooth Pursuit Movement

The smooth-pursuit component utilizes information about target position and velocity to proportionally adjust the camera head at moderate speed. Figure 48 shows a diagram of a control scheme. The visual attention network module provides information about the target in both images. These are measured in pixels. Left and right motors move proportionally to both x-axis distance of the target from the centers of the left and right images respectively. In addition, both target velocities in the x-direction from the left and right images are appended to control these verge motors. Constant gain K_L and K_R for left and right motors are applied before final verge commands are issued. Similarly, target offset and velocity in the y-direction are

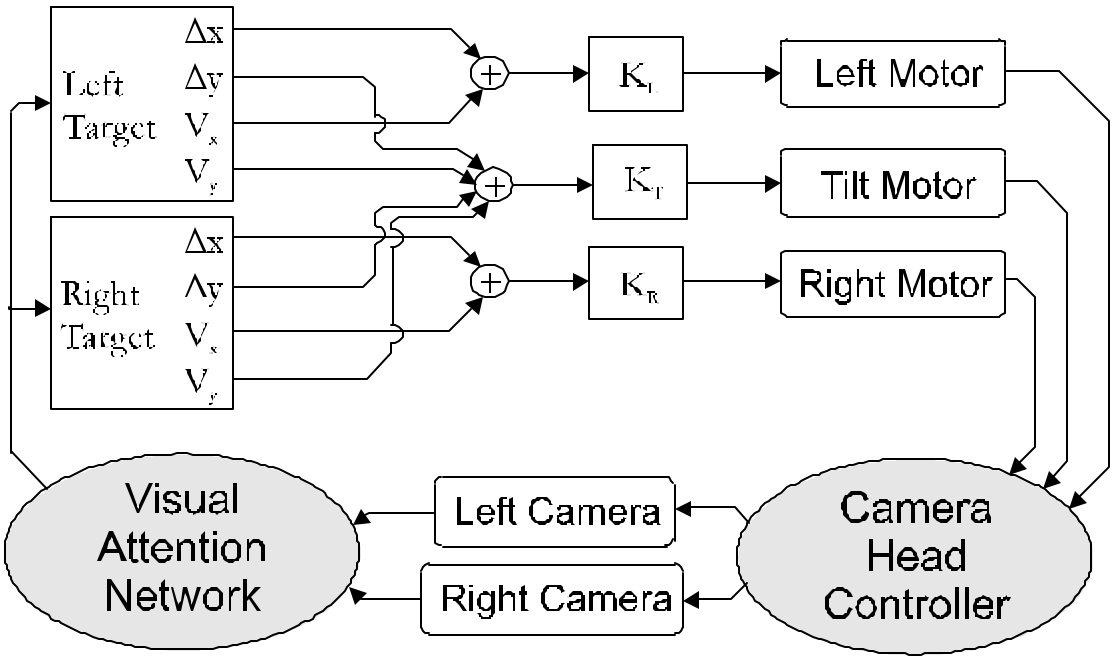


Figure 48: Smooth pursuit control

used to control the tilt motor. Since left and right cameras are mounted on the same tilt axis, inputs from the left and right cameras are averaged and then multiplied with constant K_T for a final tilt motor command.

Low-Pass Filter for Smoother Motor Movements

The camera control scheme only utilizes information from the vision system. On the other hand, no feedback about the camera head motor information is used in controlling the camera head. Consequently, unstable behaviors may occur. An important concern is oscillation problems. Because the target cannot be predicted at every frame of images (as the target may move in one direction and suddenly change course into the opposite direction), there is a chance of camera motor overshooting. A low-pass filter for motor signal is then implemented to reduce overshoots or jerky

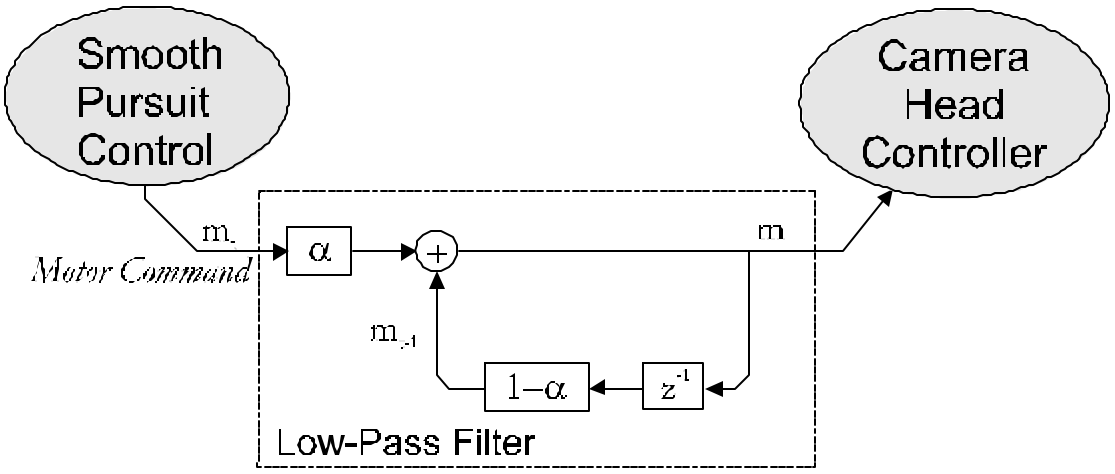


Figure 49: Low-pass ¯lter for motor signals

moves during camera head moving (see Figure 49). The following simple low-pass ¯lter is applied to each motor command before it is sent to the motor.

Let m_t be a current motor command and $m_{t\,i\,1}$ is a previous motor command. Note that a motor command is an absolute value for motor positions, i.e. for motor command m_t, it means move the motor m_t degrees from its zero-degree reference. The low-pass ¯ltered motor signal, m, can be computed as:

$$m = ® £ m_t + (1 \; ¡ \; ®) £ m_{t\,i\,1} \qquad (27)$$

where ® is a ¯lter constant and $0 \cdot ® \cdot 1$. If $® = 1$, $m = m_t$. This means no previous motor command is taken into an account, i.e. no e®ect from the ¯lter. With this low-pass ¯lter characteristic, motor signals become smoother with less overshoot and jerky movement. Parameter ® is empirically tuned during the tracking to obtain the best performance of the camera movement (in term of stability).

Vergence

Vergence control utilizes disparity clues to accommodate left and right motors. The following sections describe details of disparity estimation and vergence control using the disparity signal. A 1-D correlation-based disparity estimation algorithm is employed for disparity estimate. This algorithm is designed to be fast and e±ciently accurate. An implementation of vergence control using this disparity signal is then explained.

1-D Correlation-based Disparity Estimation

Disparity estimation is implemented based on a 1-D correlation algorithm. The algorithm can be outlined as follows:

- Gray-scale and segmented images from left and right cameras are convolved using an AND operator. Resulting images are employed as inputs (see Figure 50). The size of all images is 320 £ 240 pixels.

- Intensities of both images are projected onto the x-axis. Outputs of projection are normalized. Figure 51 shows plots of both left and right normalized projection signals.

- Both left and right projection signals are then correlated. The result of the correlation yields a signal of which its peak represents a matching point between two input signals, as can be seen in Figure 52.

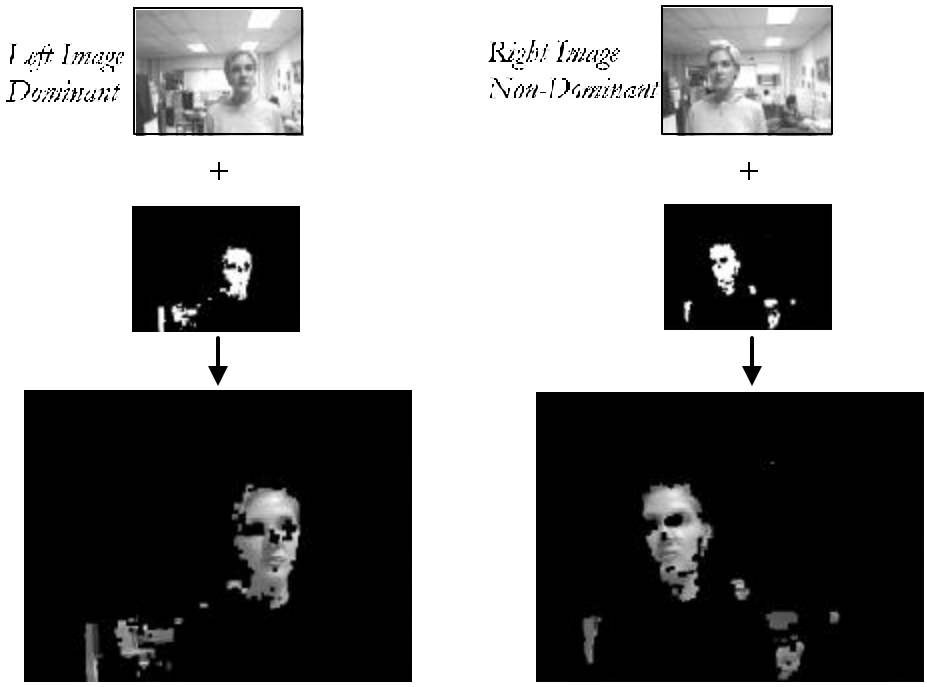


Figure 50: Combining of gray scale and segmented image from left and right camera

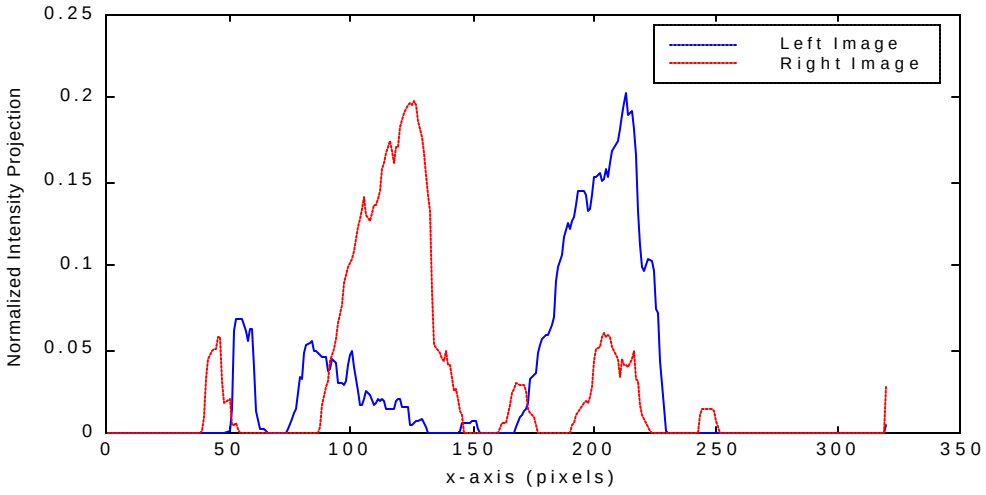


Figure 51: Intensity projection of left and right images onto x-axis

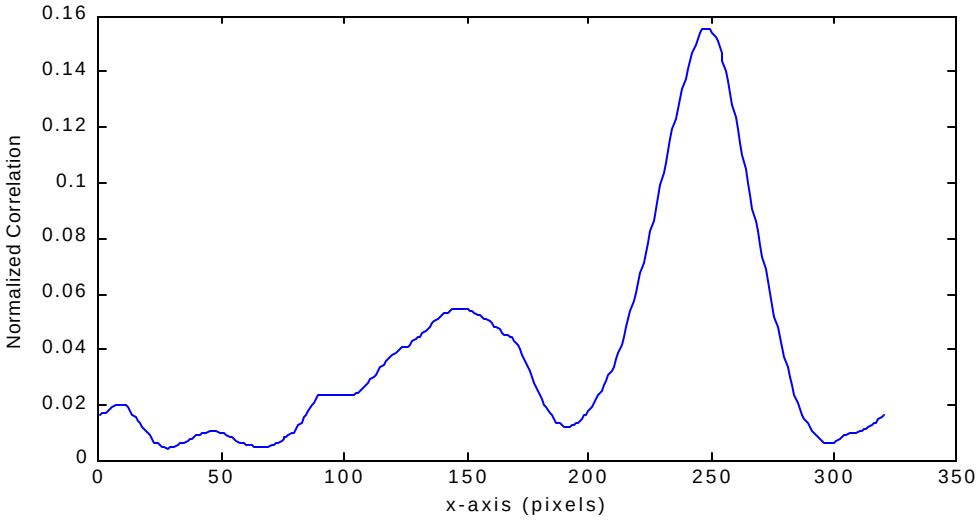


Figure 52: Correlation between left and right projected images

² From the correlation signal, disparity between left and right images can be estimated as a distance from the matching point to the center of the signal. From Figure 52 the matching point position locates at 247th pixel. Consequently, the disparity is equal to 160 ¡ 247 = ¡ 87 pixels.

This value of disparity is then used for vergence control to minimize this disparity between left and right cameras.

Vergence Control

Vergence control interprets the disparity signal into motor commands. Left and right cameras are moved in a direction that reduces this disparity magnitude. The vergence control uses the half size of this parity magnitude for tuning a proportional controller. Note that the sign of the disparity signal indicates the direction for verge motor control. There are two cases for left and right cameras to be moved together: divergence and convergence. In both cases the left and right cameras always move in

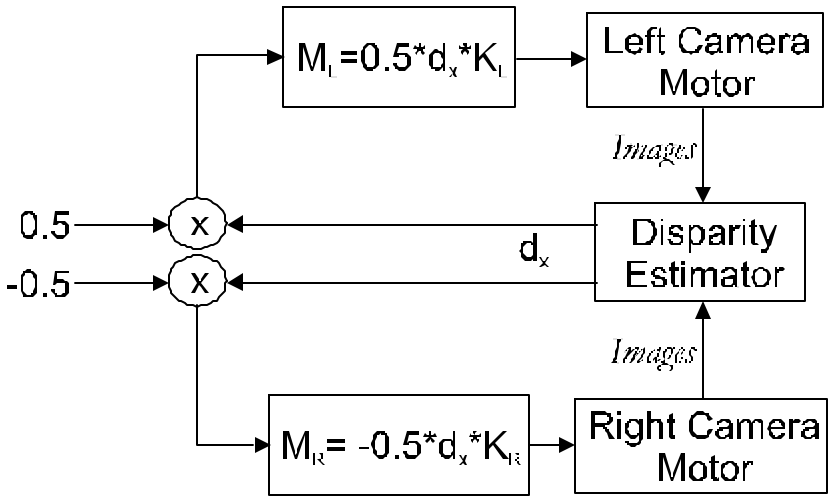


Figure 53: Vergence proportional control

opposite directions. Divergence turns the left camera to the left and the right camera to the right, while convergence turns the left camera to the right and the right camera to the left. Figure 53 shows a diagram of the vengence proportional controller.

M_L - Left motor command (degree)

M_R - Right motor command (degree)

K_L - Left constant gain (parameter)

K_R - Right constant gain (parameter)

d_x - Disparity value (pixels)

Equivalent Vestibulo-Ocular Re°ex

Left and right motor movements according to the pan motor movement are estimated with assistance of the visual tracking system. The visual tracking system keeps left and right cameras on the target while a VOR trainer observes camera head motors. Once the pan motor is commanded to move toward the target, proper motor

commands for left and right motors can be issued to compensate the pan motor movement. The VOR-like module is implemented separately in two parts: VOR tuning and VOR simulation. The VOR tuning is performed o®-line to observe and obtain camera kinetics. This allows the system to adjust relationship parameters between the verge motors and the pan motor. After all parameters are con¯gured, the VOR simulation uses them to stabilize the eyes in the tracking phase.

VOR-Tuning

VOR tuning is a training process to observe relationship parameters between the verge motors and the pan motor. The process starts from generating a visual target. Then the available tracking system is employed to stabilize the eyes on the target while the VOR tuning controls the pan motor in di®erent angles and records the verge and pan motors. Readings from the motors are then examined to use in the VOR simulation.

VOR-Simulating

After the VOR tuning is performed, a relationship between the verge and pan motors is observed. This relationship can be represented by:

$$verge(pan) = f(pan) \tag{28}$$

where verge and pan are measured in degrees.

Preliminary results from experiments show that the relationship between the verge and pan motors is likely to be linear and can be formulated as:

$$verge(pan) = K \times pan \tag{29}$$

where K is a VOR gain constant and verge and pan are examined in the ¯nest resolution.

The VOR simulation employs this VOR constant as a proportional parameter to adjust the verge motors for compensation while trying to move pan motor toward the target.

Opto-Kinetic Re°ex

Opto-kinetic re°ex using displacements between successive images to generate corresponding camera motor commands. Consequently, there are two main part to consider: (i) displacements between successive images and (ii) proper motor commands to compensate the head movement. The displacements between successive images can be obtained by optical °ow computation of the background images. This system, however, utilizes the available 1-D correlation module to calculate the entire image slip. The displacement vector is then used in the OKR proportional control for corresponding camera motors. Due to verge motors are mounted on the common tilt axis, the compensation for the head movement occurs only in x-direction. This reduces load of the system because only the image displacement in x-direction needs to be computed.

Successive Image Displacements using 1-D correlation

The 1-D correlation technique has already been mentioned in vergence section. The uses of this technique to calculate the camera head motion can be achieved with some modi¯cations and can be summarized as follows:

² Let two successive gray-scale images from each camera are acquired at time t

Image frame *t* Image frame *t+1*

Figure 54: Successive images at time t and t + 1

and t + 1 (see Figure).

- Intensity values of both images are projected onto the x-axis. Outputs of projection are normalized. Figure 55 shows plots of both normalized projection signals.

- Both projection signals are then correlated. The result of the correlation yields a signal of which its peak represents a matching point between two input signals, as can be seen in Figure 56.

- From the correlation signal, displacement between successive images can be estimated as a distance from the matching point to the center of the signal. From Figure 56 the matching point position locates at 179th pixels. Consequently, the x-displacement is equal to 160-179=-19 pixels. The sign of the displacement indicates the direction of the head movement.

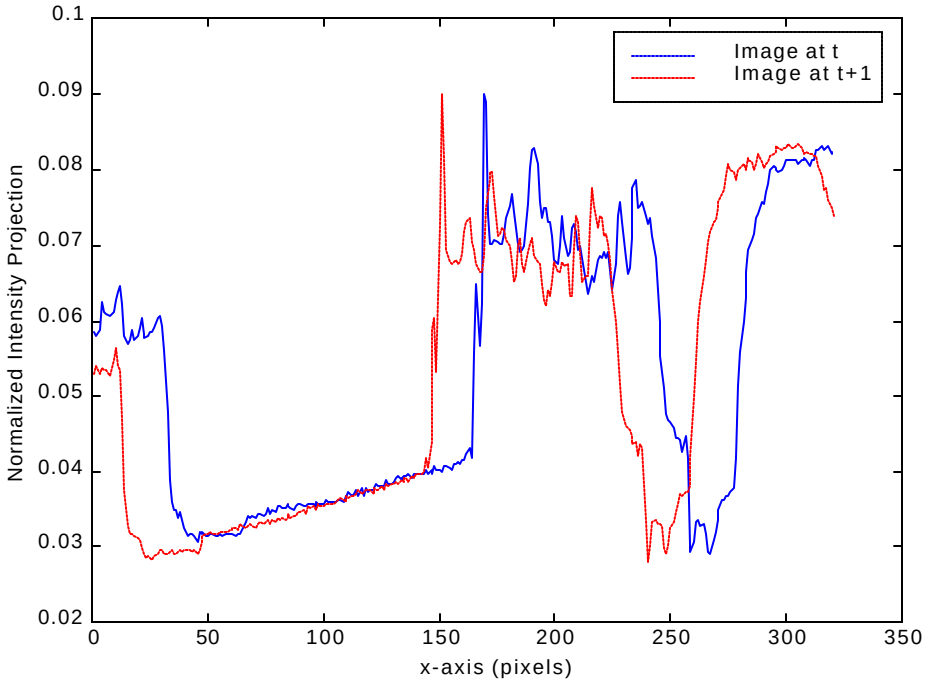


Figure 55: Normalized intensity projection of image at time t and t + 1

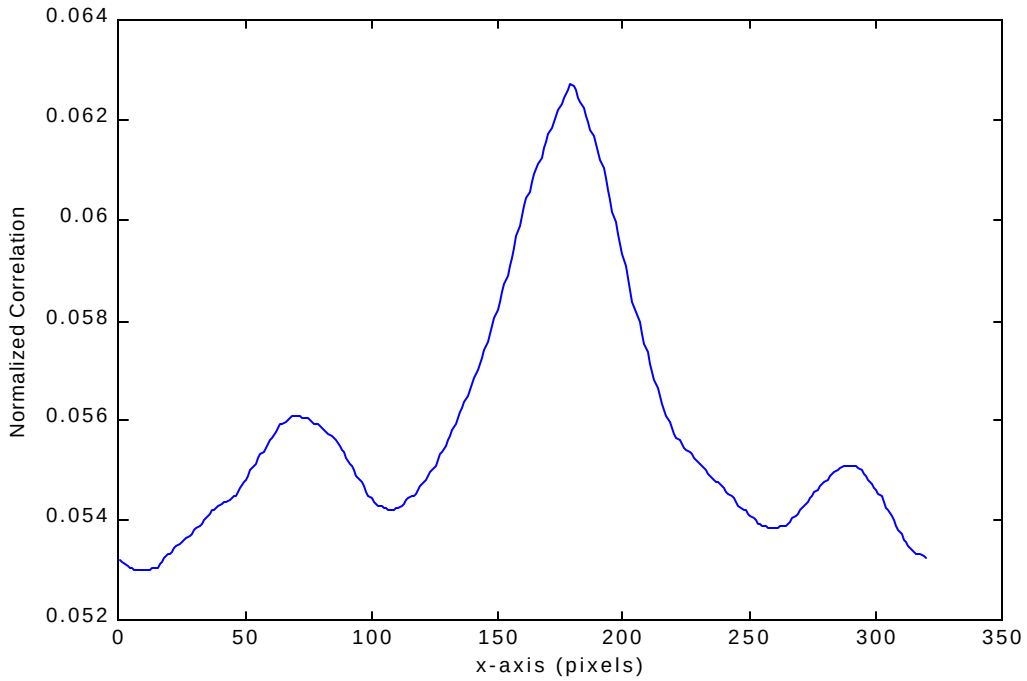


Figure 56: Normalized correlation of projection signals from successive images

The image at time t + 1 is said to be a shifted version of the image at time t. The shifted pixel values, called displacement signals, from both left and right camera are used as an input to the OKR controller.

Opto-Kinetic Re°ex Controller

The OKR controller uses displacement signals to generate proper motor commands for corresponding motor. While the pan motor is moving toward the target, OKR compensates for the left and right movements to keep the eyes on the target. The OKR controller utilizes a proportional control to move the motors.

System Integration

All proposed human eye motion-like controls for the ISAC camera head have been implemented separately. The system, however, requires integration of all components. Each module uses and shares resources available in the system. Also, each module activates in di®erent manners. The following diagram shows resource uses and states of each component in the overall system.

As can be seen from the diagram, left and right color CCD cameras are mounted on the camera head. Color images are acquired and color segmentation is performed constantly. These images are utilized by the disparity estimator, the target motion estimator and the target position locator. All of them are executed as fast as possible for up-to-date information from current images to be available to the rest of the system. This information includes a disparity signal from the disparity estimator, a velocity signal from the target motion estimator and a positional signal from the target position locator. The vergence component uses only the disparity signal to

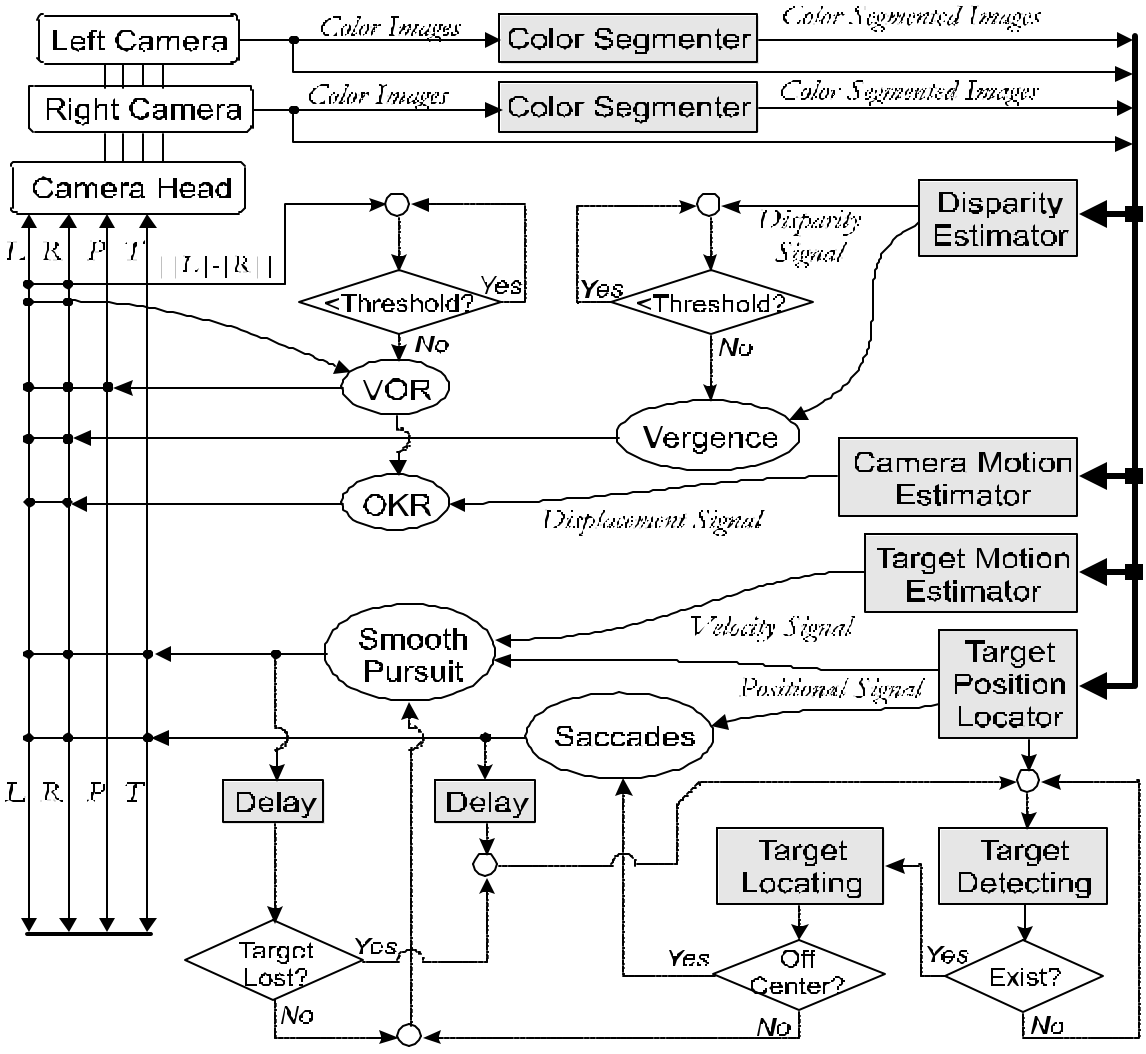


Figure 57: System integration diagram

calculate for left and right motor commands. The smooth-pursuit component utilizes both positional and velocity signals to drive left, right and tilt motors. The saccade component shares only the positional signal to control left, right and pan motors.

The system starts from an inactive state. Once a target is detected in a scene, its position is located to check whether it is inside or outside the center of the image (de¯ned as fovea). If the target is found outside the fovea, the saccade component will activate to rapidly intercept the target. Because images are usually blurred by the camera motion, an empirical value of delay time is then added to the system to wait for the saccade results. If the saccade motion is satisfactory (target is on the fovea), the smooth-pursuit component is then activated to keep the target onto the fovea. This action is performed until target is lost (leaves the fovea). The system then reverts to its starting state.

The saccade and smooth-pursuit components work closely together while the vergence and VOR components each run independently. The vergence component performs checking on a disparity signal to control the verge motor so that the disparity is kept below a threshold. Similarly, the VOR component keeps the absolute di®erence between the value of the left and right motors below threshold. This is because the VOR component always moves the pan motor toward the target. Once it reaches the target, the left and right motors should have an equal value. The OKR component performs together with the VOR component. It obtains the displacement signal to control the verge motor for keeping them on the target while the VOR is driving the pan motor.

Each component in the system is implemented to be adjustable. The entire system is run and individual components tuned to yield the best performance.

Summary

Details of design and implementation of the system have been discussed. Hardware details of the ISAC head have been examined to provide better background for understanding the design of the rest of the system, including the camera head controller. The entire system implementation has been described, along with the design of its components. These include saccade, smooth-pursuit, vergence, OKR, and VOR-like components. Special techniques have been employed to overcome problems that arose during implementation. System integration with finite-state machines has been shown, as configured for saccades, smooth pursuit, vergence, OKR and VOR.

CHAPTER V

EXPERIMENTAL RESULTS

Introduction

In order to demonstrate the overall performance and the human-like behaviors of the proposed active vision system (AVS), several experiments have been conducted. These involved active saccades and smooth pursuit tracking, vergence control using disparity clues, and eye compensation against head movements. The overall robustness and performance of the system are characterized. The results are compared to the previously used camera head control, as well as to other well-known binocular systems. In addition, the proposed system was partially implemented on a di®erent robot platform to test portability. Comparisons between this system and human eye movement system are also discussed. Finally, weaknesses of the proposed system that have been noticed during the experimental sessions are discussed.

Experimental Setup

This section describes the experimental setup to evaluate the system. Note that the aim of the experiments is to observe the performance and behavior of the camera movements. Consequently, information about the target needed for the AVS to perform tracking, such as its position in the image plane, is required. An example of target passive tracking (without camera movement) is shown in Figures 58 and 59.

The image sequence was taken at roughly two frames a second. Figure 58 also

shows corresponding color-segmented images. The white blob in the segmented images represents the target, which in the experiment is a green glove. The position of the blob in the image is determined using the L_1 norm [68][69]. Figure 59 shows a plot of the target x- and y-position errors in the image plane. The target follows a circular trajectory in the image; its position plot has a sinusoid shape. Note that the target coordinates are not filtered. By using color segmentation, the target is robustly located, regardless of its motion in the image.

The plot of the target position error represents the target distance from the center of the image as a percentage, where zero percent means the target is perfectly located at the center of the image. As the target moves away from the center of the image, the percentage of the target position error increases. At the border of the image, the target position error is equal to 100%. In these experiments, the dimension of the sampling image is 320×240 pixels. Hence, the center of the image is $(160; 120)$. This yields the maximum target position error (100%) of ± 160 pixels for the x-axis and ± 120 pixels for the y-axis.

Experiments

In this section, the experiments on the AVS are described. Each experiment is designed so that the AVS performs an action, thereby allowing the basic behavior of each eye movement to be observed. There are two types of behavior: tracking and non-tracking. Tracking behavior includes saccades and smooth pursuit. Vergence, vestibulo-ocular reflex (VOR), and opto-kinetic reflex (OKR) are secondary to the main tracking system. Because they are not used mainly for tracking purposes, they are non-tracking behavior. The vergence system is used to adjust the verge motors

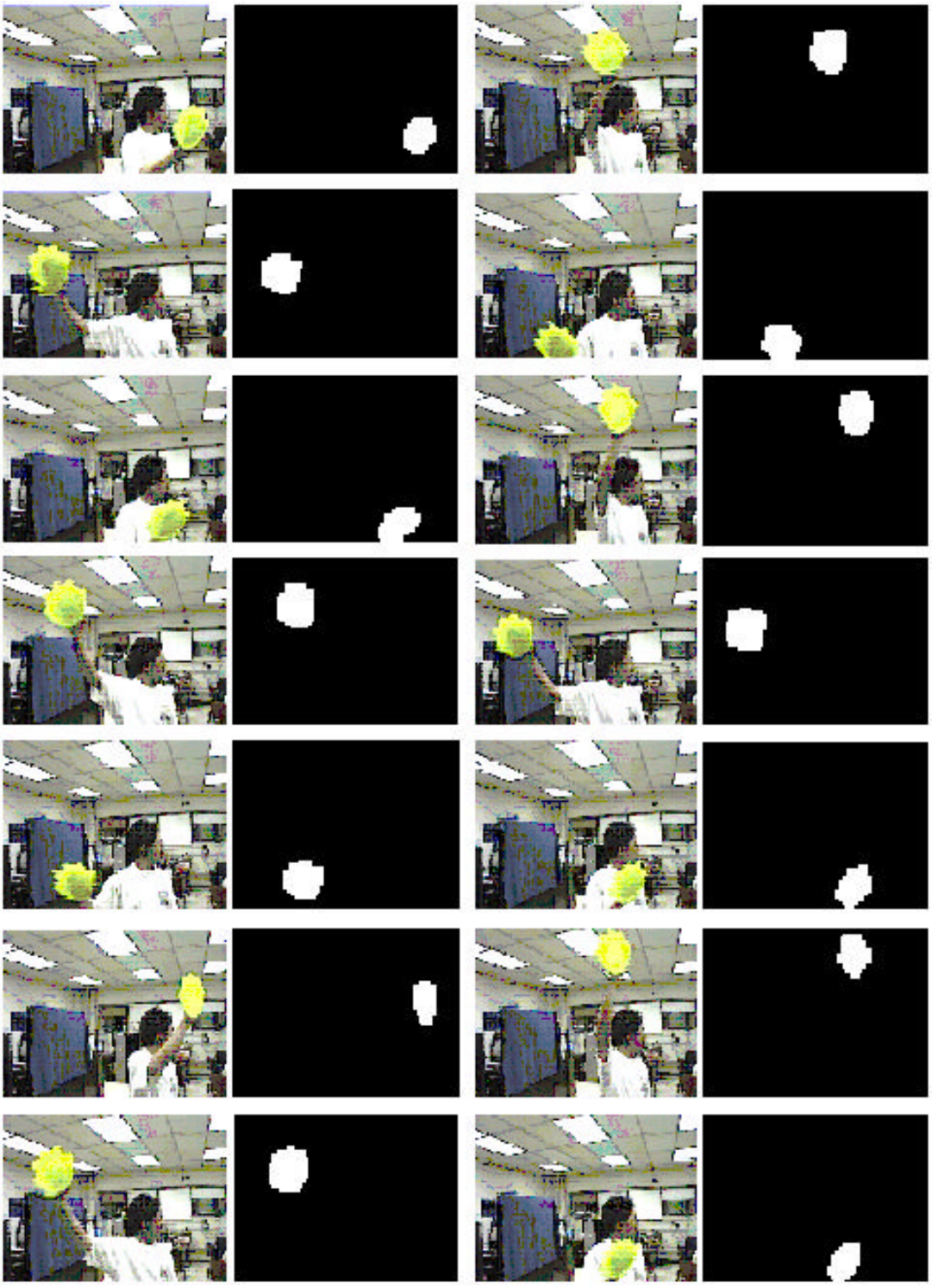


Figure 58: Image sequence taken every 2 seconds during passive tracking (with no camera movement)

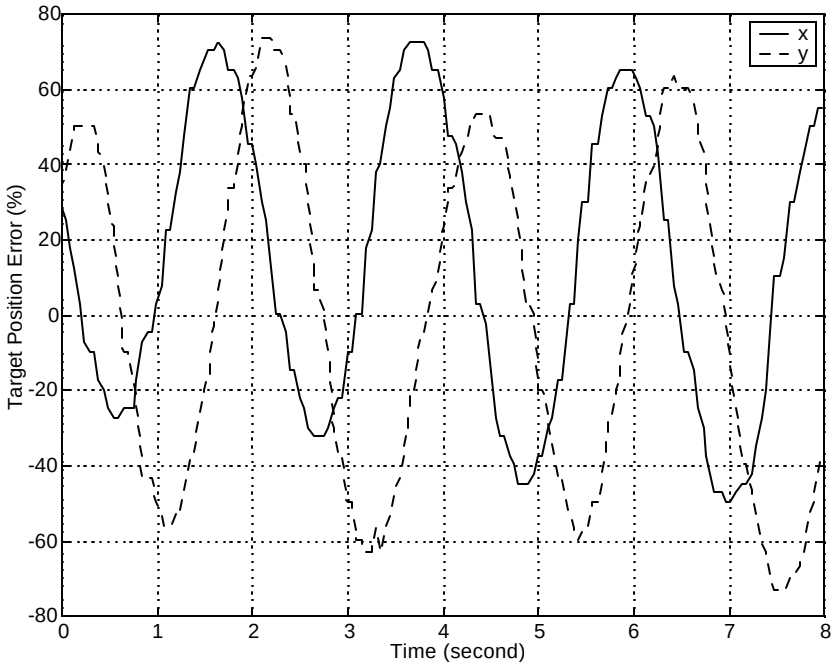


Figure 59: Passive tracking: target position error

toward zero disparity between left and right images, VOR is used to stabilize the verge motors against the pan motor movement, and OKR is used as a backup for VOR to compensate for head motion.

Tracking Behavior

Saccades

A saccade is a rapid movement of the eyes to intercept the target. In this experiment, the target was quickly moved and stopped within the workspace. When the target moves out of the saccade zone in the image plane, saccade commands are automatically issued. The cameras then ¯xate on the new location of the target. The results of this experiment are displayed in Figures 60, 61, and 62.

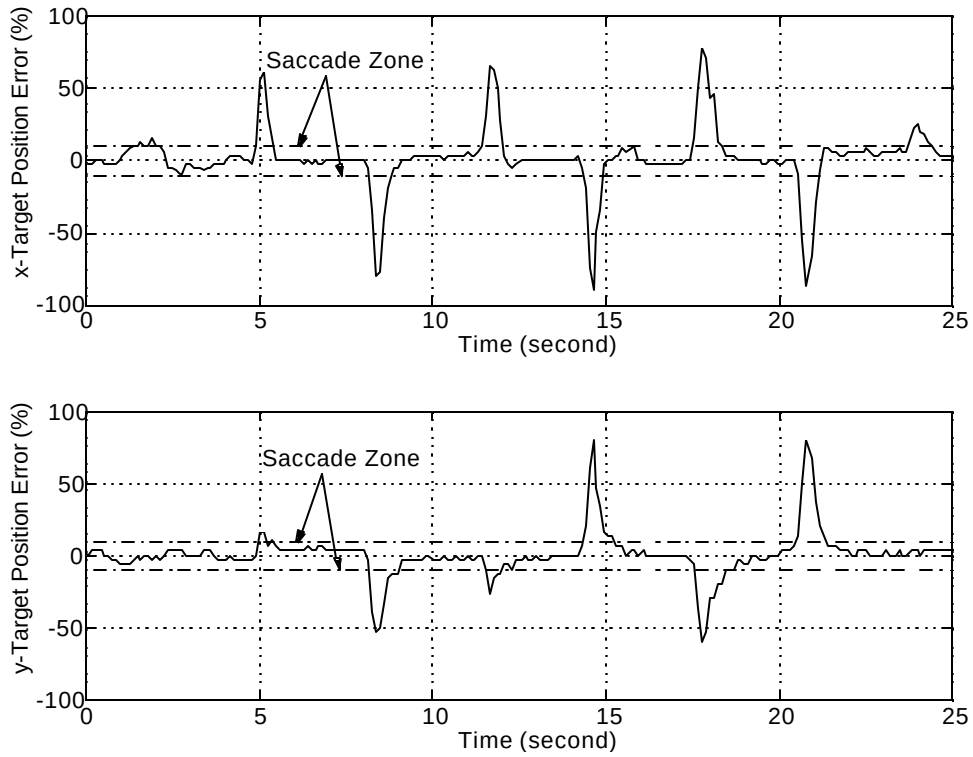


Figure 60: Saccacde: left target position error during saccade motion

Figures 60 and 61 show the target position error in the x- and y-directions, respectively, during the 25-second duration of the saccade experimental session. The corresponding motor positions are shown in Figure 62. The target was moved out of the center of the image roughly at time t = 5; 8; 11:5; 14; 17:5; and 20:5 seconds. Shortly after the target was moved out of the saccade zone, saccade commands were issued automatically by the control system. The camera then moved to intercept the target. It can be seen from the graph plots that the target position error returns to zero after the saccade movement of the motors. The saccade zone is equal to 10% in this experiment (16 pixels in x-axis and .12-pixels in y-axis)

Figures 63 and 64 show one period of saccade motion. The saccade sequence can be summarized as follows:

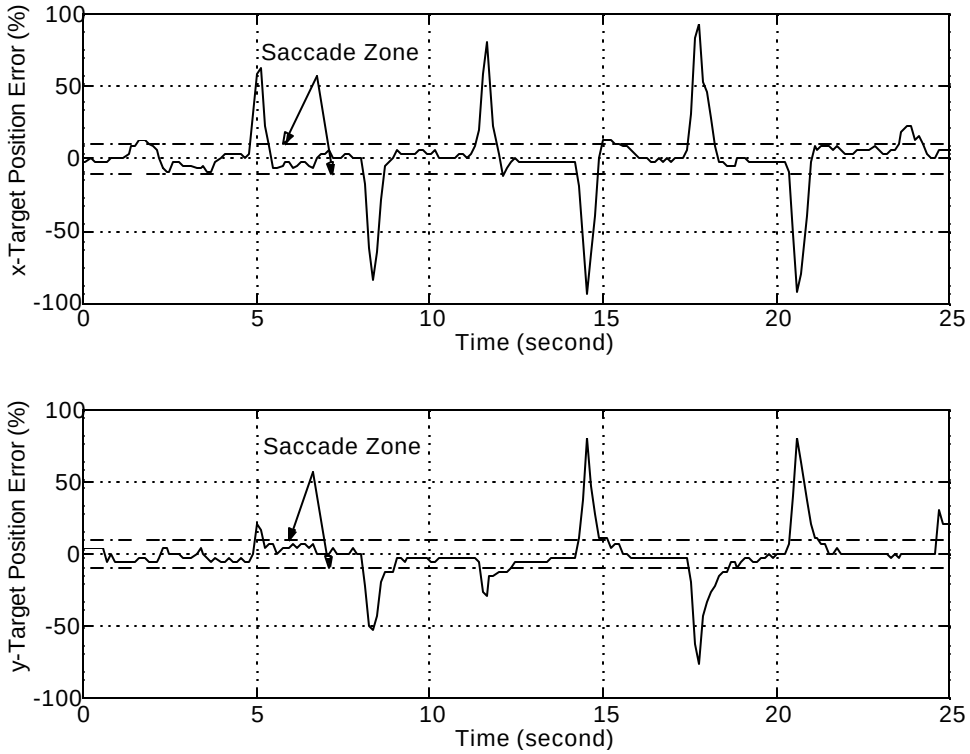


Figure 61: Saccade: right target position error during saccade motion

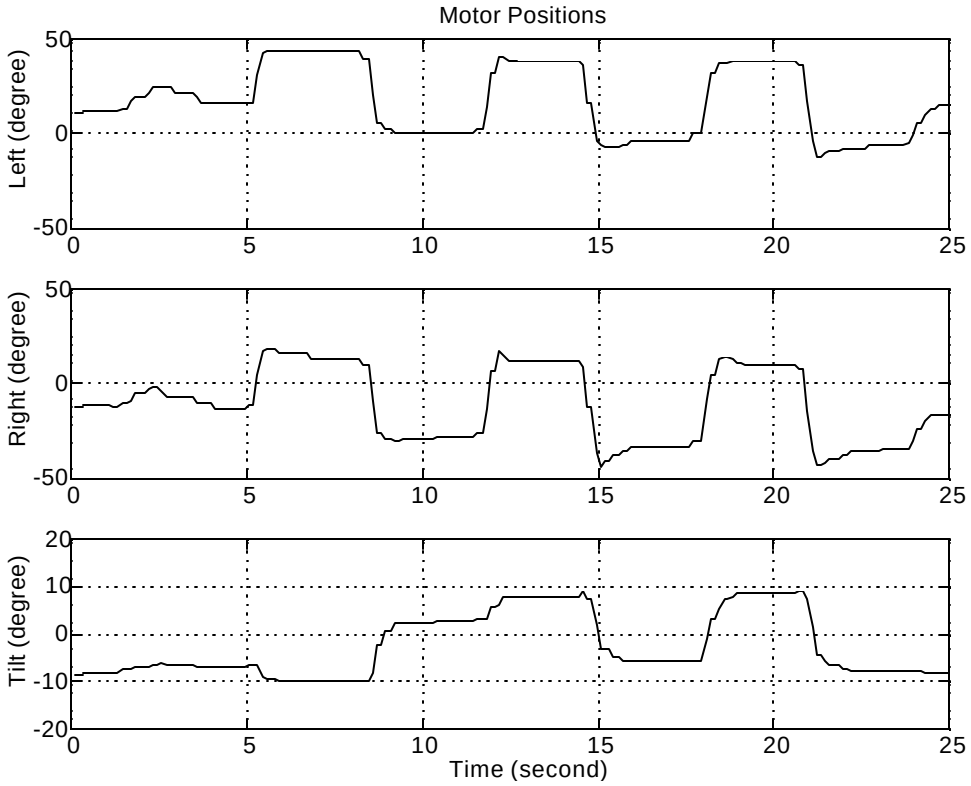


Figure 62: Saccade: motor positions during saccade motion (left, right, and tilt)

- From time t = 0 ¡ 1:5 seconds, the target is initialized at the center of the image.
- From time t = 1:5 ¡ 1:8 seconds, the target moves away from the center of the image.
- At time t = 1:8 seconds, the saccade command is ¯red.
- From time t = 1:8 ¡ 2:2 seconds, the camera motors respond to the saccade command and move to intercept the target.
- At time t = 2:2 seconds, the saccade motion is completed. The camera ¯xates on the target.

Note that the target position error passes the saccade zone, which is set at a 16-pixel radius from the center of the image, at approximately 1:6 seconds. The saccade command, however, ¯res at time t = 1:8 seconds. This indicates that there is a delay between the visual processing and the camera controller module. The average of this delay time was roughly estimated to be at least 200 milliseconds (see saccade experiments with single and double step stimuli for more detail).

Besides the precise ballistic movement of the camera to bring the target onto the fovea (a small area at the center of the camera image plane), this neural-network-based saccade control also results in desirable characteristics of the motors' movement. The result from Figure 63 shows that the saccade function generates motor commands that move the motors to ¯xate on the target with no overshoot. This quick movement also shows a very small undershoot characteristic of the motors' movement, as

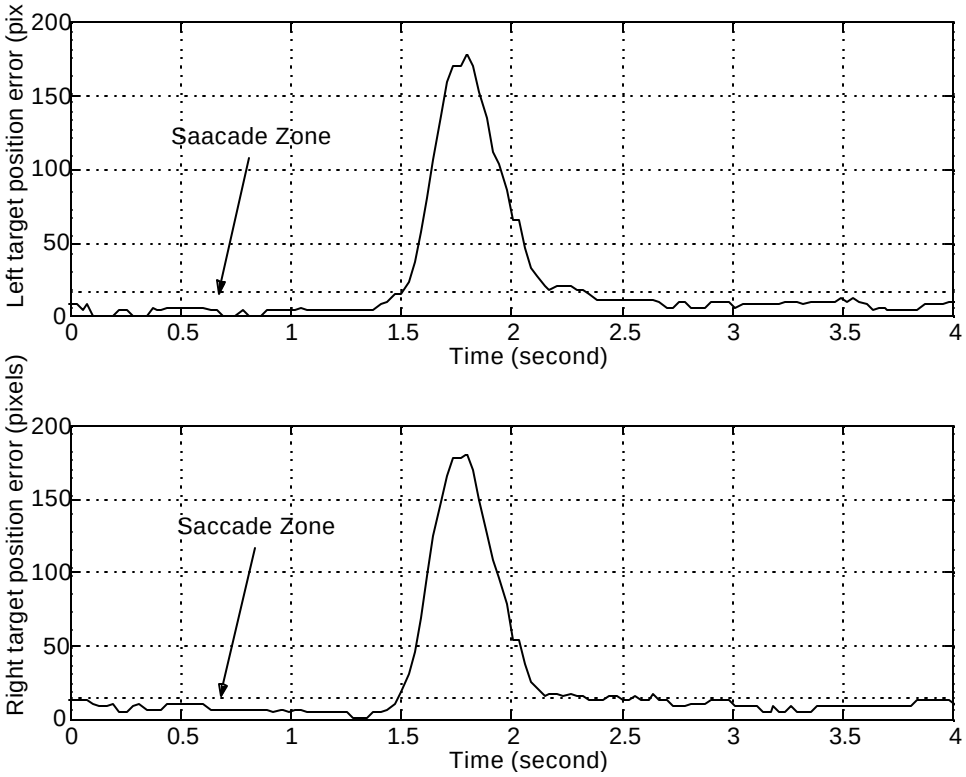


Figure 63: Saccade: target position error during one period of saccade motion

a consequence of the algorithm. Without the overshoot, the system is able to perform the tracking with less chance of entering the unstable state. The undershoot characteristic is thus, preferable to the overshoot in terms of stability.

In order to numerically measure the accuracy of the saccade function, the experiment was also performed as follows:

- For each trial, the target was randomly located away from the center of the image. Figure 65 shows samples of random target position error generated for this experiment.
- Only one saccade command was then issued for moving the camera to ¯xate on the target.

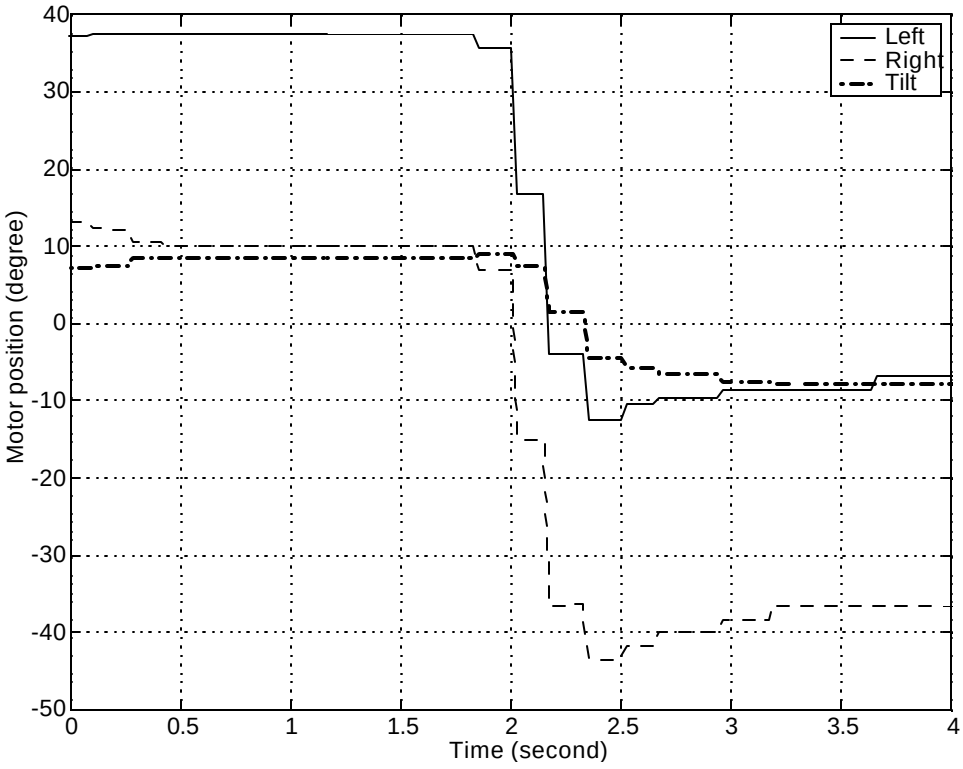


Figure 64: Saccade: motor positions during one period of saccade motion

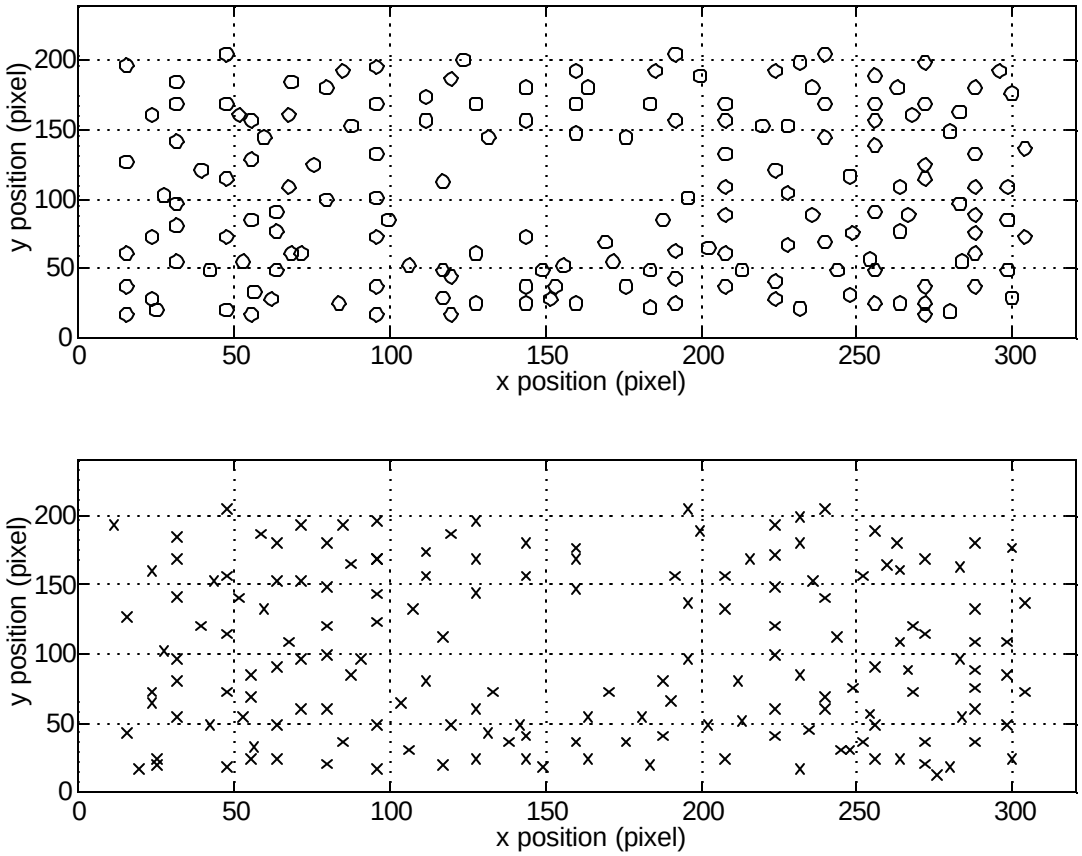


Figure 65: Samples of left and right target position before saccade

[2] After the saccade command was ¯red and the camera motors had stopped moving, the target position was then recorded and the position error computed.

This experiment allows the accuracy of the saccade function to be evaluated ef¯ciently. Table 6 shows the average target position error of 168 trials, covering the entire image. From the results, the saccade system yields a very accurate movement of the motors for less than 3% average (9.6 and 7.2 pixels in x- and y-directions) of the target position error after the saccade movement.

Table 6: Saccade position error percentage (where ¾ is a standard deviation)

	Left Camera		Right Camera	
	x-position	y-position	x-position	y-position
Position error average (%)	2.17	2.17	2.45	2.81
¾ (%)	1.94	2.55	3.71	2.42
Position error average (pixels)	6.94	5.21	7.84	6.74
¾ (pixels)	6.21	6.12	11.87	5.81

Smooth Pursuit

Smooth pursuit maintains the target on the fovea. In this experiment, the target was moved around in the workspace manually at moderate speeds in smoother motion (slower and continuous trajectory) than those in the saccade experiments (quicker and discrete trajectory). By turning on the smooth pursuit module only (no saccade), the target is expected to remain inside the dead zone during the camera head tracking. The dead zone is de¯ned as the circular area in which, if the target position is located inside this zone, it is said to be on the fovea (i.e. if the target is inside the dead zone, no camera adjustment will be made). Figures 66 and 67 show the target position error during the 50-second period of smooth pursuit motion. The corresponding positions of the motors is shown in Figure 68. In this case, the dead zone is 5% of the image size, which is equal to 16 pixels and 12 pixels on the x- and y-axis, respectively. The graph plots demonstrate that the target remains on the fovea for most of the time. In addition, table 7 shows the measure of time in which the target stays on the fovea for average of 10 trials during the smooth pursuit tracking experimental session. Each trial has the average of 60-second interval in which the data sampling rate is approximately 30 samples/second. The table also shows di®erent con¯gurations for both speed of the target and size of the dead zone. From the results, the smooth pursuit shows to maintain the target on the fovea for most of the time at

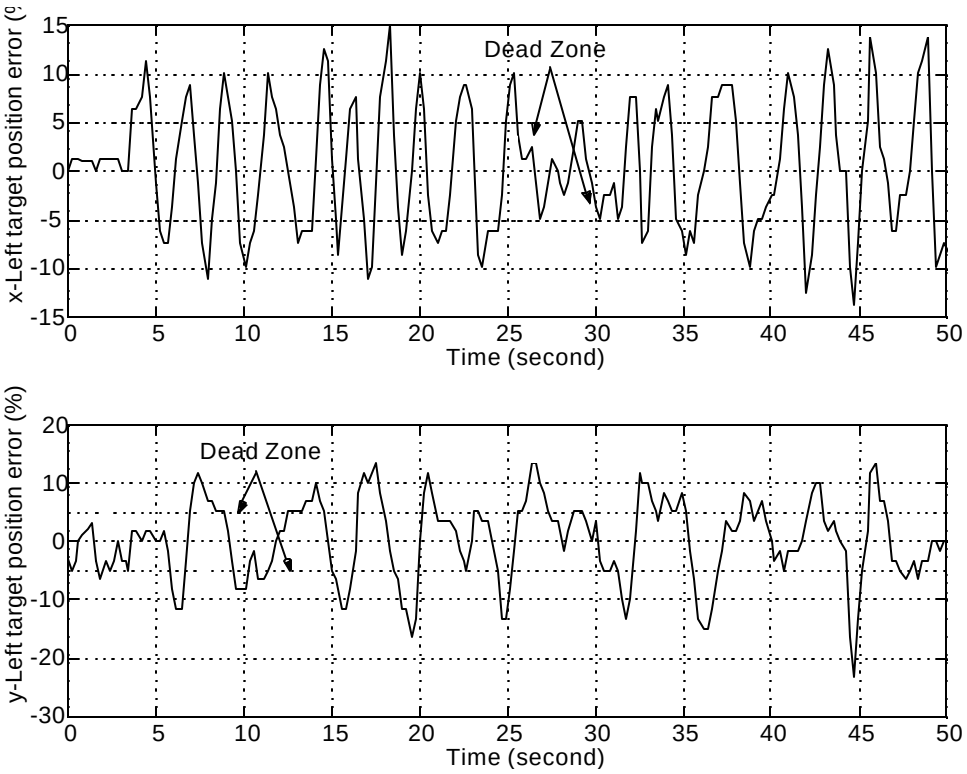


Figure 66: Smooth pursuit: left target position error during smooth pursuit motion

the with moderate target speed (average $30^{\pm}$=s and $10^{\pm}$=s for verge and tilt motors, respectively). The performance of the smooth pursuit signi¯cantly decreases as the target moves at higher speed due to the limitation of the motors' speed. Obviously, by increasing the size of the dead zone, the smooth pursuit performs considerably better.

Figures 69 and 70 show the target position error (distance from image center) and corresponding positions of the motors for a short interval (3 seconds).

In practical, the system uses saccade and smooth pursuit separately because they work in di®erent zone. The system activates the smooth pursuit if the target location is inside the fovea. If the target moves much faster than the camera head can follow,

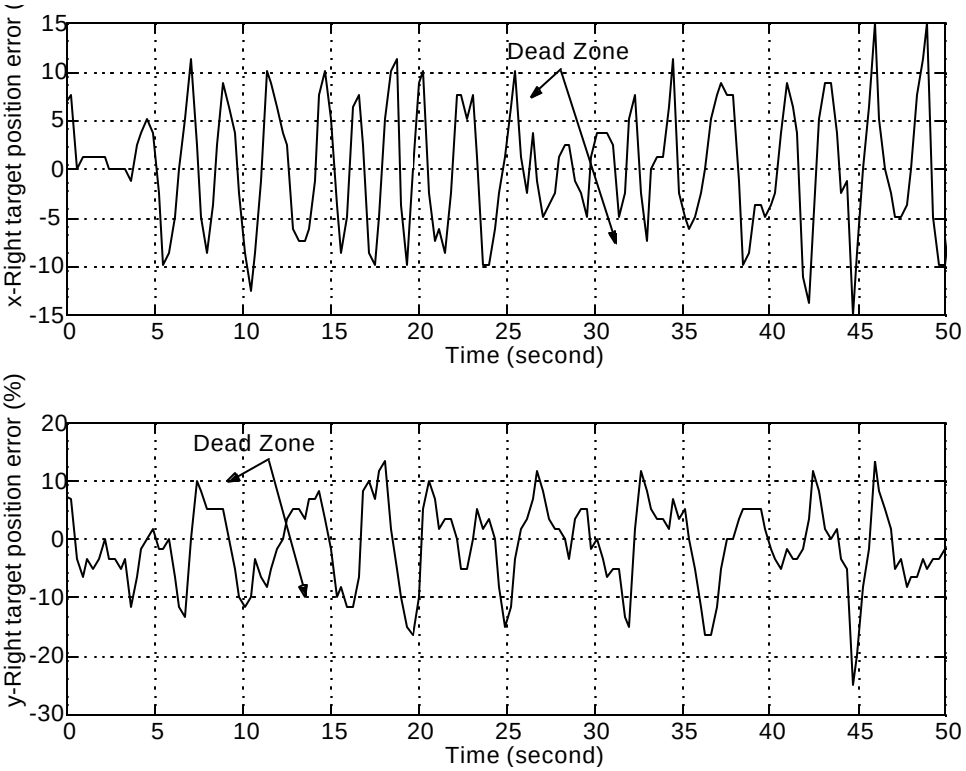


Figure 67: Smooth pursuit: right target position error during smooth pursuit motion

Table 7: Total amount of time on which the target remains on the fovea during the smooth pursuit tracking (where ¾ is a standard deviation).

	Left Camera		Right Camera	
	x-direction	y-direction	x-direction	y-direction
Moderate speed				
within 5% dead zone (%)	58.97	63.57	64.85	69.48
¾ (%)	7.35	5.62	7.11	6.20
within 10% dead zone (%)	95.29	90.06	93.88	91.15
¾ (%)	5.22	4.69	5.01	4.33
Higher speed				
within 5% dead zone (%)	50.24	52.78	47.55	59.56
¾ (%)	12.47	10.21	11.98	11.14
within 10% dead zone (%)	78.87	75.63	80.23	83.71
¾ (%)	9.89	8.55	8.67	8.19

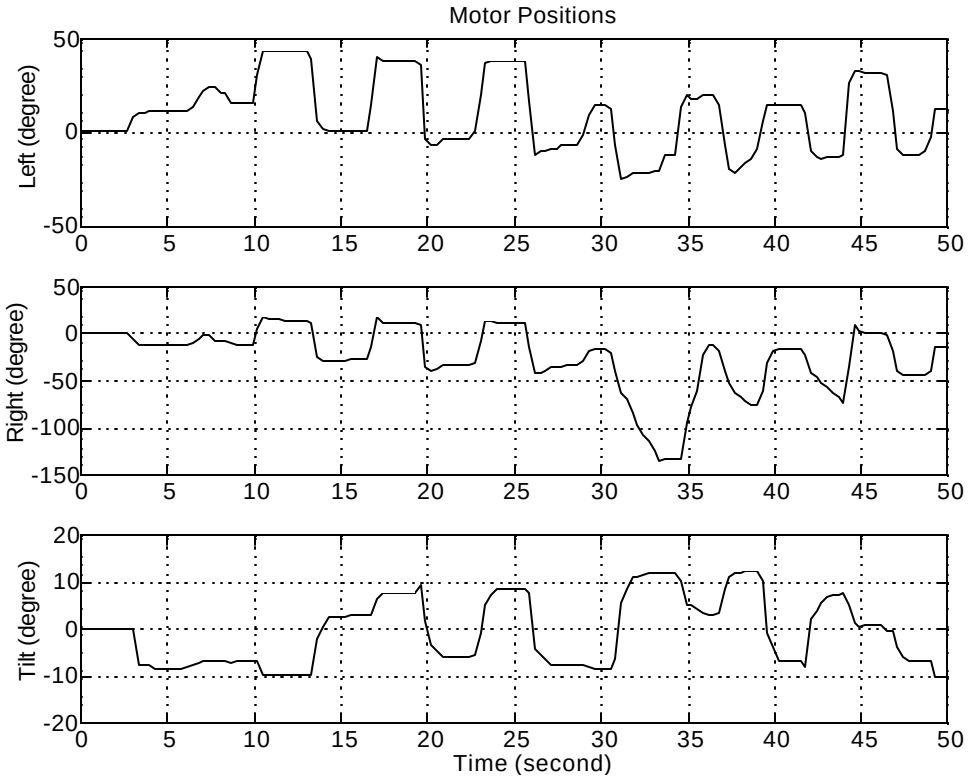


Figure 68: Smooth pursuit: motor positions during smooth pursuit motion

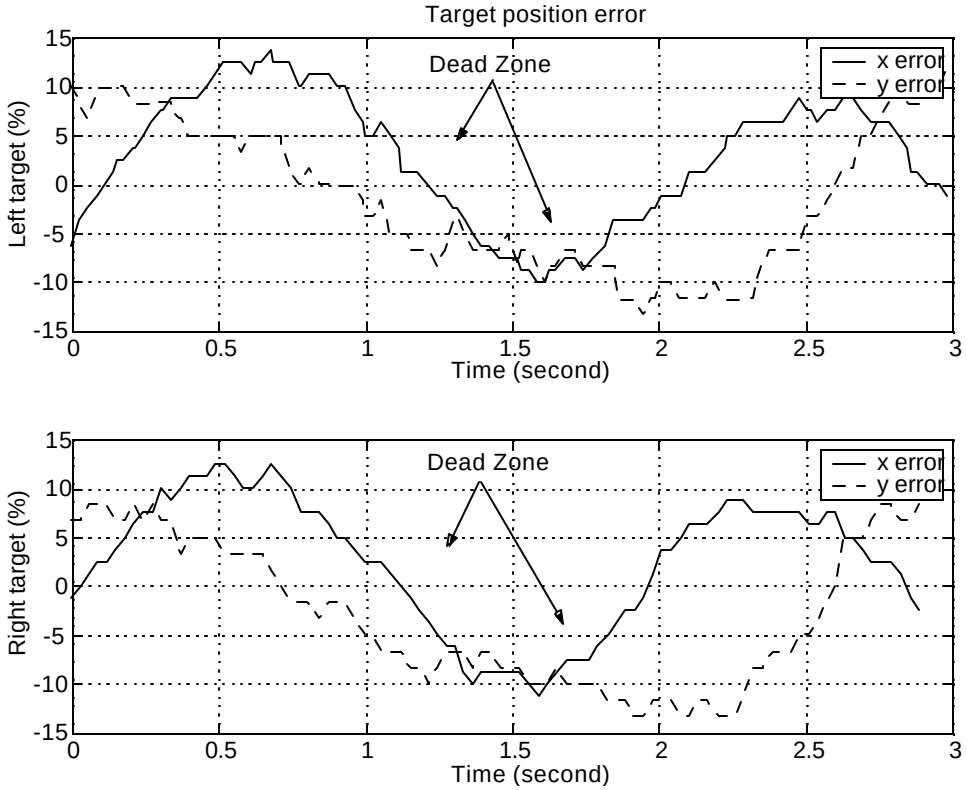


Figure 69: Smooth pursuit: target position error during smooth pursuit motion (short)

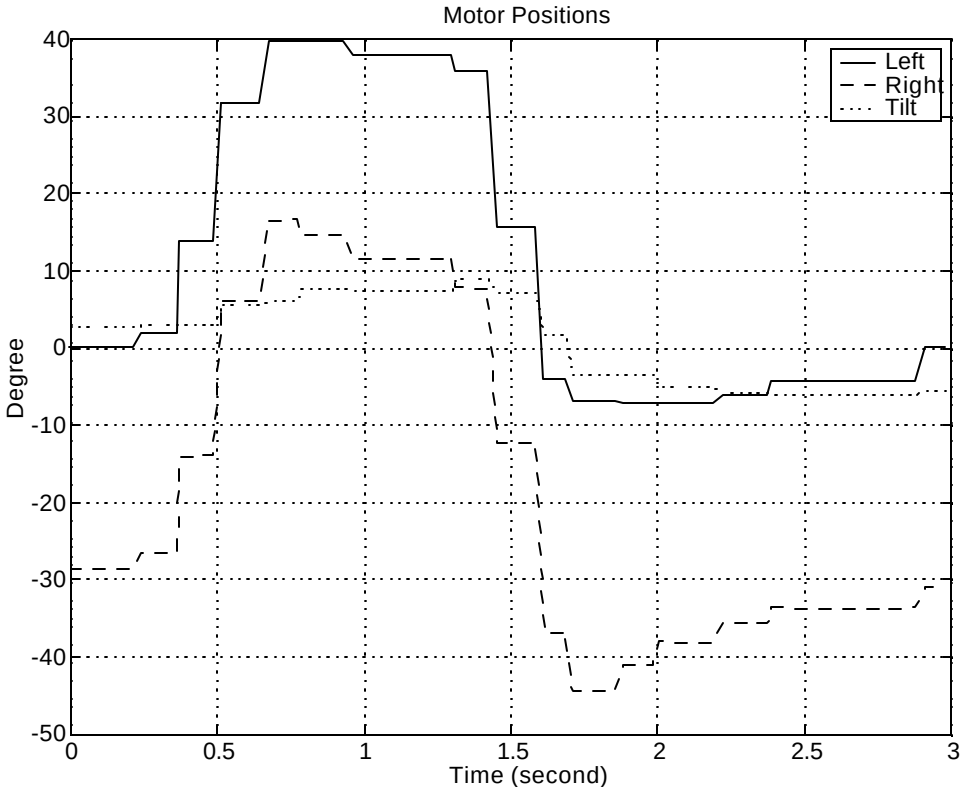


Figure 70: Smooth pursuit: motor positions during smooth pursuit motion (short)

i.e. the target quickly moves away from the center of the image, the saccade will be activated to quickly ¯xate the camera on the target. When the target is brought onto the fovea by the saccade, the system then turns on the smooth pursuit to continuously track the target.

Non-Tracking Behavior

Vergence Disparity Estimate

Vergence ensures that both left and right cameras ¯xate on the same object. It uses a disparity metric to measure similarity between the object in the left and right images (see Chapter III). The experiment was designed to observe the relationship between the disparity and the motion of the cameras. Figure 71 shows a disparity estimate during the movement of the left motor while the right camera is stationary. The result shows a linear relationship between the disparity estimate and the motion of the camera (disparity is zero when both cameras ¯xate on the same target). In practice, the corresponding disparity estimate is used to adjust both left and right cameras so that the disparity is kept at a minimum (see Figure 72). This allows the AVS to perform the vergence motion e±ciently. In this experiment, disparity cues were measured during a 60-second interval of active smooth pursuit tracking. The vergence system was set to keep the disparity under the threshold of 5 pixels. Table 8 shows the average of disparity estimate during the vergence experimental session. The experiments has been performed for 10 trials.

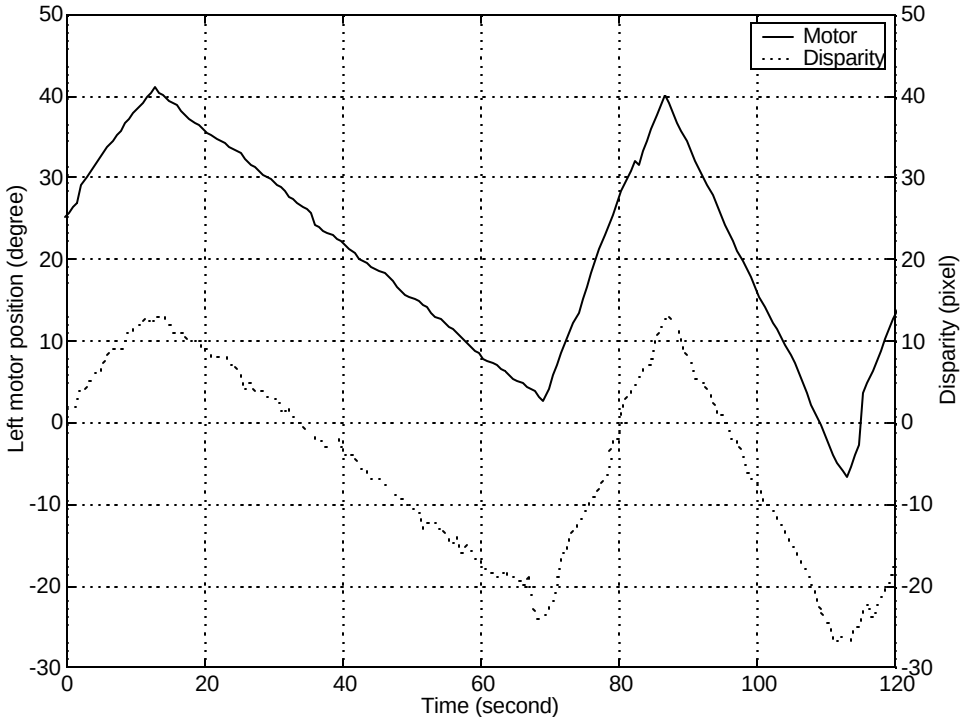


Figure 71: Vergence: disparity estimate during left motor motion (right motor is not moving)

Table 8: Disparity measurement during active smooth pursuit tracking.

Average disparity estimate (pixels)	3.00
¾ (pixels)	3.62

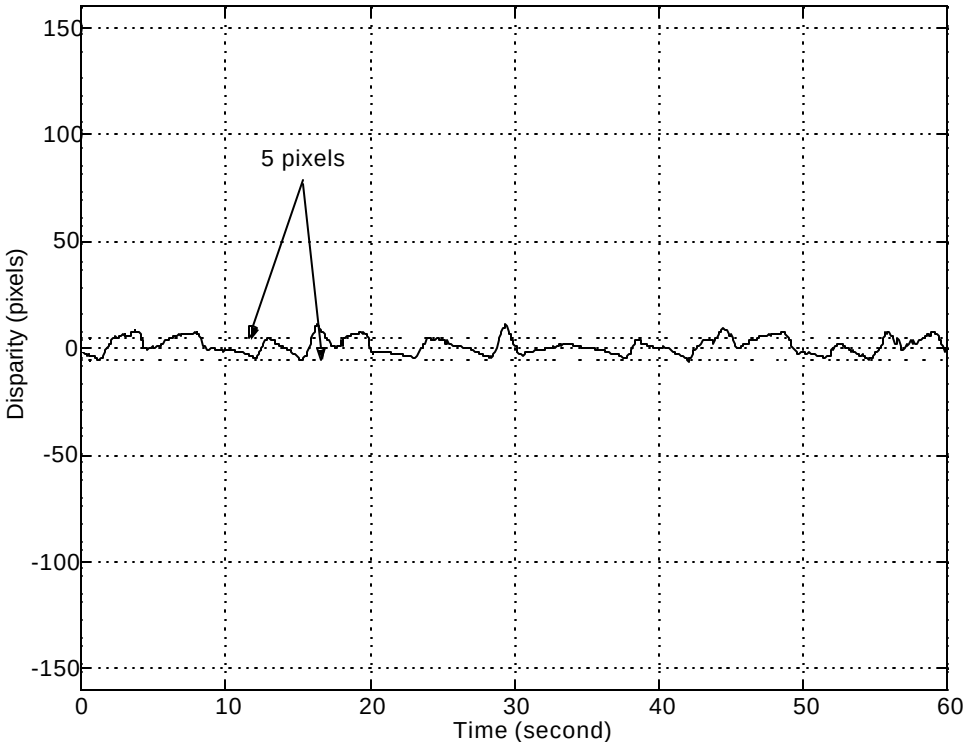


Figure 72: Disparity measurement during active smooth pursuit tracking

Eyes Stabilization

Vestibulo-ocular re°ex and opto-kinetic re°ex together are categorized as the eye stabilization system. The stabilization of the eyes involves compensation for the motion of the left and right motors against the pan motor movement. The activation of the eye stabilization behaviors depends on the angle between the target and the pan motor axis. Once the angle is above a certain threshold, the camera head then adjusts the pan motor to move toward the target. Essentially, the eye stabilization module will try to keep this angle at zero (i.e. the pan motor always looks toward the target). In this experiment, the AVS turned on the VOR and OKR modules. The target was moved back and forth, from left to right, and so on.

Figures 73, 74, and 75 show, respectively, the left target position error, the right target position error, and the motor positions during a 50-second period of target tracking with eye stabilization. Note that the eye stabilization a®ects only the verge motors. Consequently, the y-target position error is not signi¯cantly important here.

Figures 76 and 77 provide a closer look at the target position error and motor trajectory. The graph plots show one period of the target moving from left to right and then left again. This behavior is described as follows:

- The target is located inside the dead zone from time t = 0 ¡ 1:375 seconds and starts moving away from the center of the image toward the right side of the camera head from time t = 1:375 ¡ 1:7 seconds.
- At time t = 1:7 seconds, the system ¯res saccade commands and quickly moves the left and right cameras toward the target (as the target position error goes back down to the dead zone).

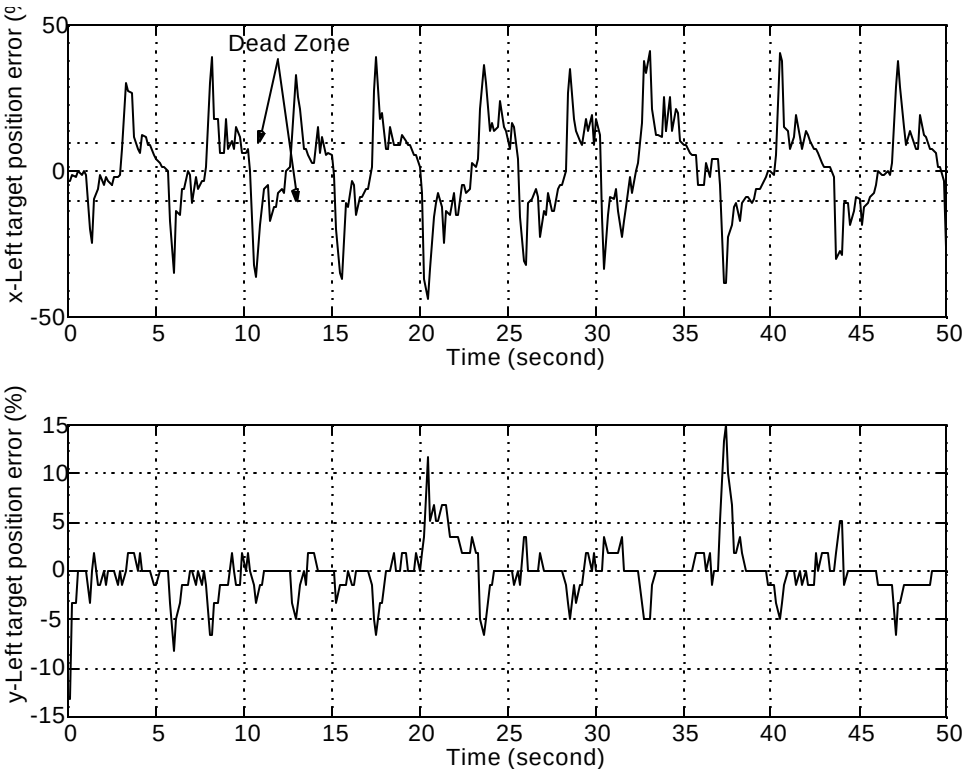


Figure 73: Eyes stabilization: left target position error during eyes stabilization

- Because the saccade commands have moved the verge motors, the current position of the verge motors is now above the threshold and causes the eye stabilization to activate from t = 2:125 ¡ 3:375 seconds. During the eye stabilization, one can observe that the AVS has maintained the target position error at the border of the dead zone. This action can also be observed in Figure 77, where the pan motor starts to change the position in the opposite direction to the verge motors (t = 1:2123 ¡ 3:375 sec). Note that there is a small bump at approximately t = 2:5 seconds, caused by the sudden motion of the pan motor.

- The second half-period of this same motion is shown from time t = 4 ¡ 8 seconds, plotted in the opposite direction, where the target is moving back from left to right of the camera head.

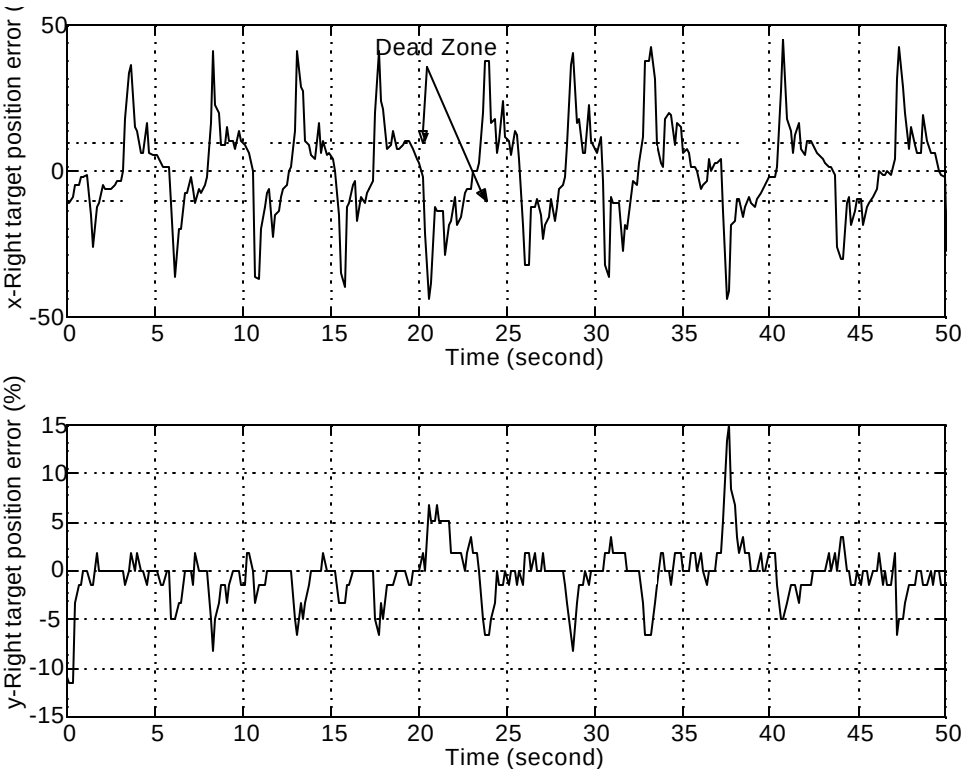


Figure 74: Eyes stabilization: right target position error during eyes stabilization

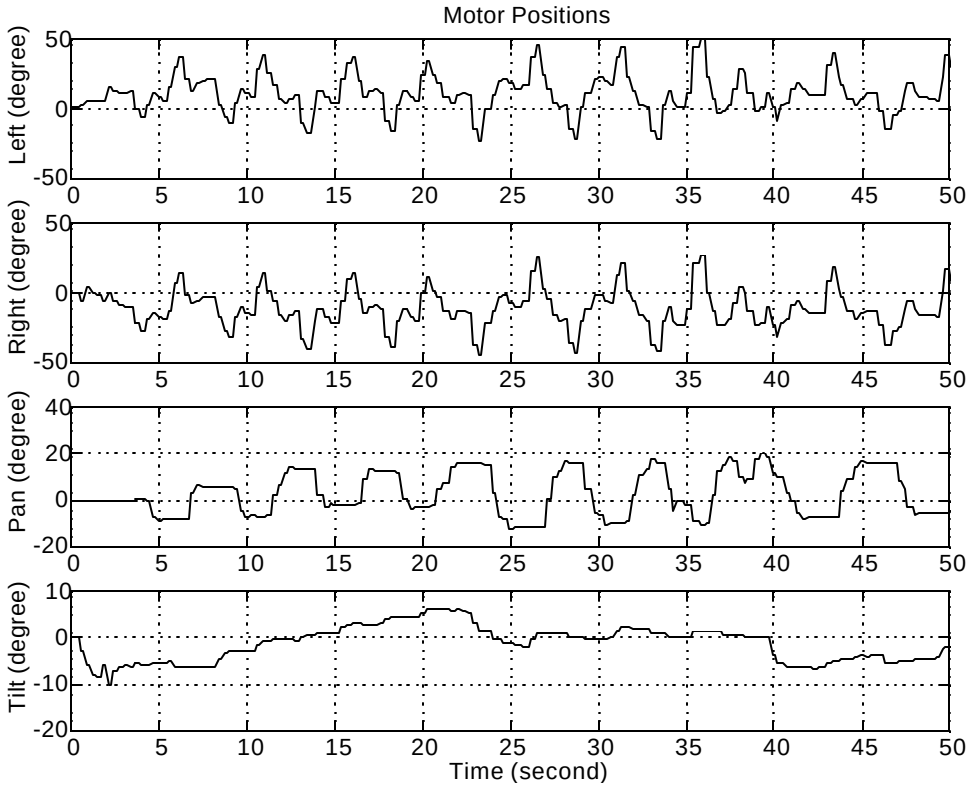


Figure 75: Eyes stabilization: motor positions during eyes stabilization

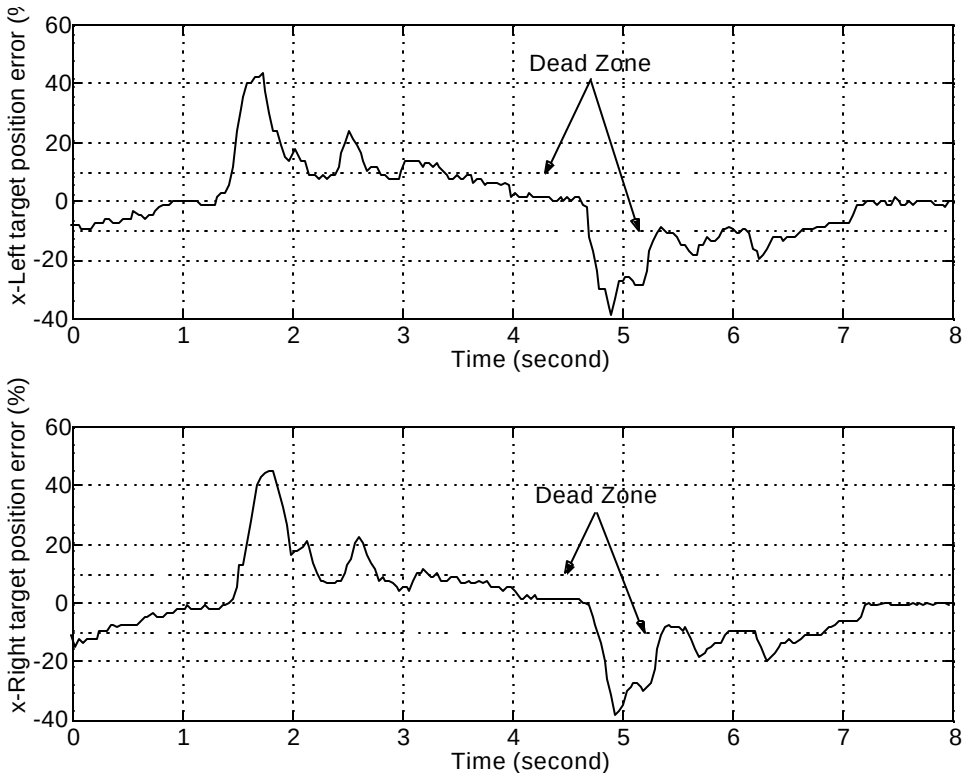


Figure 76: Eyes stabilization: x-target position error during eyes stabilization (short)

Table 9 shows the measure of time in which the target stays on the fovea for average of 10 trials during the eye stabilization experimental session. Each trial has the average of 60-second interval in which the data sampling rate is approximately 30 samples/second. From the results, the target remains on the fovea merely half of the total time. This is because there are saccade motions (as described in the previous section) which bring the target away from the center for most of the time during the eye stabilization experiments. After subtract the amount of time in which the camera performs saccade motions, the result show that the target remains on the fovea for almost 75% of the time.

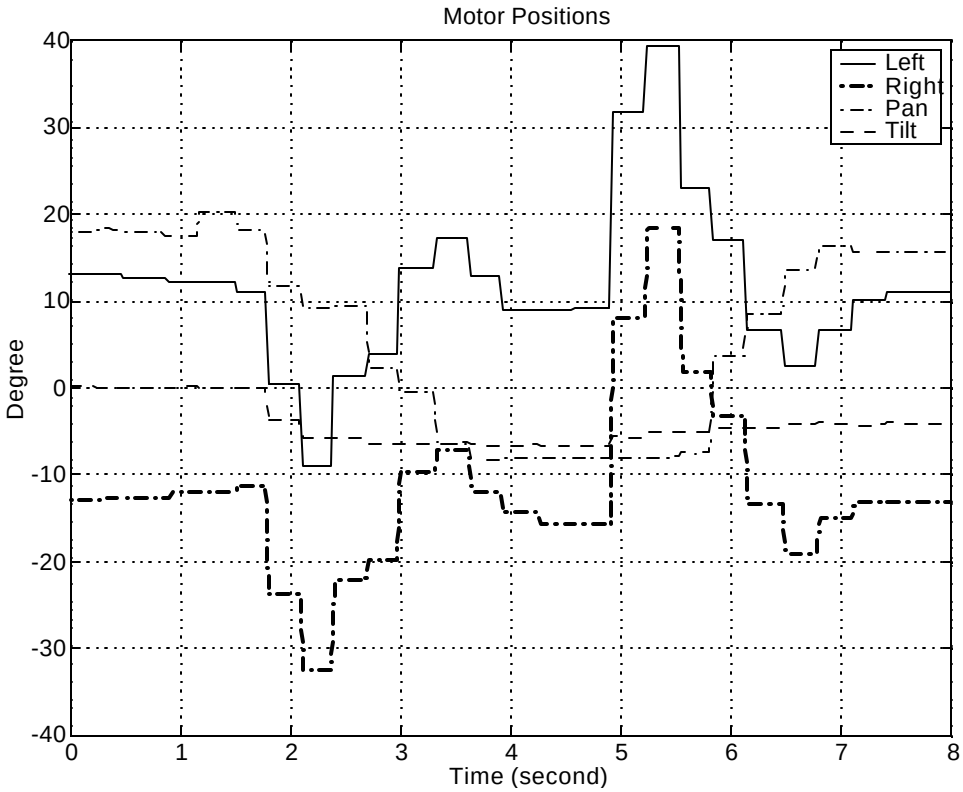


Figure 77: Eyes stabilization: motor positions during eyes stabilization (short)

Table 9: Total amount of time on which the target remains on the fovea during the eye stabilization (where ¾ is a standard deviation).

	Left camera x-direction	Right camera x-direction
Total time (%)	54.48	56.48
¾ (%)	5.36	4.11
Total time after subtract saccade motion (%)	74.08	73.07
¾ (%)	5.10	4.23

Robustness and Performance

The evaluation of the tracking performance of an active vision system has never been a simple and straightforward task, especially when it comes to a comparison with other binocular platforms: this is almost impossible because of the lack of common benchmarks in this particular ¯eld. Obviously, di®erent work conditions cannot be easily reproduced from one laboratory to another. Consequently, for this study, the tracking performance has been evaluated such that the system shows its ability to track properly; comparisons of this performance with that of previously used methods on the same system are discussed.

There are many factors that need to be taken into the account when evaluating the robustness of the AVS, with control and visual processing the two main components. However, visual processing is not the focus of this dissertation. As mentioned earlier in this chapter, in order to possess the characteristics of the control system, reliable vision processing is employed. The visual attention network (see Chapter III)uses color segmentation. Using a unique color model ensures that the target is properly detected and located most of the time, regardless of the workspace environment. Figures 78, 79, and 80 show the results of active tracking. The target is moved around in the workspace at varying speeds. For a 60-second period, the target remains at the center of the image (inside the dead zone).

Table 10 shows the measure of time in which the target stays on the fovea for average of 10 trials during the active tracking experimental session (all eye motion controls are activated). Each trial has the average of 60-second interval in which the data sampling rate is approximately 30 samples/second. The result show that

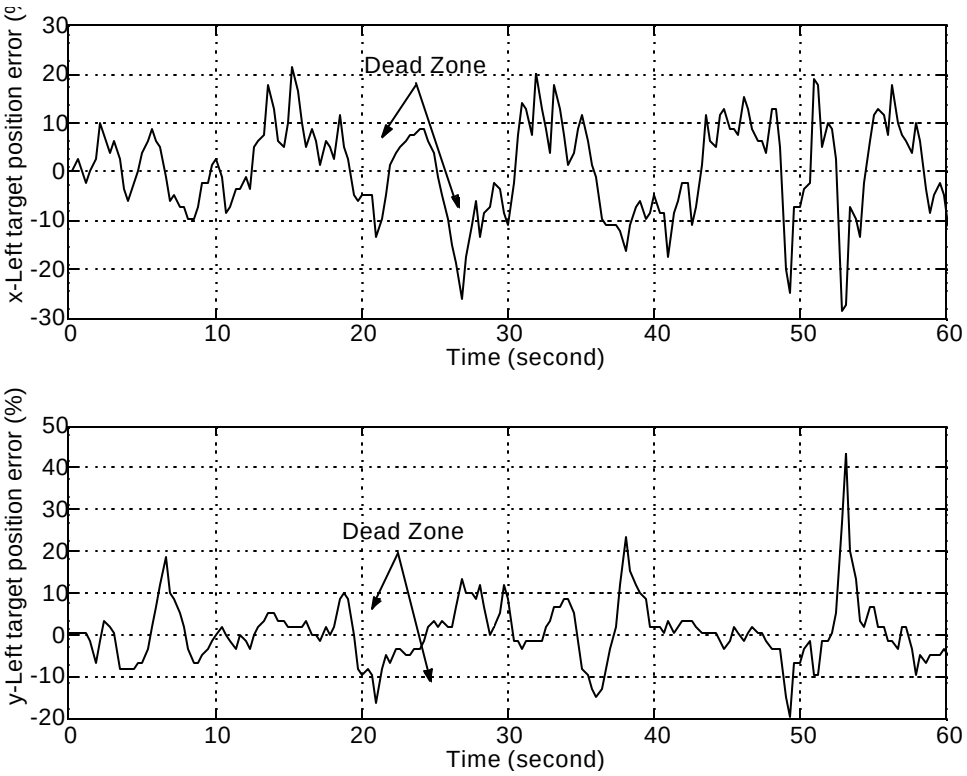


Figure 78: Overall tracking: left target position error during active tracking

Table 10: Total amount of time on which the target remains on the fovea during the active tracking (where ¾ is a standard deviation).

	Left camera		Right camera	
	x-direction	y-direction	x-direction	y-direction
Total time (%)	77.66	88.88	75.61	87.50
¾ (%)	5.44	3.27	5.12	4.40
With speed above average (%)	68.47	70.21	69.33	72.17
¾ (%)	6.38	5.89	6.18	5.78

the target remains on the fovea for more than 75% of the time. The average speed of the camera tracking is 30±=s; 20±=s; and 12±=s; for verge, pan, and tilt motors, respectively. Due to the limitation of the motors, by increasing the target speed, the performance of the system decreases as can be seen from Table 10.

One major problem of the AVS is the stability of tracking. Often the control

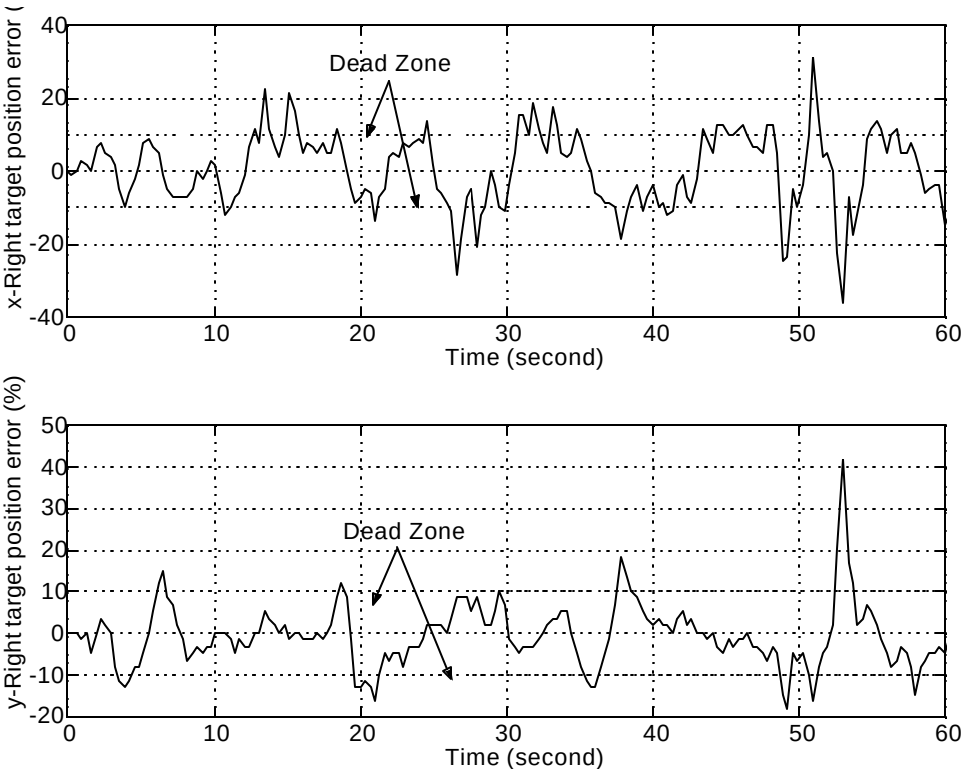


Figure 79: Overall tracking: right target position error during active tracking

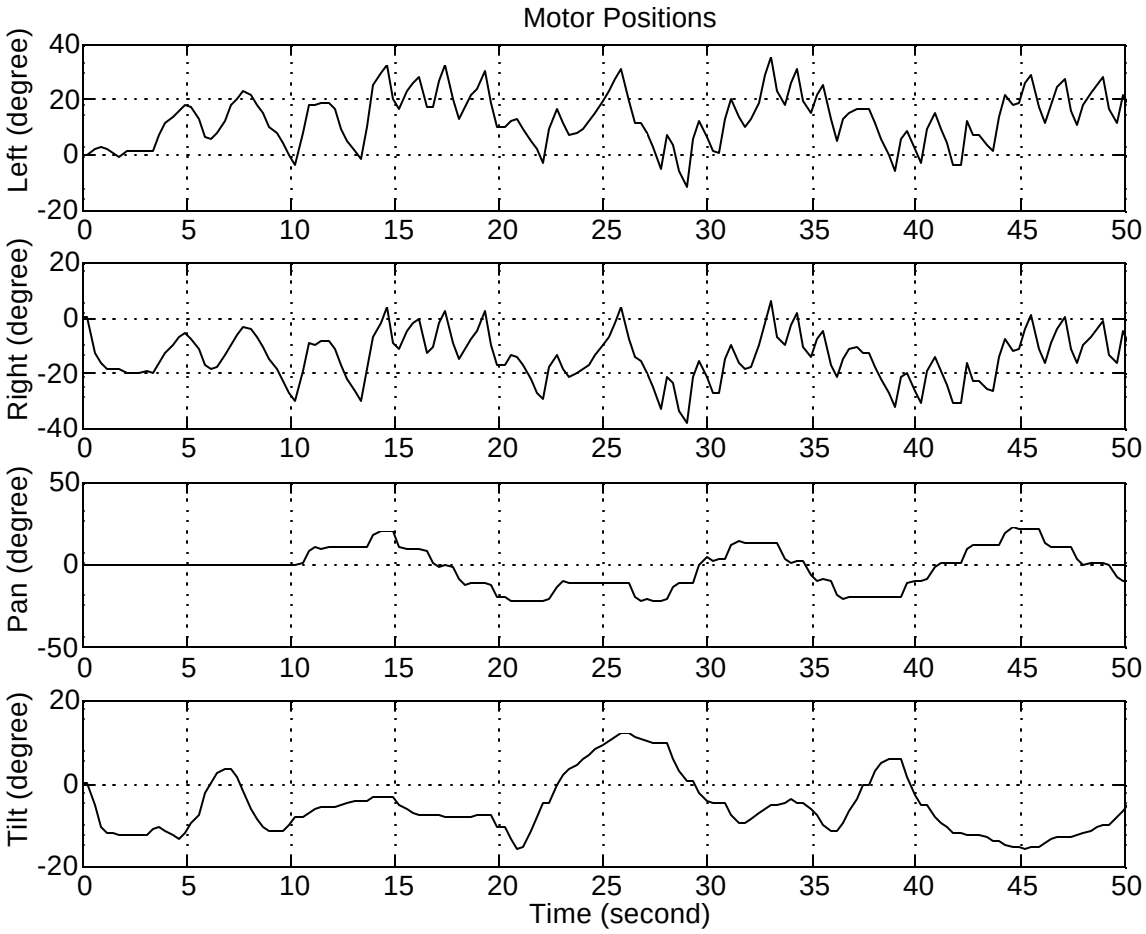


Figure 80: Overall tracking: motor positions during active tracking

system, which relies mainly on the visual processing unit, can become unstable, resulting in loss of the target being tracked. In this system, the smoothing filter is used to reduce the possibility of this happening (see Chapter IV). In Figures 81, 82, and 83, the graph plots of the target position error and motor positions are shown in one period of tracking, in which all eye movement modes are activated. Two important behaviors during this motion are saccade and eyes stabilization, which occur at time $t = 1:7 \pm 2:3$ seconds and $t = 2:3 \pm 3:0$ seconds, respectively. The saccade makes a rapid camera movement and brings the target onto the fovea. After this, the eye stabilization turns the camera head toward the target (see Figure 83, where the pan motor is moving). This particular motion can probably cause the camera head to become unstable, as can be seen from the peak at the time $t = 2:5$ seconds in Figure 81 and 82. The sudden motion of the motors at the begining of the eye stabilization causes this peak to occur. By applying the smoothing filter, the AVS provides a better, less jerky movement of the motors (see Figures 84, 85, and 86). The gain of the smoothing filter is 0.35 (see Chapter IV).

Even though the smoothing filter yields a better result in term of smoother motor motions, the system obtains that at the expense of its tracking speed. This can be seen by noting the time it takes for the motors to settle down. Without the smoothing filter, the AVS takes approximately 800 milliseconds for the pan motor to be stabilized. With the smoothing filter, it takes roughly 1500 milliseconds. This settling time for the pan motor to be stabilized affects the overall performance of the AVS in term of tracking speed. Considering when the pan motor moves toward the target, the shorter time it takes, the faster speed it can follow the moving target.

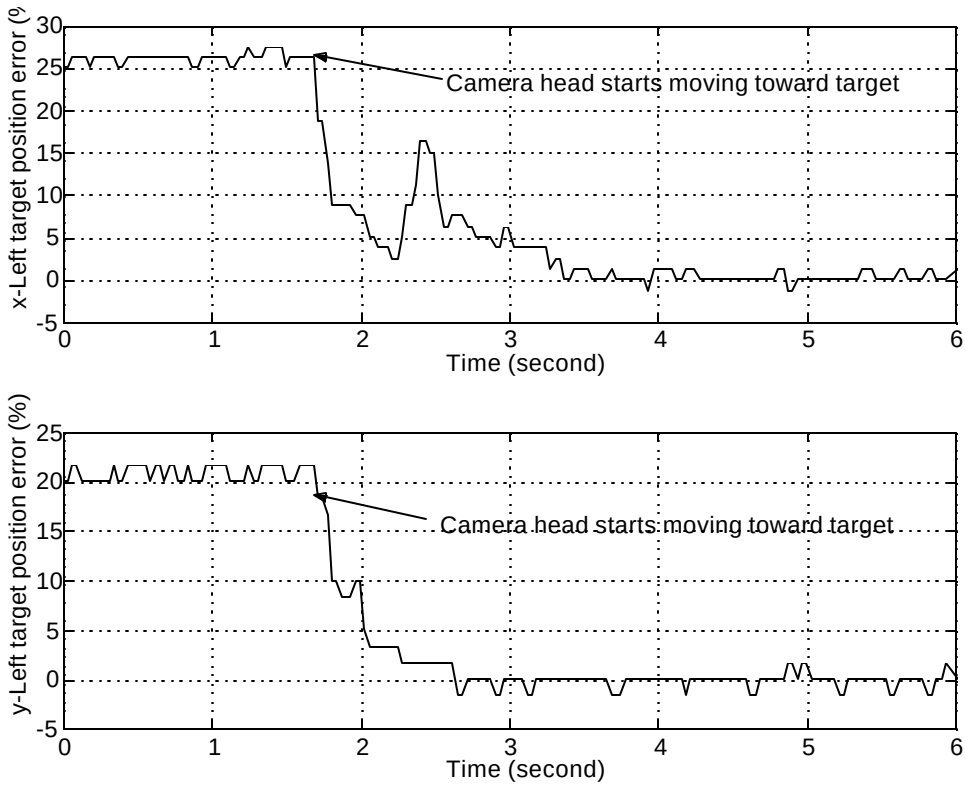


Figure 81: Left target position error (short) without smoothing ¯lter

From Figure 83, the average tracking speed of the verge, pan, and tilt motors can be calculated as 30±=s, 20±=s, and 12±=s, respectively. The speed of these motors is slower with the smoothing ¯lter implemented. In this case (smoothing ¯lter gain = 0.35), the average tracking speed of the verge, pan, and tilt motors are 20±=s, 10±=s, and 4:5±=s, respectively.

In order to evaluate the performance of the proposed AVS, a comparison with the previously used tracking system on ISAC was made. The old system uses a simple proportional control. Each camera moves toward a direction corresponding to the target position error sign. The amount of movement is computed by a multiplication of the target's distance from the center of the image and a constant gain. In other words, the camera makes a smaller move if the target is closer to the fovea. The gain

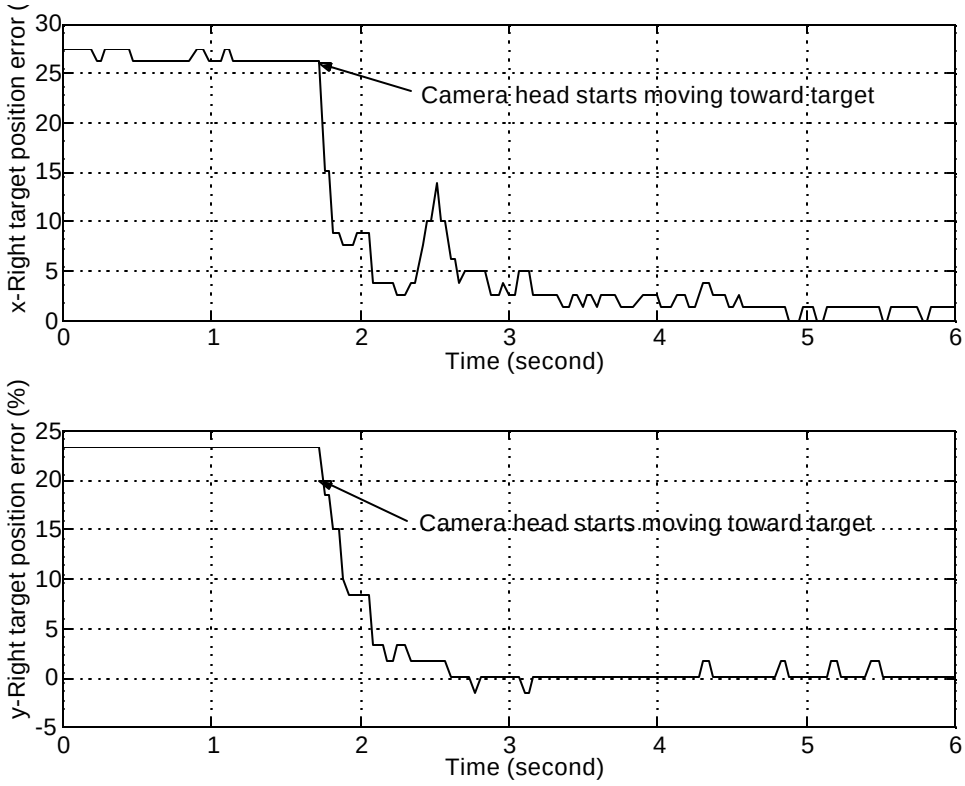


Figure 82: Right target position error (short) without smoothing ¯lter

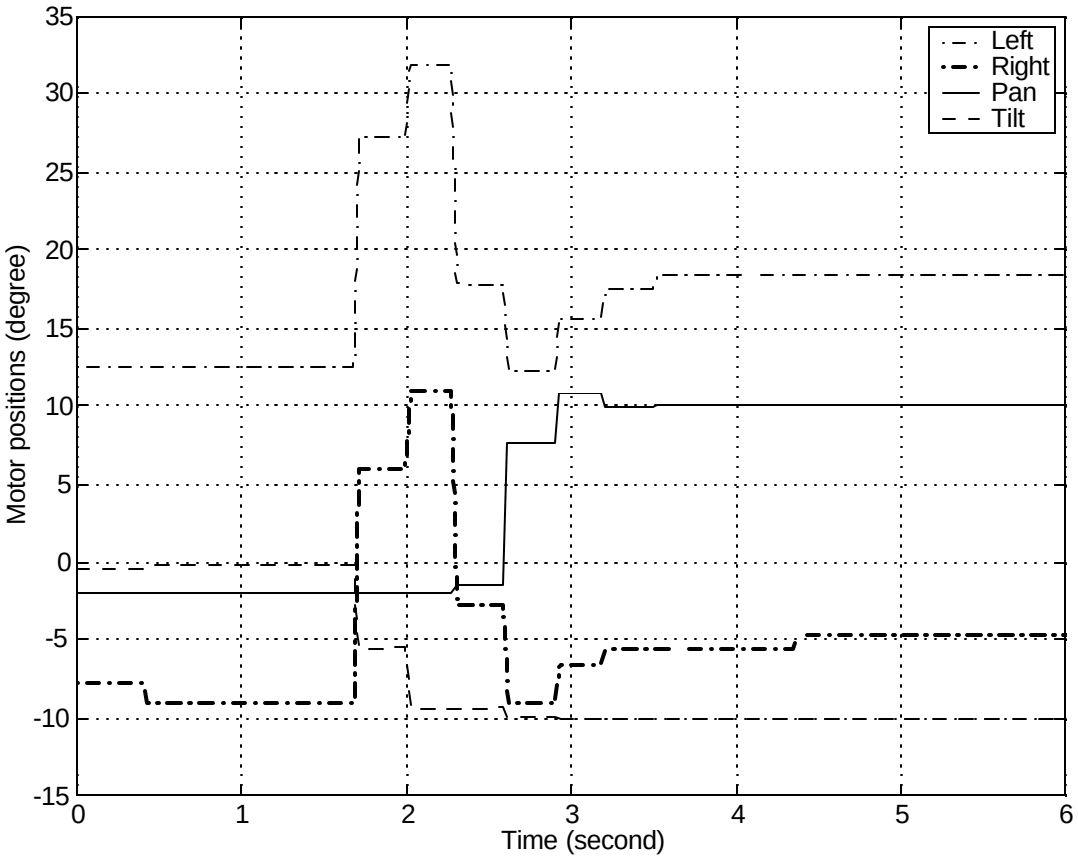


Figure 83: Motor positions (short) without smoothing filter

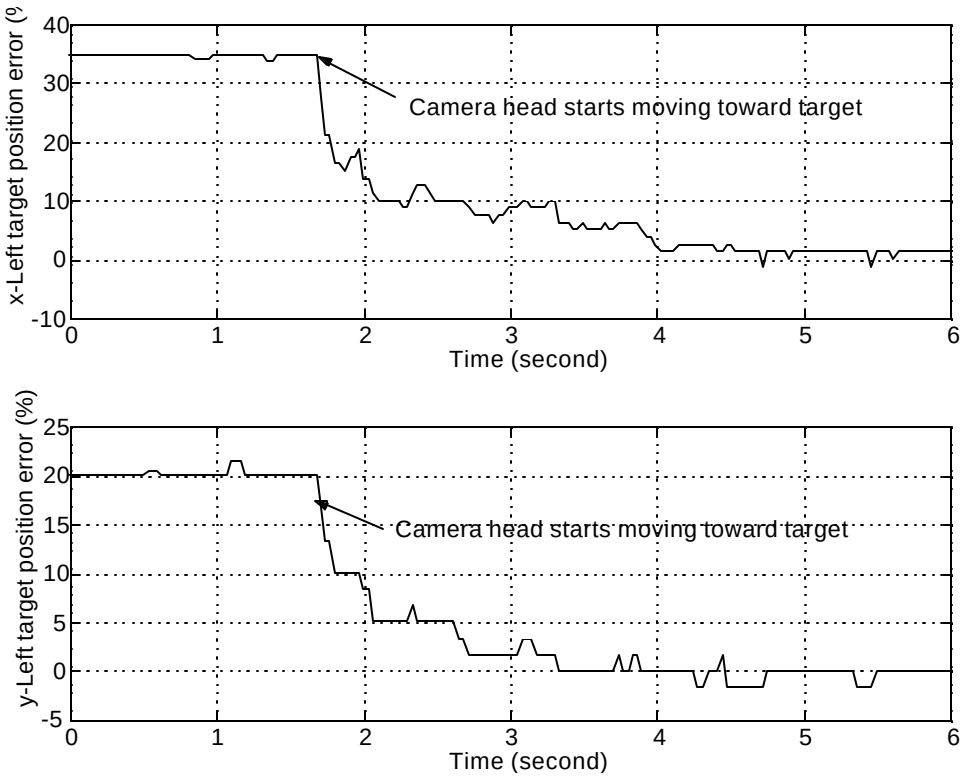


Figure 84: Left target position error with smoother filter

for each axis of the camera allows adjustment for the camera motion.

Figures 87, 88, and 89 show the results from the proportional control with the gain equal to 0.25. The gain is equal to 0.1 for the results in Figures 90, 91, and 92. For the gain of 0.25, the results show that there is an oscillation from the target position caused by the over-adjustment of the verge motors against the pan motor motion (from t = 1:5 ¡ 4 sec). With the 0.1 gain, the proportional controller provides smoother motion (less oscillation). The system, however, takes longer than the higher gain control to stabilize the verge motors and the pan motor movement.

The advantages of the human-like motion control over the proportional control can be summarized as follows:

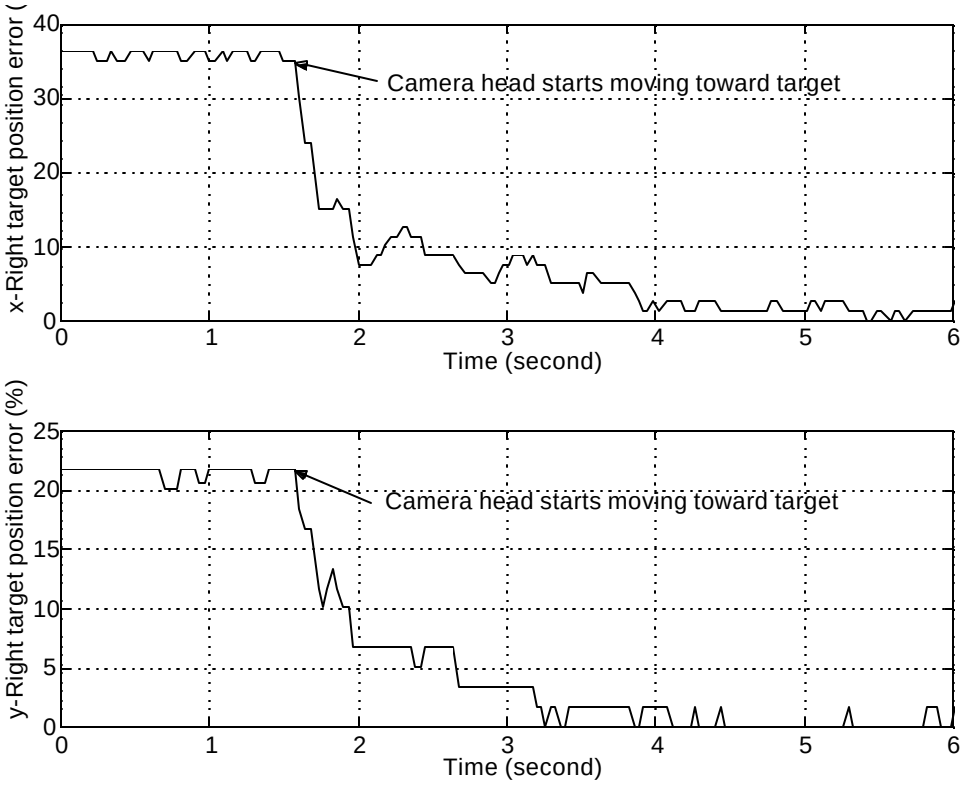


Figure 85: Right target position error with smoother ¯lter

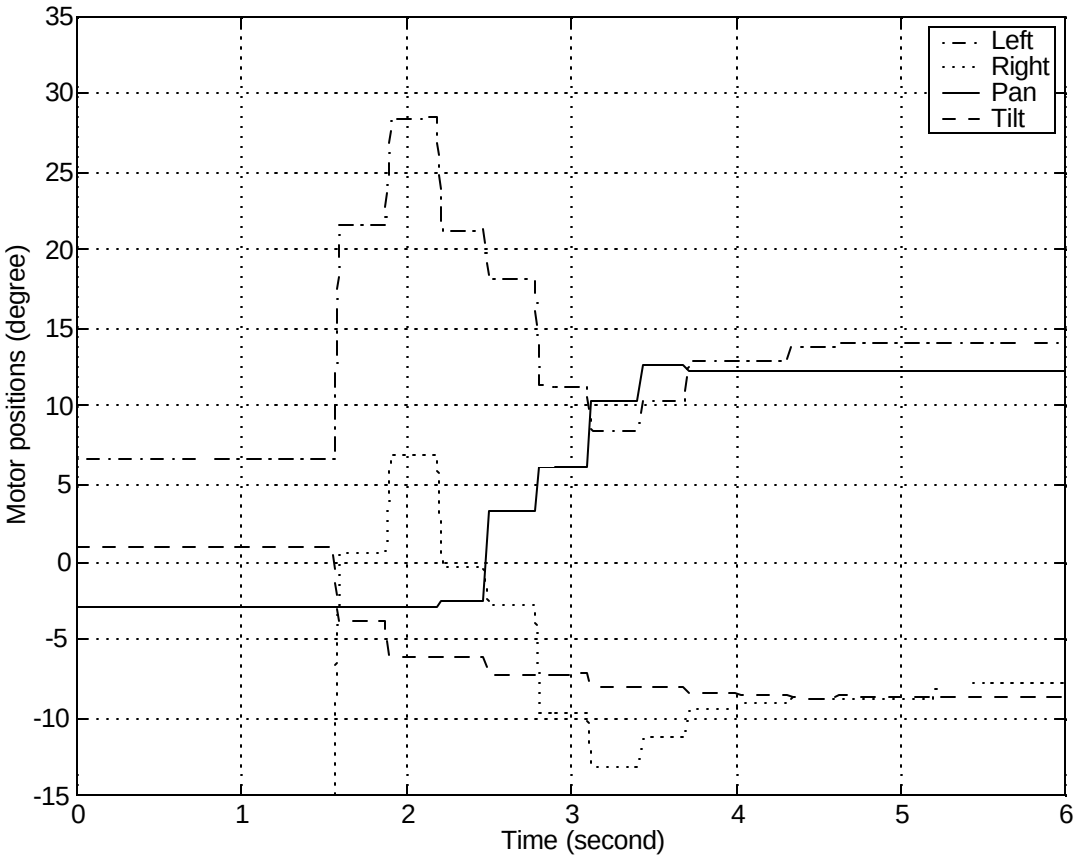


Figure 86: Motor position with smoother filter

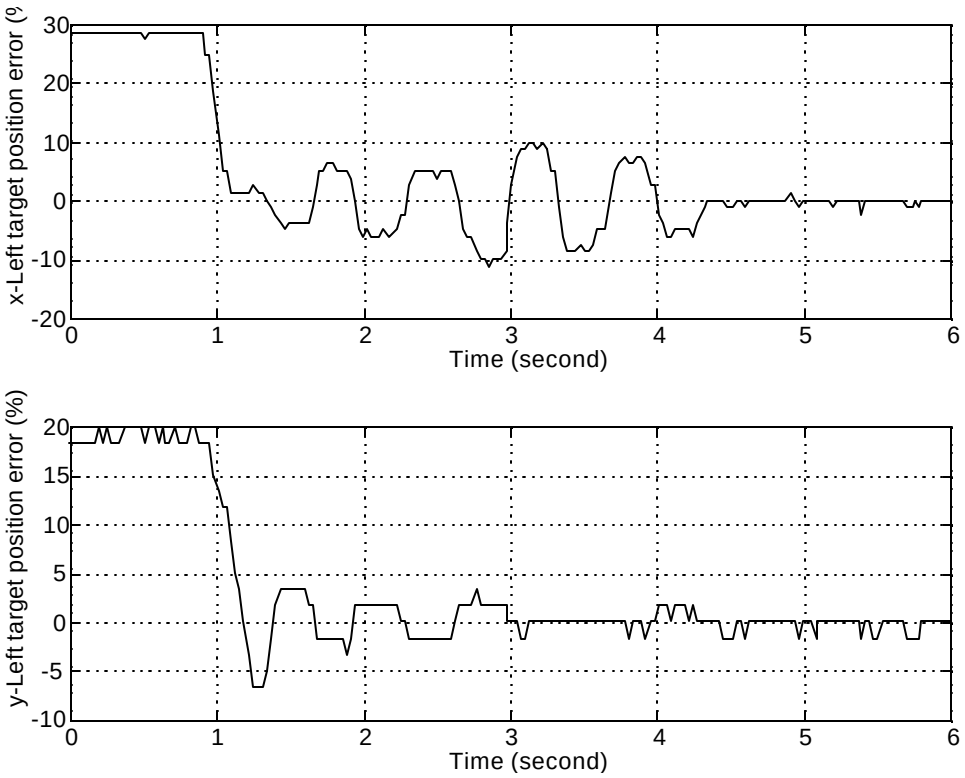


Figure 87: Proportional control with gain of 0.25: left target position error

- Saccades provide more accurate motor commands to quickly move the camera head toward the target. Due to a non-linear relationship between the image coordinates and the motor commands, the constant gain of the proportional controller, which is a linear control, is not as accurate as the saccade control which is controlled by an artificial neural network trained for the specific system. By reducing the gain of the proportional controller, a more accurate control can be obtained, but the overall speed of tracking decreases significantly.

- Eye stabilization yields a more robust and faster motion for compensating for the verge motors against the pan motor movement. The proportional controller relies on the adjustment of each motor within the restriction of the gain control. This yields a slower speed in order to stabilize the motors. However, if the gain

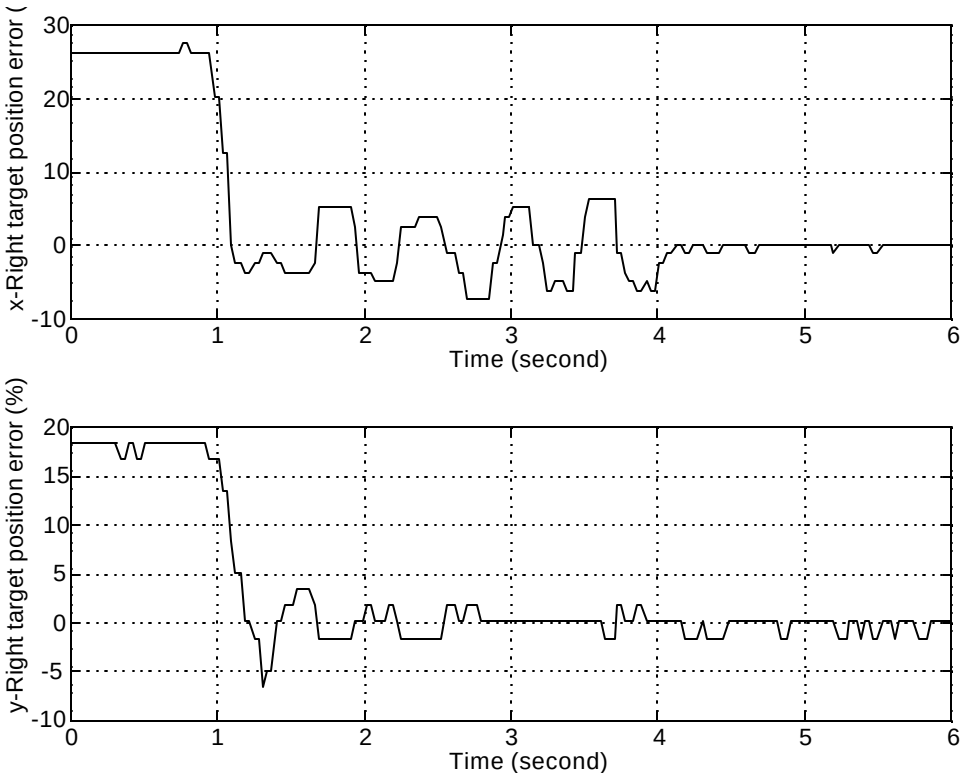


Figure 88: Proportional control with gain of 0.25: Right target position error

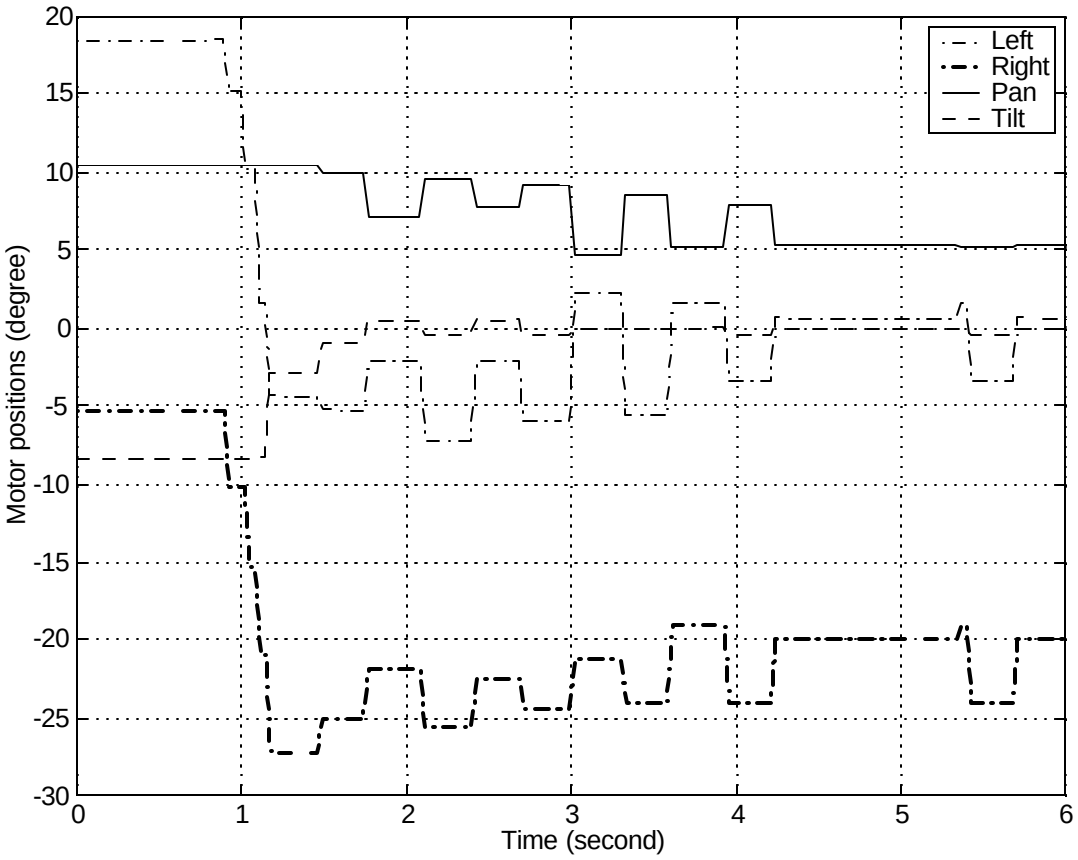


Figure 89: Proportional control with gain of 0.25: motor position

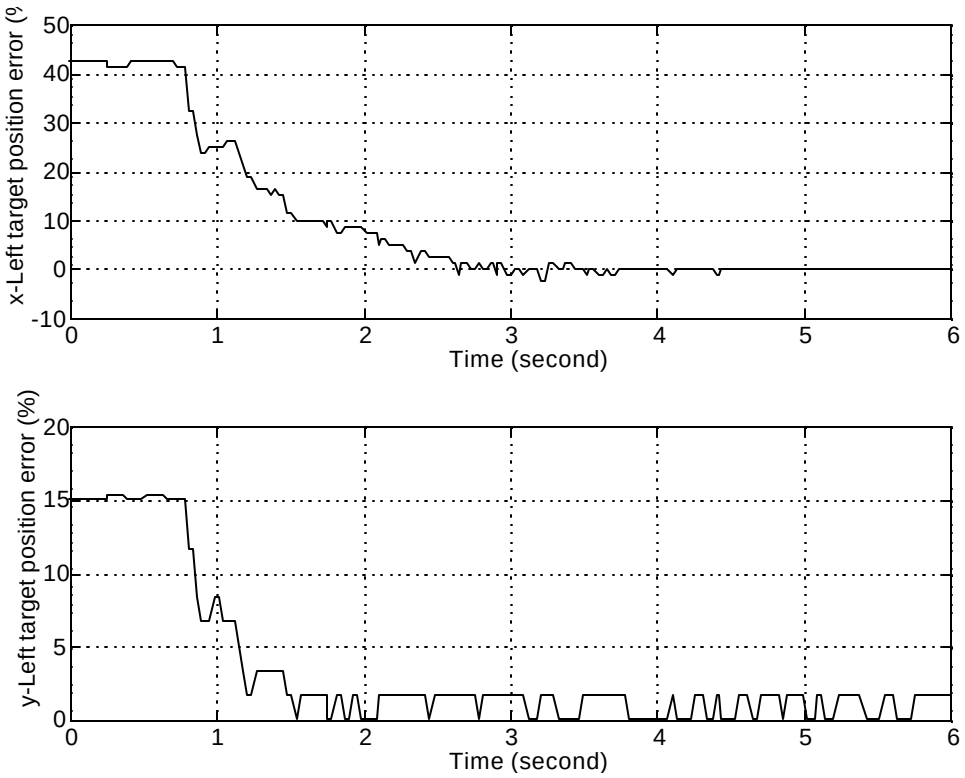


Figure 90: Proportional control with gain of 0.1: left target position error

control is increased, the system is more likely to become unstable (unable to stop oscillating).

As mentioned earlier, it is almost impossible to compare di®erent active vision systems because of the lack of common benchmarks. There are, however, some characteristics of systems that can be compared. Two wellknown systems have been chosen as references here: EsCHER [70] and LIRA [71]. Rougeaux made major contributions to active vision systems with the EsCHER head, which includes smooth pursuit, saccades, and vergence control of space-variant, multiresolution cameras. Panerai built arti¯cial gaze stabilization on the LIRA head by combining vestibulo-ocular re°ex and opto-kinetic re°ex with inertial device on standard cameras. Comparing some aspects of these two other systems should aid in evaluating the system of human-like

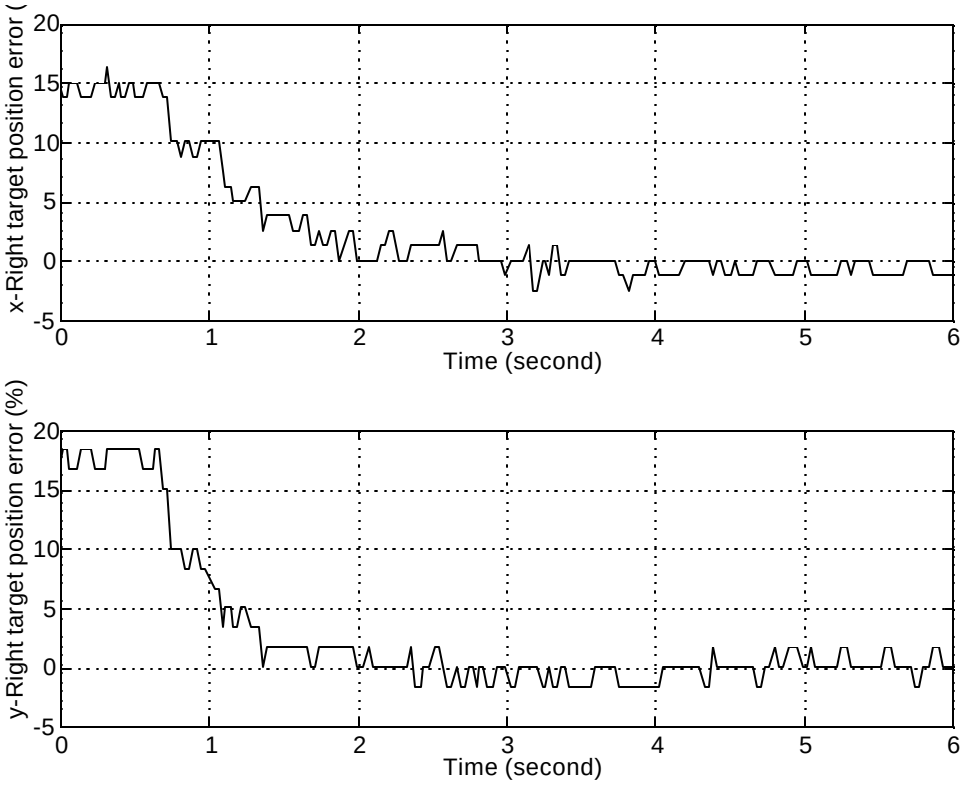


Figure 91: Proportional control with gain of 0.1: right target position error

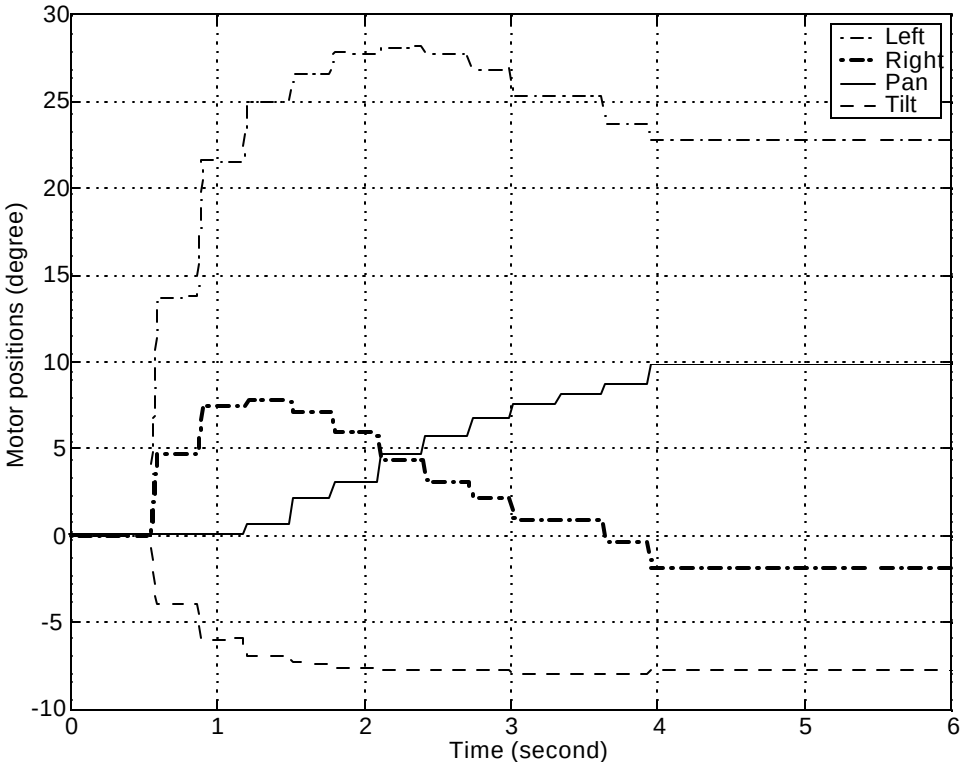


Figure 92: Proportional control with gain of 0.1: motor positions

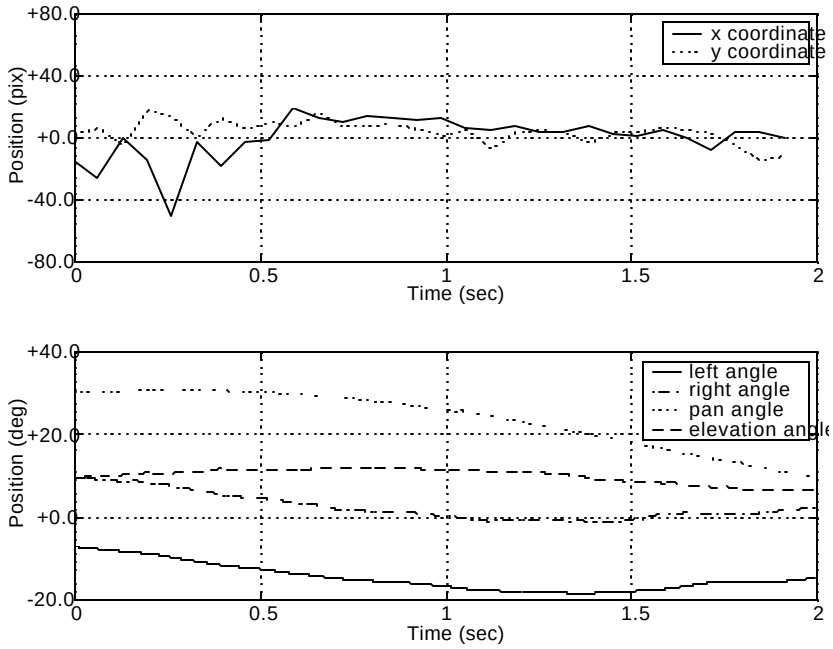


Figure 93: EsCHER smooth pursuit: (top) target right image position x_r, (bottom) joints position μ (reprinted from [70]).

camera head control proposed in this dissertation.

Smooth pursuit, saccades, and vergence were implemented on EsCHER. Figure 93 shows the target position and velocity in the image, as well as the joint positions during a smooth pursuit sequence. Figure 94demonstrates the saccade behavior of EsCHER with the joints position, and the target distance from the center of the image, during the saccade sequence.

Similar to the results presented in this dissertation, EsCHER shows the desired characteristics of both smooth pursuit and saccade behaviors. From EsCHER's results, the smooth pursuit keeps the target at the center of the image, with the target position error less than 20 pixels (compare to 10% dead zone - 16 pixels in x axis and 12 pixels in y axis - from the system proposed in this dissertation), while the saccades quickly ¯xate the camera on the target. During high-speed tracking, EsCHER

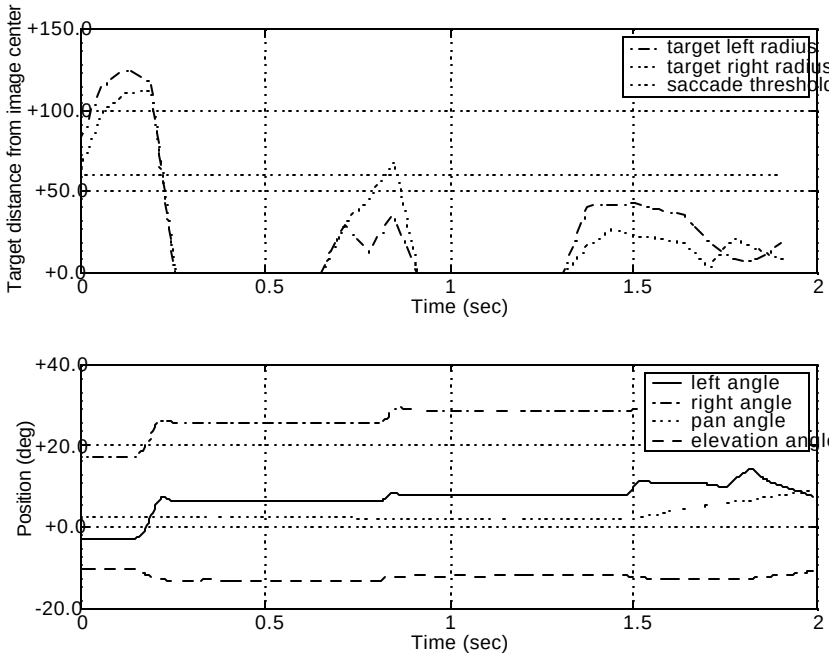


Figure 94: EsCHER saccade: (top) target radius $k\dot{x}_v k$ (bottom) joint position μ (reprinted from [70]).

reportedly has demonstrated the capability of tracking the target with a speed of up to 70±=s. In this respect, EsCHER, which is built with high performance motors, small low mass cameras, and a dedicated parallel image processor (i.e. DataCube), outperforms the tracking of the system presented in this dissertation. The EsCHER system is, however, signi¯cantly more expensive with its custom optics, and specialized computing hardware.

The LIRA system mainly implements stabilization re°exes (see Chapter II). The gaze stabilization relies on both inertial devices (for measuring head velocity) and image velocity (see Figures 95 and 96). The results show the eye velocity having the opposite direction against the head movement. In order to compare the system in the present study with LIRA, another experiment was conducted on the ISAC head.

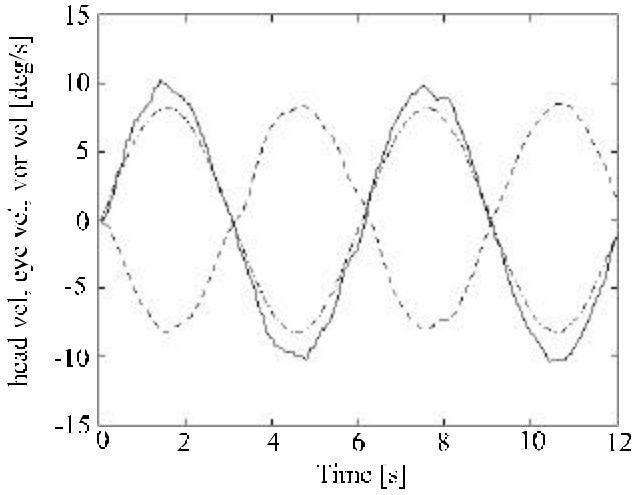


Figure 95: Inertial sensor output (continuous line), the head velocity (dash-dot line), and the generated camera movement (dash line). Reprinted from [71].

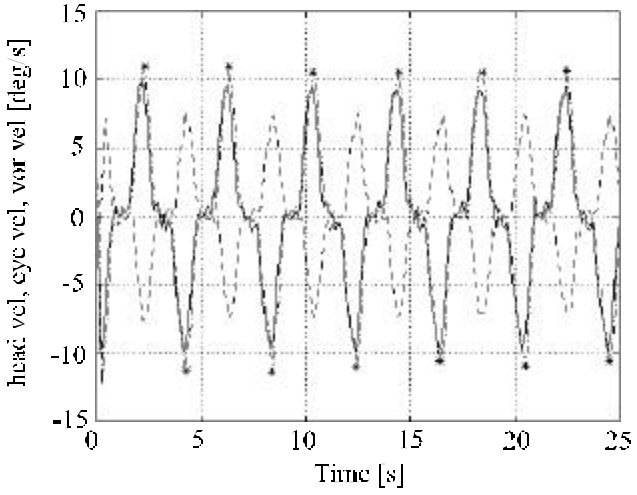


Figure 96: Sample data during one gaze stabilization experimental session (reprinted from [71])

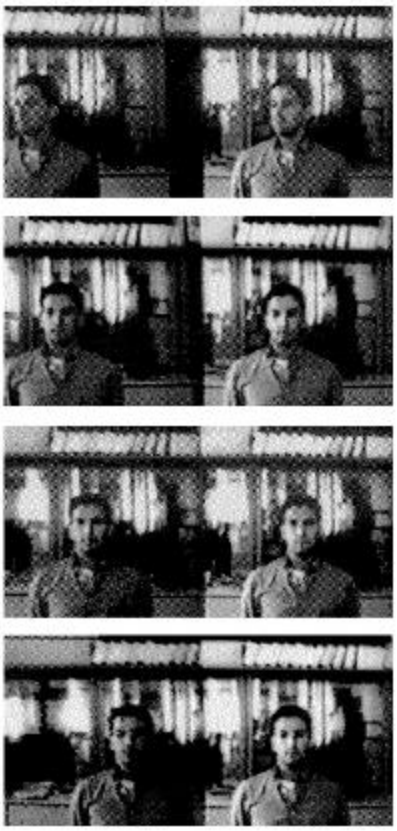


Figure 97: Binocular frames during LIRA stabilization experiment. Left: non stabilized camera; right: stabilized camera (reprinted from [46])

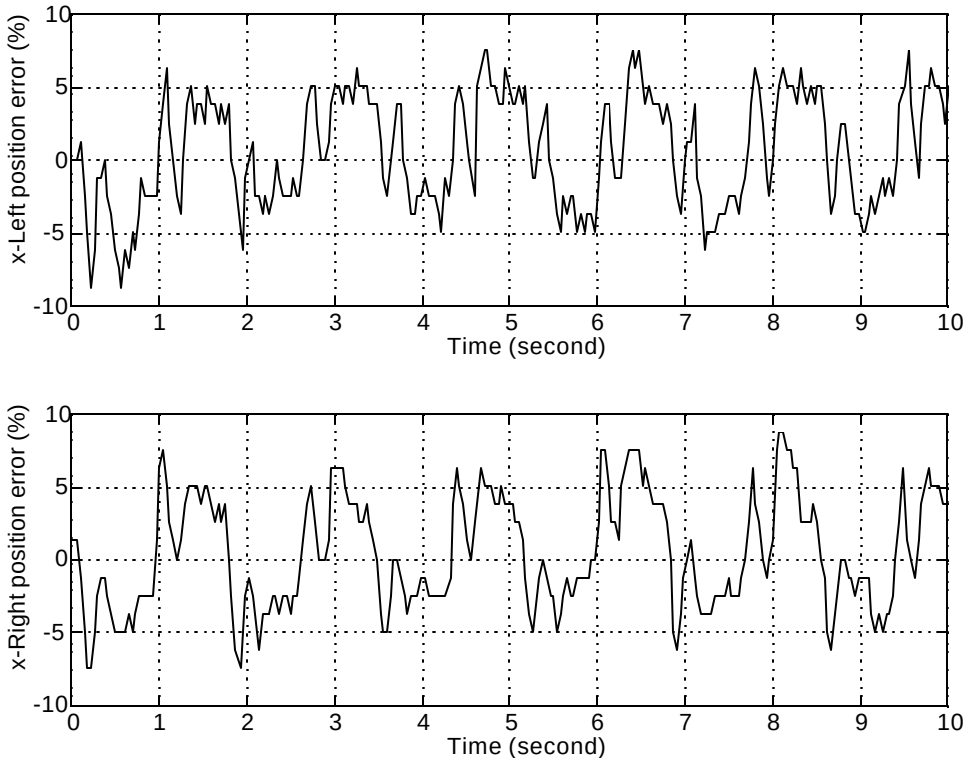


Figure 98: Target position error during the eyes stabilization

This experiment was similar to the LIRA experiment, moving the pan motor from left to right and allowing the verge motors to stabilize against the pan movement. Figures 98, 99 and 100 show the result of the experiment. Note that, as is shown in Figure 99, the verge motors move in the opposite direction of the pan motor and, as is shown in Figure 98, the target remains at the center of the image (inside the dead zone). The result is similar to the behavior of LIRA.

Saccade Experiments with Single and Double Step Stimuli

The following section discusses the experiments which are designed to observe characteristics of saccade reactions to both single and double steps of a target. The objective of the experiment is to compare some behaviors from the present saccadic

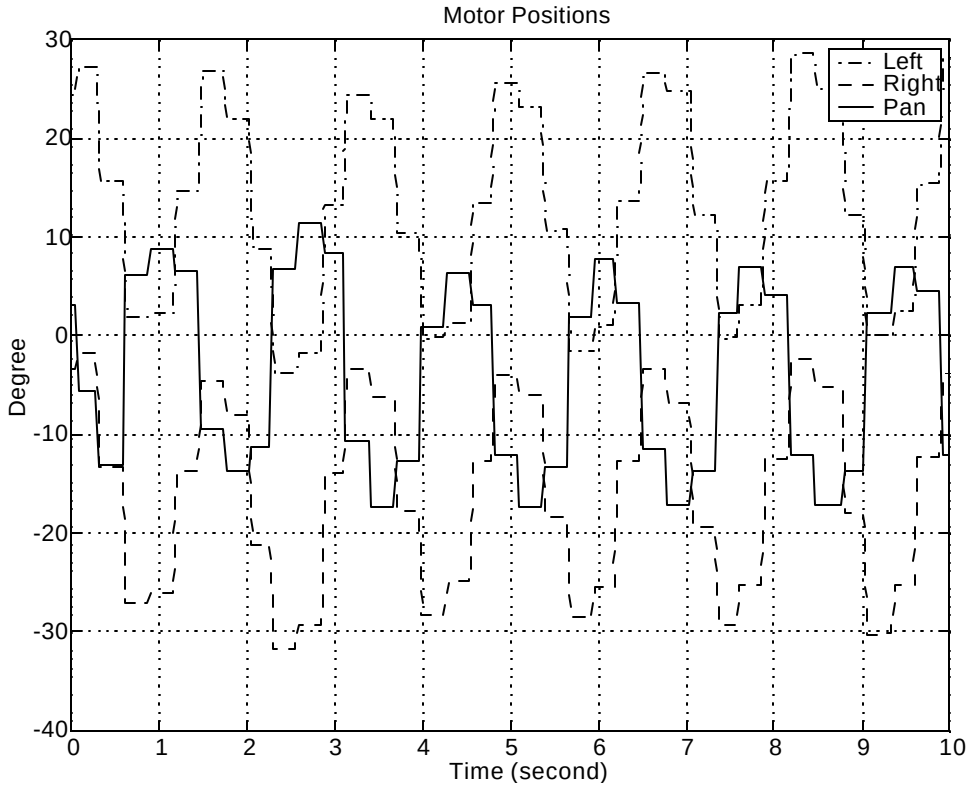


Figure 99: Motors position during the eyes stabilization

Figure 100: Image sequence during ISAC's eyes stabilization. The camera head keeps the eyes on the green target while the pan motor is moving.

system to the human saccadic system. Note that, because of many physical differences between the camera head and the human eye system, the active vision system presented here is designed to achieve similar funtionalities of the human eye system rather than to mimic how the human eye system works. Comparing the artificial human-like control of an active vision system to the human eye system is then worth to study.

This experimental session focuses on the behavior of the saccadic system because it is the primary control of the camera head system and it performs almost the same way as the human eye system does (compare to other eye movement modes). The outline of the experiment is similar to the work by Becker and Jürgens in [72] as described in the following section. Procedures, results and discussions are provided.

Experimental Setup

In [72], the human subjects were instructed: "if there is a change of target position, follow the target as rapidly as possible and fixate it anew". Horizontal eye movements were then recorded. Figures 101 and 102 show parameters of the response measures and the target patterns. The parameters are defined as follows:

- $'_1('_2)$ = amplitude of first (second) saccade of stimulus response.
- $R_1(R_2)$ = reaction time between first (second) stimulus step and onset of first (second) response saccade.
- D = time delay between second stimulus step and onset of first response saccade.
- I = interval between onset of first and onset of second response saccade.
- L = latency between end of first and onset of second response saccade.

The target pattern consisted of a random sequence of single and double steps. Single steps occurred with a frequency of 25.1% and amplitudes of 15, 30, or 60±: For the double step case, the target ¯rstly stepped to an initial position '$_i$ and then moved to its ¯nal position '$_f$: Any reaction of the eye happened after the target moved to its ¯nal position. All combinations of double steps can be summarized into four patterns (see Figure 102):

- SC \Stair Case" - the target steps twice in the same direction.
- PU \Pulse Undershoot" - the two target steps are of opposite direction with the second step being smaller than the ¯rst.
- SP \Symmetrical Pulse" - the second step returns the target to the position it started from.
- PO \Pulse Overshoot" - the two steps are of opposite direction with the second step larger than the ¯rst.

The interstep time (¿) was varied randomly between 50, 100, 150, and 200 msec.

Single Step Response

The following section describes the experimental setup for the ISAC camera head in order to obtain the similar environment as previously mentioned above. Details of setting up the experiment in software level are discussed in Appendix B.

The target used in this experiment consists of two di®erent color dots for single step pattern and three di®erent color dots for double step pattern (see Figure 103 and 106). For each color, there is a distinct color model associated with it for using in

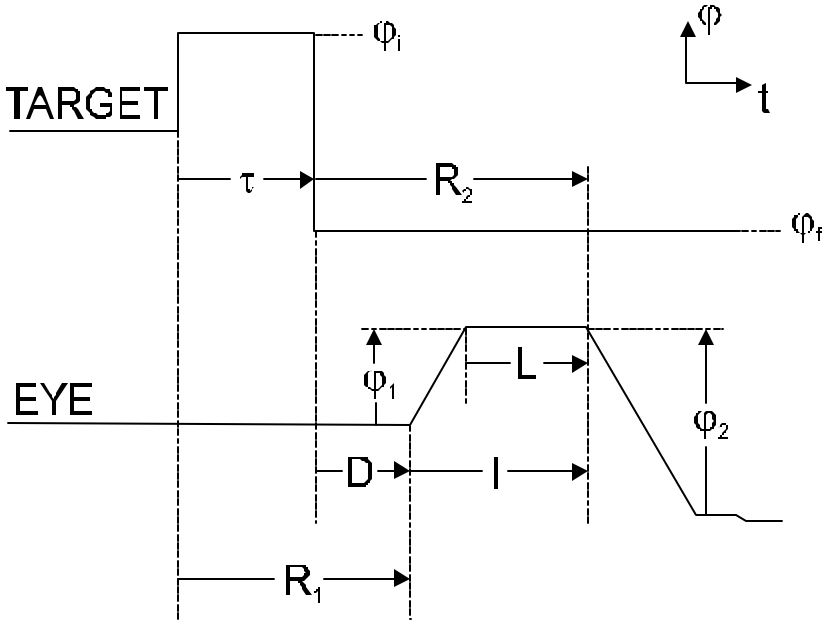


Figure 101: De¯nition of stimulus and response parameters (reproduced from [72])

color segmentation process. For single step target, the camera ¯xates on the greeen dot (labeled as \1") and then it steps to its ¯nal position at location of the maroon dot (labeled as \2"). The color dots are randomly located on the white background chart. Figure 104 shows the target pattern and camera response. The experiment had been conducted for 100 trials. The target positions in the image plane and camera trajectory were recorded. Reaction time (R_1) was manually measured from the recorded data and was found to have average of 220 msec with standard deviation of 67 msec (see Table 11 for all details). Figure 105 displays a frequency plot of R_1: The graph plot shows that the reaction time randomly occurs in a certain range of time (approximately 300 msec). It can be described as the total of a time comsuming by the visual processing unit and the camera controller unit, and the delay time of the camera hardware to response to the camera head commands.

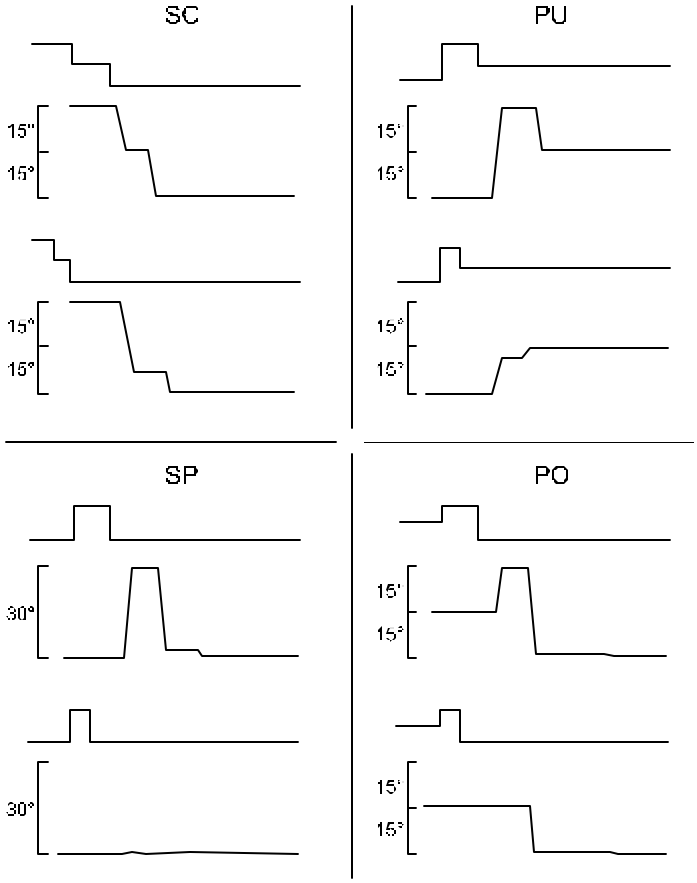


Figure 102: Each of four stimulus classes a typical stimulus pattern with an example of an initial angle response (upper pair of traces) and of a ¯nal angle response (lower pair of traces), upper trace of pair represents stimulus, lower trace response (reproduced from [72])

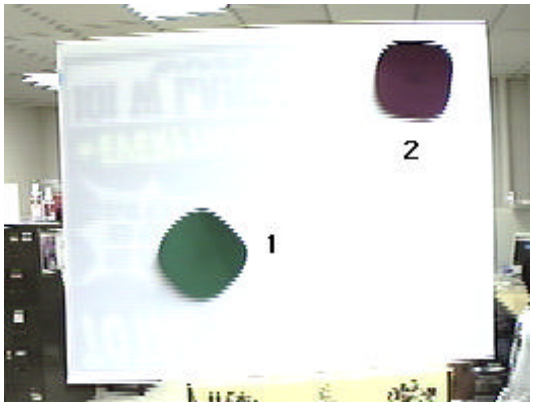


Figure 103: Two di®erent color dots for single step target

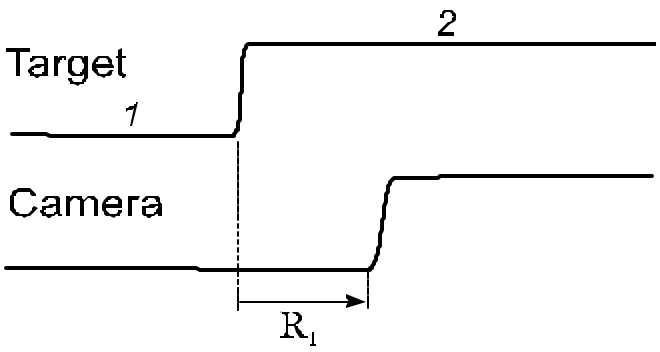


Figure 104: Single step: target pattern and camera response

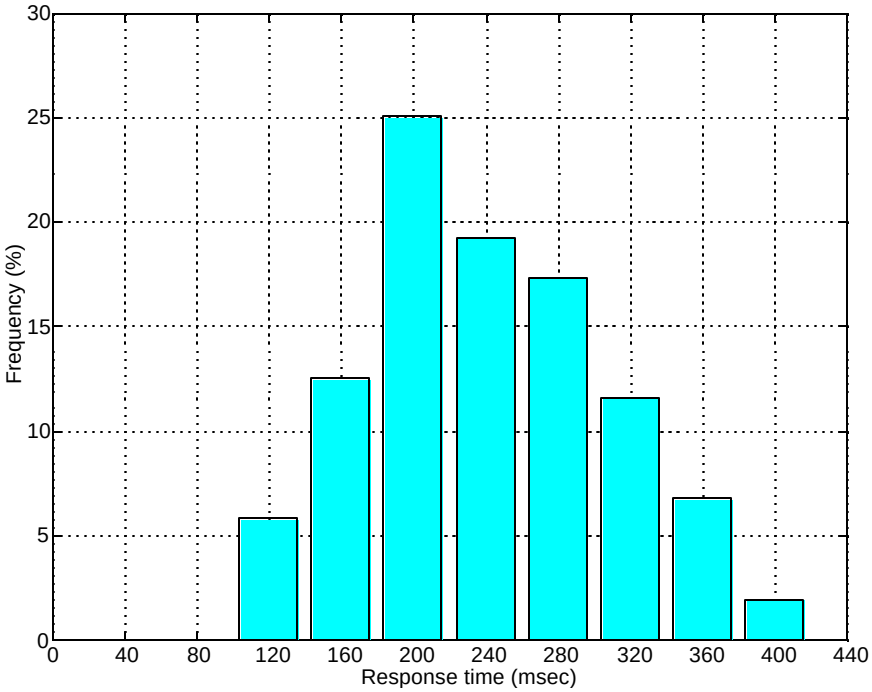


Figure 105: Single Step: frequency of response time

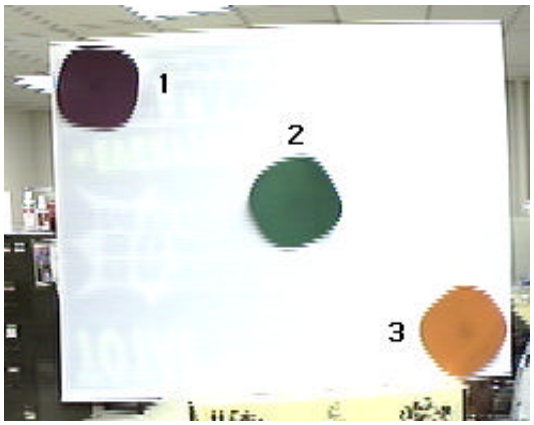


Figure 106: Three di®erent color dots for double step target

Double Step Response

In this experimental session, the target was moved in double step pattern. The same procedures as the single step response had been performed. Figure 106 shows three di®erent color dots used for the camera head to ¯xate (maroon, green, and orange - labeled as \1",\2", and \3"). Four target patterns, which are stair case, pulse undershoot, symmetrical pulse, and pulse overshoot, had been tested in the experiments.

For the stair case pattern, the ¯rst, second, and third ¯xation point of the camera are at location 1, 2, and 3, respectively. The camera stays ¯xating on the location 2 with a random interstep time (¿) of 50, 100, 150, and 200 msec. Figure 107 shows the target pattern and camera response. The experiments had been conducted for 800 trials (200 trials for each interstep time). The target positions in the image plane and camera trajectory were recorded. R_1 and R_2 were manually measured from the recorded data. Figures 108 and 109 display frequency plots of R_1 and R_2, respectively. Statistical results of R_1 and R_2 are shown in Table 11.

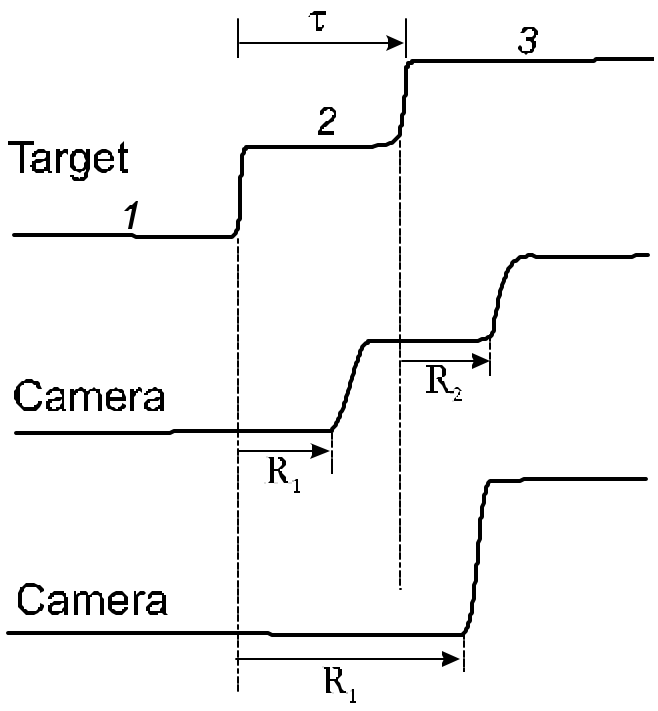


Figure 107: Stair case: target pattern and camera responses

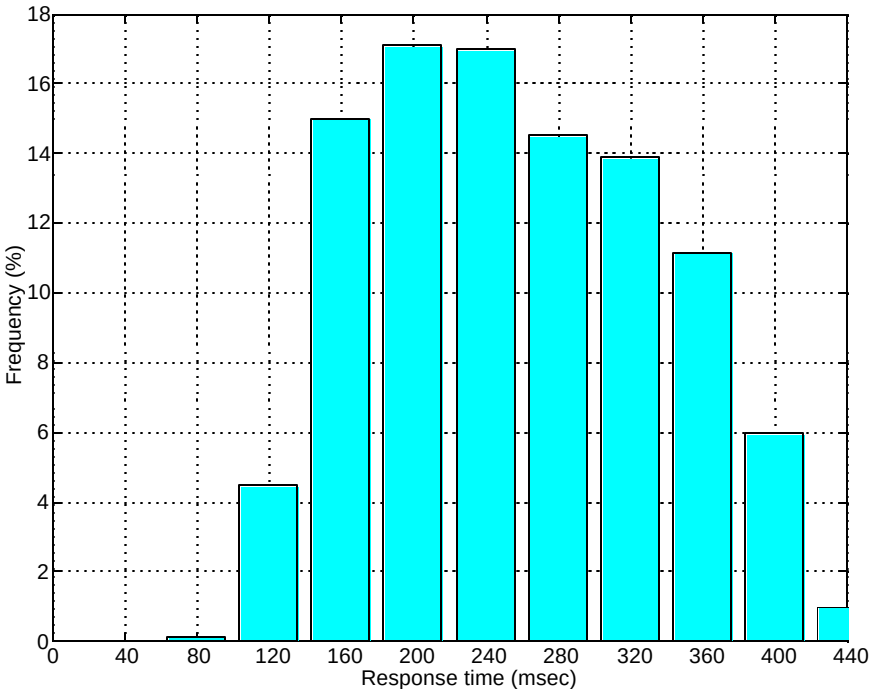


Figure 108: Stair case: frequency of response time (R_1)

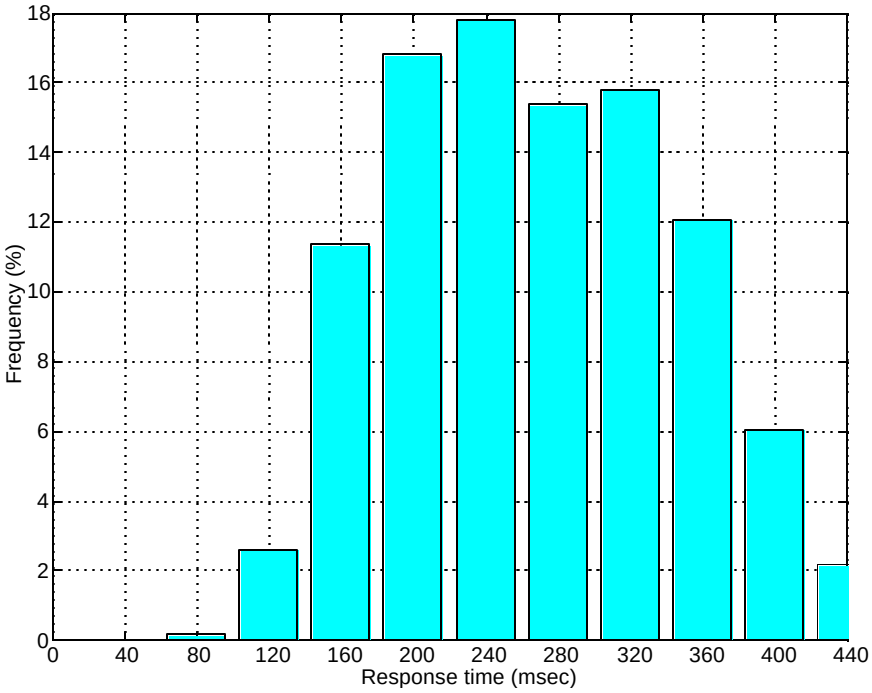


Figure 109: Stair case: frequency of response time (R_2)

Table 11: Statistical summary of R_1 and R_2 for all camera responses

Target pattern	R_1 (msec)	σ_{R_1}	R_2 (msec)	σ_{R_2}
Single step	220	67	-	-
Stair case	237	77	265	80
Pulse undershoot	242	80	272	75
Symmetrical pulse	240	79	290	72
Pulse overshoot	239	80	261	78

Same procedures had been conducted for the pulse undershoot, symmetrical pulse, and pulse overshoot response. All the target patterns and camera responses are shown in Figure 110, 113, and 116. The target sequence for the camera head to ¯xate on is numerated in the graph plots. Figures 111, 112, 114, 115, 117, and 118 show frequency plots of R_1 and R_2 for all of camera responses. Table 11 shows statistical results of R_1 and R_2:

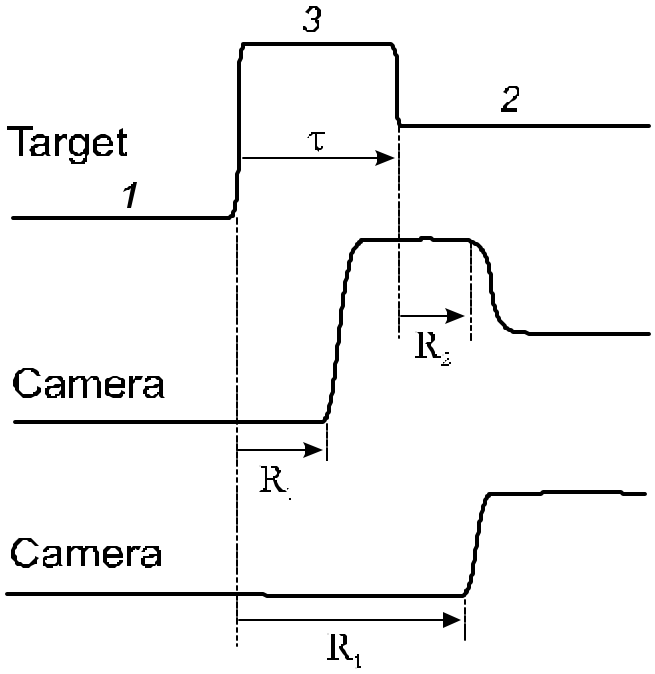


Figure 110: Pulse undershoot: target pattern and camera responses

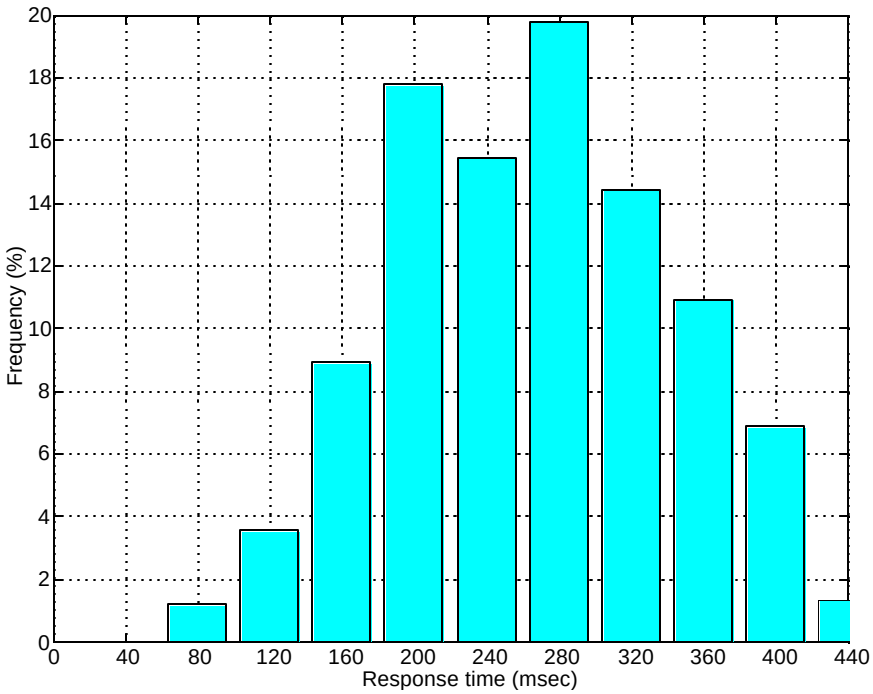


Figure 111: Pulse undershoot: frequency of response time (R_1)

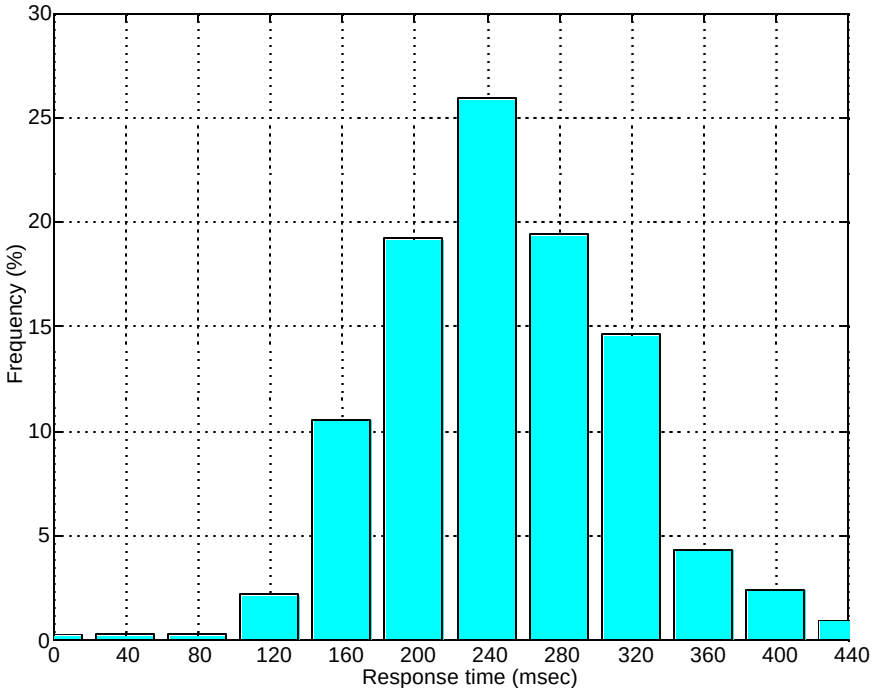


Figure 112: Pulse undershoot: frequency of response time (R_2)

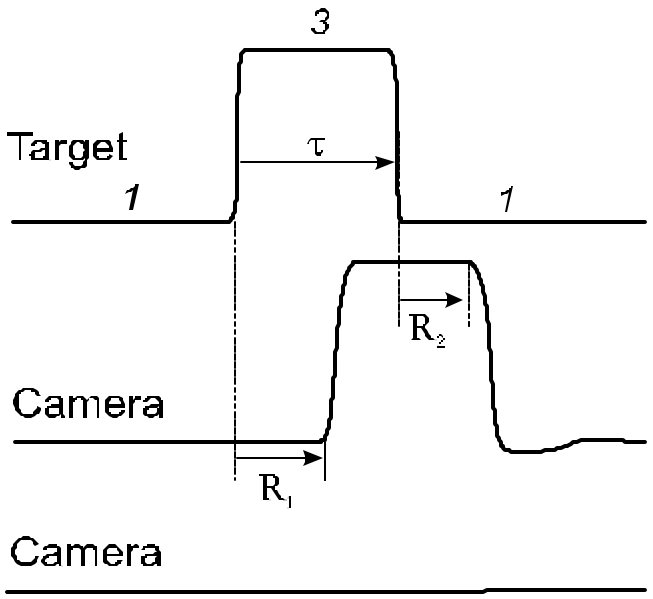


Figure 113: Symmetrical pulse: target pattern and camera responses

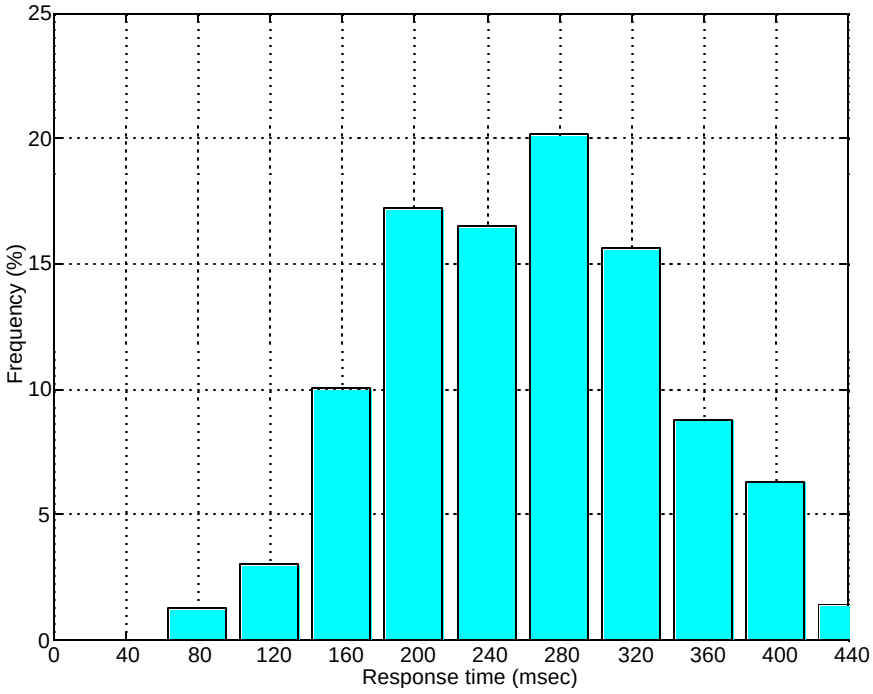


Figure 114: Symmetrical pulse: frequency of response time (R_1)

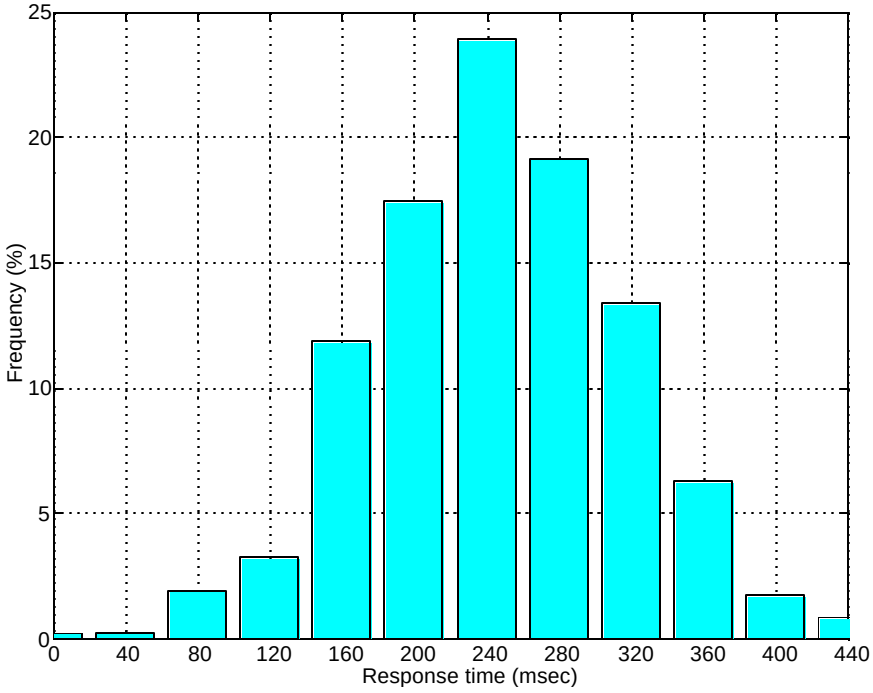


Figure 115: Symmetrical pulse: frequency of response time (R_2)

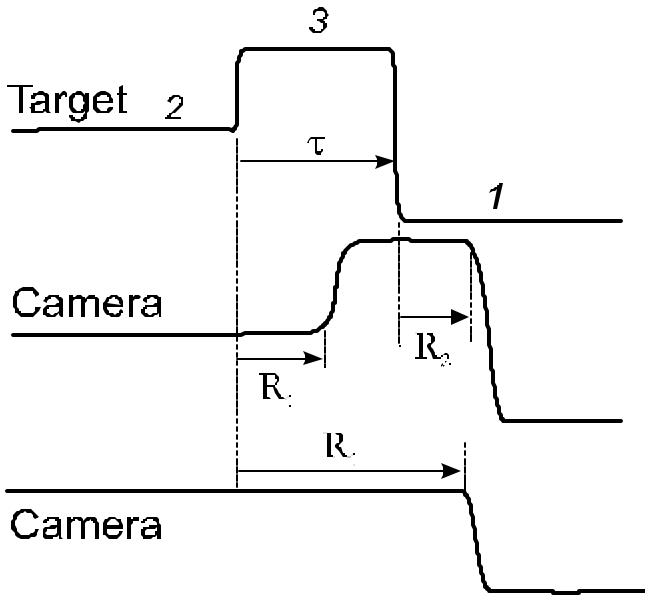


Figure 116: Pulse overshoot: target pattern and camera responses

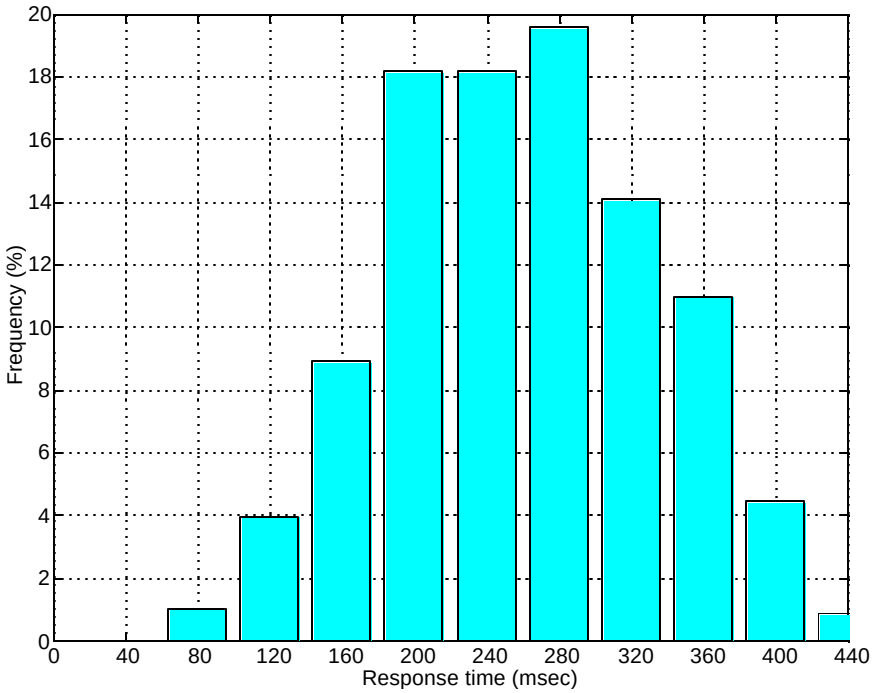


Figure 117: Pulse overshoot: frequency of response time (R_1)

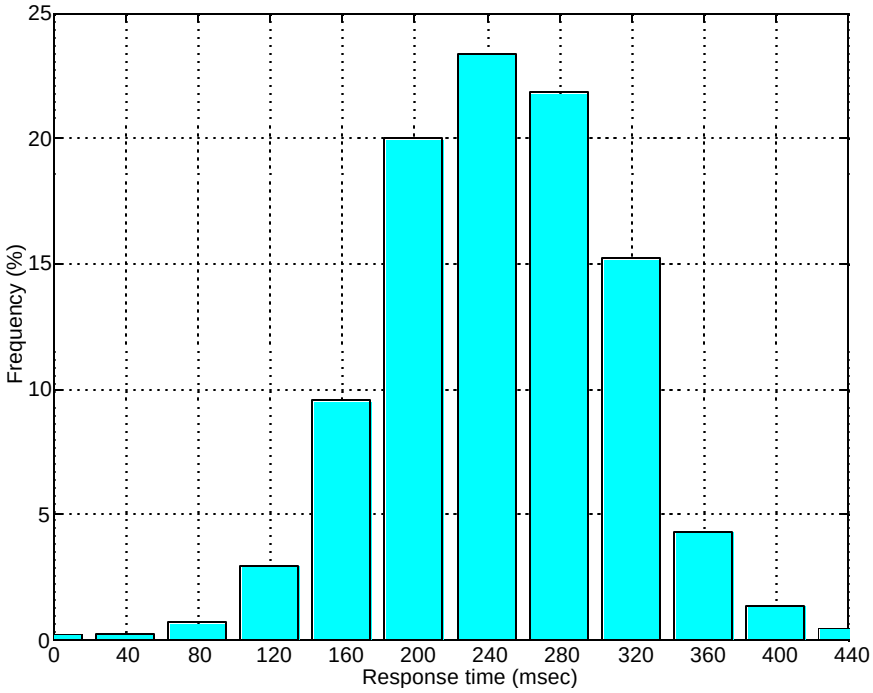


Figure 118: Pulse overshoot: frequency of response time (R_2)

Discussion

The saccade experiments had been performed to observe how the camera responds to single and double step stimuli. Summary of the study in human reports that there are roughly two types of saccadic responses to double steps of a target: (i) initial angle responses, and (ii) ¯nal angle responses (see [72] for all details). Initial angle responses consist of two large saccades that approximately correspond to the two target steps. Final angle responses consist of only one main saccade that moves the eyes closely to the ¯nal target position. Figure 102 show these behaviors.

Most of the properties of double-step responses are examined by D which is a time delay between second stimulus step and onset of ¯rst response saccade. This delay determines the amplitude of the ¯rst response saccade ($'_1$) and the interval (I) (if the response is an initial angle responses). The delay depends on the interstep time

¿ and R_1. The experiments control the interstep time whereas the reaction time is a random variable.

Beside the delay, the target pattern obviously a®ects the characteristics of double-step responses. That is the double step pattern yields very similar response characteristics, for instance, a pulse overshoot results in a pulse-overshoot shape of resonse.

By comparing to the experiments on ISAC camera head, the results are summarized as follows:

- The reaction times obtained from single steps were used as a reference for the double-step reaction times. The single-step reaction time of human eye increases for the greater distance between the initial position and ¯nal position of the target (with averages of 276, 287, and 311 msec for 15, 30, and 60± step responses) while the single-step reaction time (with averages of 220 msec) of the camera head does not depend on such a factor . It can be determined as the total of a time comsuming by the visual processing unit and the camera controller unit, and the delay time of the camera hardware to response to the camera head commands.

- For the double step target of the camera head, the reponses may roughly be divided into two types. The ¯rst type of the response occurs when the reaction time R_1 is less than interstep time ¿. In this case, the camera response to double-step target belongs to its class (SC, PU, SP, and PO). The camera ¯rst ¯xates on the initial target position and then follows the target to its ¯nal position. The amplitudes of the ¯rst and second saccade response depend on the position of the target with respect to the center of the image plane. Measures of R_1

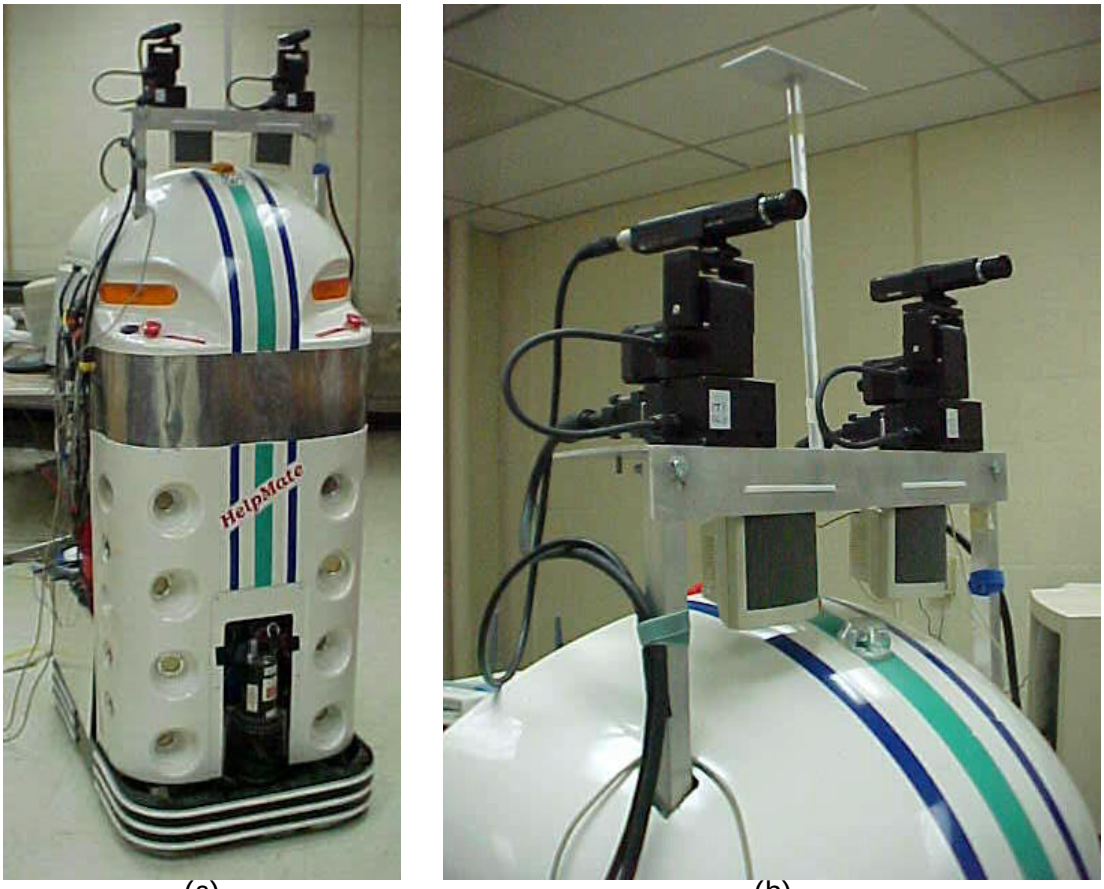
and R_2 are shown to be random. The second type of the response occurs when the reaction time R_1 is more than interstep time ¿. In this case, the camera will do saccade onto the ¯nal position of the target only. The amplitude of the ¯rst saccade reponse is determined by the distance between the target and the center of the image.

- The distributions of R_1 and R_2 share the same characteristic among all responses. These reaction times are nothing but a total delay time in both software and hardware level of the system which yields predictable behaviors for all camera responses to both single and double step stimuli.

Even though there is only some behavior shared between the human eye syetem and ISAC camera head system, the experiments show that the camera head somehow performs in similar manner as human does, e.g. respond to double step targets. There are many physical di®erences between the human eye system and the camera head system that must be taken into the account in order to build the system that performs and behaves like a human eye system.

System Portability

The prototype control system has been ported to another robot platform, called HelpMate, currently under development along with ISAC at the Intelligent Robotic Laboratory. HelpMate is a mobile robot with two pan-tilt units mounted on top of its head. Each pan-tilt unit, with a color CCD camera, is controlled independently (see Figure 119). Due to the structural di®erences between HelpMate's and ISAC's AVSs, further design and implementation are necessary to complete the HelpMate



(a) (b)

Figure 119: HelpMate: (a) mobile robot, (b) camera head

system. Some of the HelpMate design issues, however, are not relevant to the work presented here; hence, will not be discussed. For example, ISAC has one pan-tilt unit upon which a vergence unit is mounted. This mechanically couples both cameras to common pan-tilt motions. HelpMate, however, has two pan-tilt units. Each camera has independent pan and tilt, and vergence must be controlled using all four motors rather than the two in the ISAC system. Consequently, there is no common tilt axis between the two cameras. Thus, eye stabilization is needed against the entire robot body motion rather than just the camera head itself.

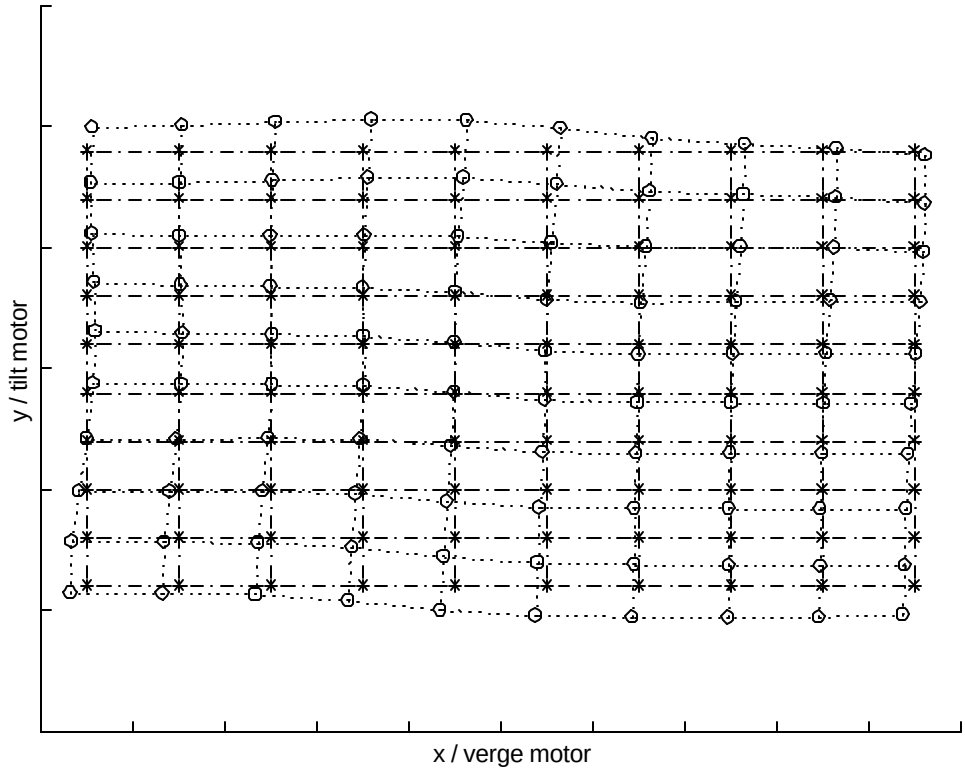


Figure 120: Saccade training map with 10 neural units of one hidden layer

Currently the saccade module has been successfully implemented and tested on the HelpMate head. The back-prop neural network with one hidden layer of 10 neural units (fewer than the one on the prototype which has 20 neural units) has been employed for the saccade function. Also, fewer sample data points have been used to train the neural network. The reason of having fewer sample data points is that to compare the training time and the generalization of the neural network. The system, however, exhibits the same desirable actions as the ISAC head, and appears to have a better performance than the proportional control currently used on the HelpMate. The result of the training saccade map is shown in Figure 120.

Because the HelpMate system is still very much under development, various software resources and facilities were not ready at the time of these experiments. Therefore no measurement data can be presented here. However, for greater insight into the overall performance and robustness of both the ISAC and HelpMate systems, the reader may view videos of real-time tracking experiments that are available on the Internet at:

\http://shogun.vuse.vanderbilt.edu/CIS/IRL".

System Weaknesses

Certain weaknesses of the system were found during the experimental sessions. Firstly, because the primary control of the motors is saccade, and because of their design, the verge motors have open-loop control. Consequently, there is no feedback that guarantees the actual instaneous positions of the motors. The system relies entirely on the visual element (saccade) to control and update the position of the motors. Therefore it is possible that the camera head incorrectly updates the motors' positions, even though the saccade module provides an accurate estimate of the motor movement.

Secondly, due to its physical limitations, the camera head cannot handle a high-acceleration target. Unlike the human head, which is capable of high speed and accurate movement, the physical limit of the camera head structure prevents the system from tracking above a certain level of target acceleration. Normally, the camera head can recover the loss of the target in a later time frame if the target is still located within the image plane. It should be noted that physical improvements to the ISAC head are possible which would improve its performance.

Finally, camera head oscillation can occur in some situations due to the control of the motors, for example, when the pan motor has to make a large movement toward the target due to the high speed of the target which causes a large displacement of the target position in consecutive frames. In this system, the camera head uses position control rather than velocity control of the motors (due to hardware limitations). This can lead to unstable camera head motion, especially during the eye stabilization process. This behavior results from a velocity mismatch between the verge motors and the pan motor. The proposed method uses only visual information for eye compensation against head motion; consequently, without knowledge of the current velocity of each motor, the greater the amount of movement for both verge and pan motors, the higher the chance of their velocity mismatch.

Nevertheless, the system still demonstrates highly robust and efficient tracking behavior. With these basic tracking skills, one can use this system as an integral part of a general active vision system on any robot platform with similarly configured AVS.

Summary

Control methods have been tested through experimentation. Five basic human-like eye movements have been robustly attained, providing demonstrable improvements in overall performance over the previous proportional controller. The AVS performs with more than adequate speed and accuracy given environment and tasks for which ISAC was designed. Furthermore, because of its simple structure, this AVS is conveniently portable to other robots with active vision systems. Comparisons between this system and human eye movement system have been investigated. A few

weaknesses of the system have also been discussed. Overall, however, these human-like camera head motions demonstrate a smoother, more stable, and more robust performance than traditional control methods for an active camera head.

CHAPTER VI

CONCLUSION

This dissertation has presented both the design and implementation of a camera head control system for a humanoid robot using an active vision system. The camera head controls are designed to be similar to human eye movements. These can be classi¯ed into three voluntary movements: saccades, smooth pursuit, and vergence, and two involuntary movements: vestibulo-ocular re°ex and opto-kinetic re°ex.

The purpose of this dissertation - the design and implementation of ¯ve basic human-like eye movement control schemes on the ISAC camera head - has been ful¯lled. These basic controls have been demonstrated to enable human-like camera head motion. The system has been integrated as part of a visual attention system for ISAC, and demonstrates smoother, more accurate, and more stable and robust motion than traditional camera head control methods. In summary, the following goals have been accomplished:

- Robust attainment of ¯ve basic human-like eye movements, shown to improve overall performance.

Each eye movement control has been successfully implemented and tested through experimentation.

- The saccade module can ¯xate the camera head at maximum speed to center a target. The back-propagation neural network provides e±ciently accurate mapping between the camera image plane and motor positions for the saccade function. The

simple structure of the network also makes the design and implementation less difficult than other explicit functional methods. It is also adaptive in that the system can be retrained to achieve greater accuracy of the saccade function.

- The smooth-pursuit module can track an object at moderate speeds and keep it centered in the images. The smooth-pursuit system uses both target position and velocity for properly controlling the camera head. An additional smoothing filter enables smoother movement of the motors than without it.

- The vergence system successfully keeps both left and right cameras fixating on the same object using a disparity measure. The system can efficiently estimate the disparity and use it to adjust the verge motors for minimum disparity between the left and right images.

- The eye stabilization module, which includes both vestibulo-ocular reflex (using pre-determined relationship between eyes and a head) and opto-kinetic reflex (using visual information), can compensate for the movement of the verge motors (eyes) against the pan motor (head) motion. While the system keeps the pan motor following the target (within its mechanical limits), the verge motors compensate so that the target properly remains at the center of the images.

Finally, all five human-like eye movement controls run simultaneously. The system can track the moving target continuously until the target is out of the visual field (determined by the mechanical limits of the AVS).

[2] Performance of all five basic human-like eye movement controls with desirable speed and accuracy, but without additional special hardware

Without additional special hardware such as a dedicated high speed image processor, the system has successfully demonstrated all ¯ve basic human eye movements with desirable speed and accuracy, using only simple, straightforward image-processing technology. With only personal computers as a computing and controlling unit, the system can perform all ¯ve basic controls e±ciently. Even though the ISAC head was built with regular step motors, the system can still achieve the human-like motion without using any special kind of device such as an inertial mechanisms (gyros or accelerometers).

In summary, by simply using the neural network for the saccade, color information for the smooth pursuit, 1-D correlation-based disparity estimate for the vergence, 1-D correlation-based image velocity estimate for the opto-kinetic re°ex, and vision-based information for vestibulo-ocular re°ex, a real-time human-like motion tracking system has been achieved.

- Convenient portability of the prototype system to other binocular vision platforms

The standard PC basis of this system, combined with its relatively simple design and implementation, makes it possible to port this AVS control system to other binocular vision platforms with minimum e®ort. The system has successfully been implemented on the ISAC head, which has a simple structure. Additionally, the use of simple image-processing routines makes it far less complicated to design robust software for controlling the system. Finally, the porting the prototype control system to HelpMate, the mobile robot platform, has been successfully demonstrated.

Contributions

There are original contributions in the work proposed here. Firstly, without a sophisticated computational hardware binocular head system, a basic four-degrees-of-freedom AVS head can improve overall performance with the ¯ve human-like eye movement controls. These control methods can be implemented at no extra cost, as they do not use special hardware such as high-performance DSP devices for intensive computations, nor gyroscopes and accelerometers for head-motion measures. In addition, the software control system is fully recon¯gurable. That allows for creating, editing, inserting or deleting di®erent system con¯gurations. Each component runs independently which allows the system to perform tasks in parallel as a distributed system.

Secondly, each human-like eye movement control method is unique. Each uses visual computations only to reach the desired goal. The unique control methods can be summarized as follows:

For the implementation of saccades, the method used was most closely related to the work of the COG project (see [8]). Their camera head, however, is much more complex than the ISAC head. They train a mapping between head-eye coordinates for saccade action using image correlation. In the study reported here, color information is utilized to assist the neural network training process (COG does not use color); this provides faster and more accurate target location than the use of image-patch correlation, such as COG uses. The monochrome tracking, however, gives COG some advantages in term of speed and tracking of shape-independent objects.

The smooth-pursuit control method employs target velocity from optical °ow estimation, which is essentially the same concept as that used in much other work (see

[13],[9], and [12]). However, the traditional motion estimation and camera-motion compensation di®er from the work proposed here. By using color information, the proposed motion segmentation method can be robustly implemented while maintaining computational simplicity.

Work is described in [13], [11], and [14] on vergence-tracking systems using disparity cues. In these phase-based disparity estimation is required to obtain the disparity cues. 2D methods are, however, nearly impossible to implement for any system without a special hardware module, since they are computationally expensive. The present study found that the use of color segmentation with a 1-D correlation-based disparity estimate gives a su±ciently accurate disparity estimate for driving vergence control. This method works su±ciently well to implement and achieve real-time performance on a regular PC-based machine.

Finally, equivalent VOR and OKR control using only visual information has been proposed. Many researchers have presented accurate stabilization models and implemented them with inertial sensor devices (see [8] and [16]). Without any of these special devices, the present work uses currently available visual systems to achieve VOR-like and OKN motion for head movements. Moreover, a human-assisted learning of VOR parameters for the camera head was uniquely designed and presented.

Future Work

In order to improve the system's performance and robustness, as well as make it more general-purpose for a tracking system, it would probably be useful to focus on the following issues:

Hardware Improvement

There are two main parts to any AVS: visual processing and mechanical control. The visual processing part can be improved with better equipment such as smaller CCD color cameras and lenses. Di®erent CCD cameras can a®ect how images appear with respect to features such as color balance, brightness and noise. The choice of lens for the camera is also important. For instance, a wide-angle lens provides a wider ¯eld of view, which allows the vision system to cover a greater space but at the cost of some image resolution. Space-variant lenses can also be very useful in image-processing routines such as motion segmentation [13] [73].

In order to build a system that has human-like performance, the hardware should have su±cient speed and accuracy. The speed and accuracy of the motors used in this project is far from that possessed by the human eye and head system. The design of the camera head with better hardware would result in a better performance of the binocular system [58]. Also, closed-loop control of the motors (that allows a high rate of position acquisition) should provide more °exibility and robustness from the control point of view. A motor controller that controls velocity can be more useful and provide better control than a position motor control.

Other special hardware, such as inertial devices [8][47] for gaze stabilization, could improve the tracking performance.

More Robust Control

Besides the speed of the system, robustness is probably the most desired goal of any control system. The proposed system relies entirely on visual processing to control the camera head. Most of the time, such a system will experience a delay

between the vision system and the control system. By increasing the visual processing speed, the system can reduce the delay to achieve a real-time performance. Despite the fact that high speed vision processing can be obtained, a delay still occurs within both the hardware and software units. Instead of relying only on visual processing, system robustness can be signi¯cantly improved by additional mechanical controls to overcome this problem [11] and to ensure that unstable states will occur less often (even if the vision system has lost sight of the target).

Visual Attention Network

In order to obtain a general-purpose active vision system, the visual attention network plays an important role. It acts like the human brain in telling the eyes where to look. By applying a more sophisticated visual attention network (e.g. [66]), the active vision system can perform considerably and biologically better in a more complex environment. This can also be very useful in integrating the active vision system into the humanoid robot.

BIBLIOGRAPHY

[1] R. A. P. II and M. Bishay, \Centering peripheral features in an indoor environment using a binocular log-polar 4 dof camera head," in Journal of Robotics and Autonomous Systems, vol. 18, p. 271, 1996.

[2] S. Asaad, M. Bishay, D. M. Wilkes, and K. Kawamura, \A low-cost ,dsp-based, intelligent vision system for robotic applications," in Proceedings of the 1996 IEEE International Conference on Robotics and Automation, (Minneapolis, Minnesota), pp. 1651{1661, April 1996.

[3] M. E. Goldberg, H. M. Eggers, and P. Gouras, \The ocular motor system," in Principles of Neural Science (E. R. Kandel, J. H. Schwartz, and T. Jessell, eds.), Appleton and Lange, 3rd ed., 1992.

[4] P. Gouras, Principles of Neural Science. Elsevier, 2nd ed., 1985.

[5] J. B. Barile, M. Bishay, M. E. Cambron, R. Watson, R. A. P. II, and K. Kawamura, \Color-based initialization for human tracking with a trinocular camera system," in Proceedings of the IASTED International Conference on Robotics and Manufacturing, (Cancun, Mexico), May 1997.

[6] B. K. P. Horn and B. G. Schunck, \Determining optical °ow," in Arti¯cial Intelligence, vol. 17, pp. 185{203, 1981.

[7] P. A. Laplante and A. D. Stoyenko, eds., Real-time Imaging: Theory, Techniques, and Applications. IEEE Press, 1996.

[8] R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. M. Williamson, \The cog project: Building a humanoid robot," in IAPR First International Workshop on Humanoid and Human Friendly Robotics, pp. 1{36, October 1998.

[9] D. Murray, \Driving saccade to pursuit using image motion," in International Journal of Computer Vision, vol. 16, pp. 205{228, 1995.

[10] K. Bradshaw, P. McLauchlan, I. Reid, and D. Murray, \Saccade and pursuit on an active head/eye platform," in Image and Vision Computing Journal, vol. 12, pp. 155{163, 1994.

[11] D. Coombs and C. Brown, \Real-time smooth pursuit tracking for a moving binocular robot," in Computer Vision and Pattern Recognition, pp. 23{28, 1992.

[12] H. Araujo, J. Batista, P. Peixoto, and J. Dias, \Pursuit control in a binocular active vision system using optical °ow," in ICPR - 13th International Conference on Pattern Recognition, August 1996.

[13] S. Rougeaux and Y. Kuniyoshi, \Velocity and disparity cues for robust real-time binocular tracking," in Proc. Int. Conf. On Computer Vision and Pattern Recognition (CVPR'97), (San Juan, Puerto-Rico), pp. 1{6, June 1997.

[14] K. Pahlavan, T. Uhlin, and J.-O. Eklundh, "Dynamic fixation and active perception," in International Journal of Computer Vision, vol. 17, pp. 113–135, 1996.

[15] C.-J. Westelius, Focus of Attention and Gaze Control for Robot Vision. PhD thesis, Electrical Engineering, Linkoing University, Sweden, 1995.

[16] F. Panerai and G. Sandini, "Visual and inertial integration for gaze stabilization," in SIRS'97, (Stockholm), 1997.

[17] B. A. Wandell, Foundations of Vision. Sinauer Associates, Inc., 1995.

[18] M. A. P. Mark F. Bear, Barry W. Connors, Neuroscience, Exploring the Brain. Williams & Wilkins, 1996.

[19] M. Wessler, "A modular visual tracking system," Master's thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1995.

[20] M. J. Marjanović, B. Scassellati, and M. M. Williamson, "Self-taught visually guided pointing for a humanoid robot," in From Animals to Animats: Proceeding of 1996 Society of Adaptive Behavior, pp. 35–44, Massachusetts: Cape Cod, 1996.

[21] J. Yamato, "Tracking moving object by stereo vision head with vergence for humanoid robot," Master's thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1998.

[22] S. Rougeaux and Y. Kuniyoshi, "Robust real-time tracking on an active vision head," in Proceedings International Conference on Intelligent Robots and Systems (IROS'97), (Grenoble, France), September 1996.

[23] B. Scassellati, "A binocular, foveated active vision system," tech. rep., Massachusette Institute of Technology, 1997.

[24] R. Krauzlis and S. Lisberger, "A control systems model of smooth pursuit eye movements with realistic emergent properties," in Neural Computation, vol. 1, 1989.

[25] S. Lisberger, E. Morris, and L. Tychsen, "Visual motion processing and sensorymotor integration for smooth pursuit eye movements," in Ann. Rev. Neurosci., vol. 10, pp. 97–129, 1987.

[26] D. J. Coombs, Real-Time Gaze Holding in Binocular Robot Vision. PhD thesis, University of Rochester, June 1992.

[27] D. J. Coombs and C. M. Brown, "Real-time binocular smooth pursuit," in International Journal of Computer Vision, vol. 11, pp. 147–164, 1993.

[28] S. Rougeaux, N. Kita, Y. Kuniyoshi, S. Sakane, and F. Chavand, \Binocular tracking based on virtual horopters," in IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2052{2057, 1994.

[29] D. J. Coombs and C. M. Brown, \Cooperative gaze holding in binocular vision," in IEEE Control Systems, pp. 24{33, 1991.

[30] S. Rougeaux, N. Kita, Y. Kuniyoshi, and S. Sakane, \Tracking a moving object with a stereo camera head," in 11th Annual Conference of Robotics Society of Japan, pp. 373{376, 1993.

[31] N. Kita, S. Rougeaux, Y. Kuniyoshi, and S. Sakane, \Thorough zdf-based localization for binocular tracking," in Proc. MVA'94 (IAPR Workshop on Machine Vision Applications), 1994.

[32] P. von Kaenel, C. M. Brown, and D. J. Coombs, \Detecting regions of zero disparity in binocular images," tech. rep., University of Rochester, 1991.

[33] C. L. Fennema and W. L. Thompson, \Velocity determination in scenes containing several moving objects," in Computer Graphics and Image Processing, vol. 9, pp. 301{315, 1979.

[34] H. Araujo, J. Batista, P. Peixoto, and J. Dias, \Gaze control of a binocular active vision system using optical °ow," in SIRS'96 - 4th International Symposium on Intelligent Robotics Systems, (Lisbon), 1996.

[35] J. Batista, P. Peixoto, and H. Araujo, \Real-time vergence and binocular gaze control," in IROS97 - 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, (France), September 1997.

[36] J. Batista, J. Dias, H. Araujo, and A. T. Almeida, \The isr multi-degrees-of-freedom active vision robot head," in M2Vip95 - Second International Conference on Mechatronics and Machine Vision in Practice, (Hong Kong), 1995.

[37] R. Manzotti, R. Tiso, E. Grosso, and G. Sandini, \Primary ocular movements revisited," tech. rep., LIRA-Lab-DIST - University of Genova, Genova, Italy, November 1994.

[38] M. Jekin and J. Tsotsos, \Techniques for disparity measurement," in CVGIP: Image Understanding, vol. 53, pp. 14{30, 1991.

[39] A. P. Witkin, D. Terzopoulis, and M. Kass, \Signal matching through scale space," in Proceedings 5th National Conference on AI, (Philadelphia), pp. 714{719, 1986.

[40] M. M. Marefat, L. Wu, and C. C. Yang, \Gaze stabilization in active vision { i. vergence error extraction," in Pattern Recognition, vol. 30, pp. 1829{1842, 1997.

[41] A. Maki, T. Uhlin, and J.-O. Eklundh, \Phase-based disparity estimation in binocular tracking," tech. rep., Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology (KTH), Stockholm, Sweden, 1993.

[42] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin, \Phase-based disparity measurement," in CVGIP: Image Understanding, vol. 53, pp. 198{210, 1991.

[43] T. Sanger, \Stereo disparity computation using gabor ¯lter," in Biological Cybernet, vol. 57, 1988.

[44] A. Maki and T. Uhlin, \Disparity selection in binocular pursuit," tech. rep., Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology (KTH), Stockholm, Sweden, 1993.

[45] M. M. Marefat, L. Wu, and C. C. Yang, \Gaze stabilization in active vision { ii. multi-rate vergence control," in Pattern Recognition, vol. 30, pp. 1843{1853, 1997.

[46] F. Panerai and G. Sandini, \Oculo-motor stabilization re°exes: Integration of inertial and visual information," in Neural Networks, vol. 11, February 1995.

[47] F. Panerai, \Inertial sensors for controlled camera systems," tech. rep., LIRA-Lab-DIST - University of Genova, Genova, Italy, February 1995.

[48] C. Capurro, F. Panerai, and E. Grosso, \The lira-lab head: mechanical design and control," tech. rep., LIRA-Lab-DIST - University of Genova, Genova, Italy, August 1993.

[49] \http://www.lira.dist.unige.it/.".

[50] \http://www.isr.uc.pt/.".

[51] \http://www.neuroinformatik.ruhr-uni-bochum.de/ini/projects/namos/namos.html.".

[52] \http://viriato.isr.ist.utl.pt/.".

[53] J. Santos-Victor, F. van Trigt, and J. Sentieiro, \Medusa - a stereo head for active vision," tech. rep., Grenoble, France, July 1994.

[54] L. Berthouze, S. Rogueaux, F. Chavand, and Y. Kuniyoshi, \Calibration of a foveated wide-angle lens on an active vision head," in Conference of Computer Vision and Pattern Recognition, (San Francisco), June 1996.

[55] L. Berthouze, S. Rougeaux, F. Chavand, and Y. Kuniyoshi, \A learning stereo-head control system," in World Automation Congress/International Symposium on Robotics and Manufacturing, (France), 1996.

[56] Y. Kuniyoshi, N. Kita, S. Rougeaux, and T. Suehiro, \Active stereo vision system with foveated wide angle lenses," in Asian Conference on Computer Vision, (Singapore), 1995.

[57] Y. Kuniyoshi, N. Kita, K. Sugimoto, S. Nakamura, and T. Suehiro, \A foveated wide angle lens for active vision," in Proceedings IEEE International Conference Robotics and Automation, 1995.

[58] A. Takanishi, S. Hirano, and K. Sato, \Development of an anthropomorphic head-eye system for a humanoid robot -realization of human-like head-eye motion using eyelids adjusting to brightness-," in Proceedings of the 1998 IEEE International Conference on Robotics & Automation, (Leuven, Belgium), May 1998.

[59] \http://diwww.ep°.ch/lami/team/carmona/.".

[60] \http://www.vision.auc.dk/ hic/auc-head.html.".

[61] H. I. Christensen, \The auc robot camera head," tech. rep., Aalborg University, Laboratory of Image Analysis, October 1995.

[62] \http://www.robots.ox.ac.uk/ lav/equip/yorick.html.".

[63] P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan, \A modular head/eye platform for real-time reactive vision," in Mechatronics Journal, vol. 3, pp. 517{535, 1993.

[64] B. C. Madden and U. M. C. von Seelen, \Penneyes: A binocular active vision system," tech. rep., GRASP Laboratory, Dept. of Computer and Information Science, University of Pennsylvania, December 1995.

[65] \http://www.cis.upenn.edu/ grasp/head/penneyes/penneyes.html.".

[66] J. A. Driscoll, R. A. P. II, and K. S. Cave, \A visual attention network for a humanoid robot," in Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robotic Systems, (Victoria, B. C., Canada), October.

[67] \http://www.dperception.com/.".

[68] A. Srikaew, M. Cambron, S. Northrup, R. P. II, D. Wilkes, and K. Kawamura, \Humanoid drawing robot," in Proceedings of the IASTED International Conference on Robotics and Manufacturing, (Ban®, Canada), July 1998.

[69] S. Charoenseang, A. Srikaew, D. Wilkes, and K. Kawamura, \3-d collision avoidance for a dual-arm humanoid robot," in Proceedings of the IASTED International Conference on Robotics and Manufacturing, (Ban®, Canada), July 1998.

[70] S. Rougeaux, Real-Time Active Vision for Versatile Interation. PhD thesis, Electrotechnical Laboratory, Humanoid Interaction Laboratory, Japan, February 2000.

[71] F. Panerai and G. Sandini, \Role of visual and inertial information in gaze stabilization," in SIRS'97, (Stockholm), 1997.

[72] W. Becker and R. Jrgens, \An analysis of the saccadic system by means of double step stimuli," in Vision Research, vol. 19, p. 967, 1979.

[73] G. Sandini, G. Metta, and J. Konczak, \Human sensori-motor development and arti¯cial systems," in AIR and IHAS'97, (Japan), February 1995.

[74] R. T. Pack, IMA: The Intelligent Machine Architecture. PhD thesis, Vanderbilt University, Sept. 1998.

[75] \http://developer.intel.com/vtune/per°ibst/ipl/.".

Appendix A

SYSTEM OVERVIEW

The system implemented in this work has been designed entirely on the special platform called Intelligent Machine Architecture (IMA) [74]. This platform is currently developed in Intelligent Robotic Laboratory, Vanderbilt University, USA. The following sections describe specific details of the software modules implemented in this work. Note that the IMA related definition is notated by Typewriter typeset.

Agent-Level System Overview

The system is separated into three main agents: visual attention network, eye motion center agent, and camera agent (see Figure 121).

Visual Attention Agent

Visual attention agent is a general agent that provides necessary information about the target, e.g. position, velocity, and successive image displacement, to the rest of the system. Most of the time only target's position is required by other agents. Example of the visual attention agent used in this work is shown in Figure 122.

The PXCDevice component encapsulates all frame grabber functionalities. It acquires sequence of images from CCD color camera and puts the color images into an ImageRep representation. The color images are then available to the rest of the system.

The ColorSegmenter performs color segmentation on the input images based on

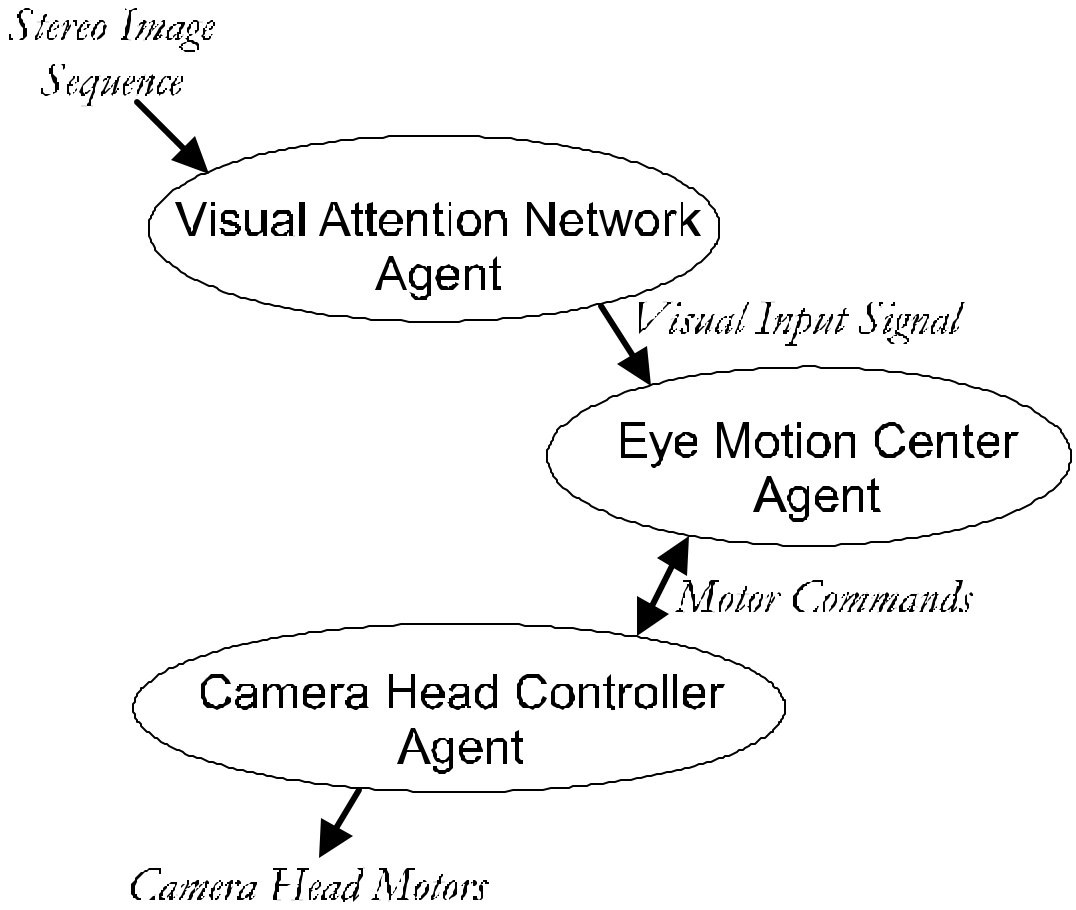


Figure 121: Overview Diagram of High-level Agents

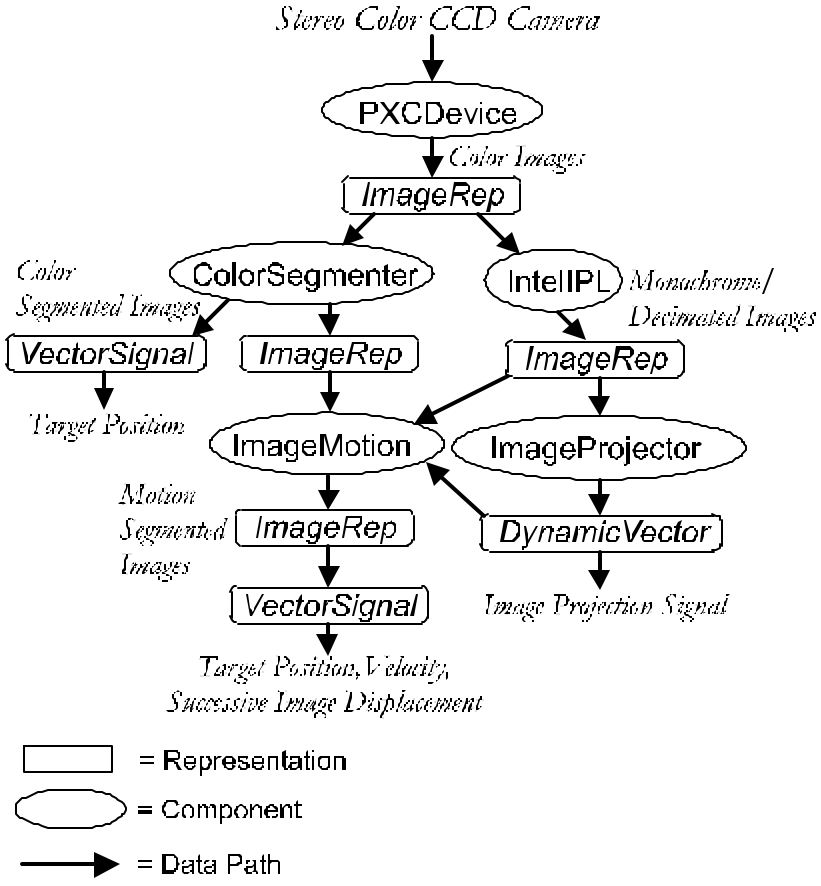


Figure 122: Visual Attention Agent

a color model stored in the system. The output, color segmented image, is a binary image which white blobs represent desire colors indicated by the color model. The position of these blobs in the image plane is determined by L-1 norm which yields the most likely largest blob's position.

The color segmented images are also supplied to ImageMotion component which performs a color segmentation-based motion detection. The ImageMotion component also requires the monochrome/decimated images from an IntelIPL in order to compute the motion segmentation. Moreover, the ImageMotion also calculates a successive image displacement which represents the camera head ego motion. The output of the ImageMotion component the composes of target's position, velocity, and successive image displacement. The ImageMotion component sends these output to a VectorSignal representation. This allows these pieces of information available to the rest of the system.

The IntelIPL component encapsulates the image processing library provided by Intel® in order to achieve the optimum performance for the MMX™ technology [75]. The color-to-monochrome and decimate functions are employed here to convert color images into gray-scale images and scale them up or down.

The ImageProjector projects image intensities onto x- or y-axis. These projection signals are utilized by the ImageMotion component for the image motion computation. They are also available for other modules such as Vergence component which will be discussed in a later chapter. The ImageProjector provides the projection output to a DynamicVector representation.

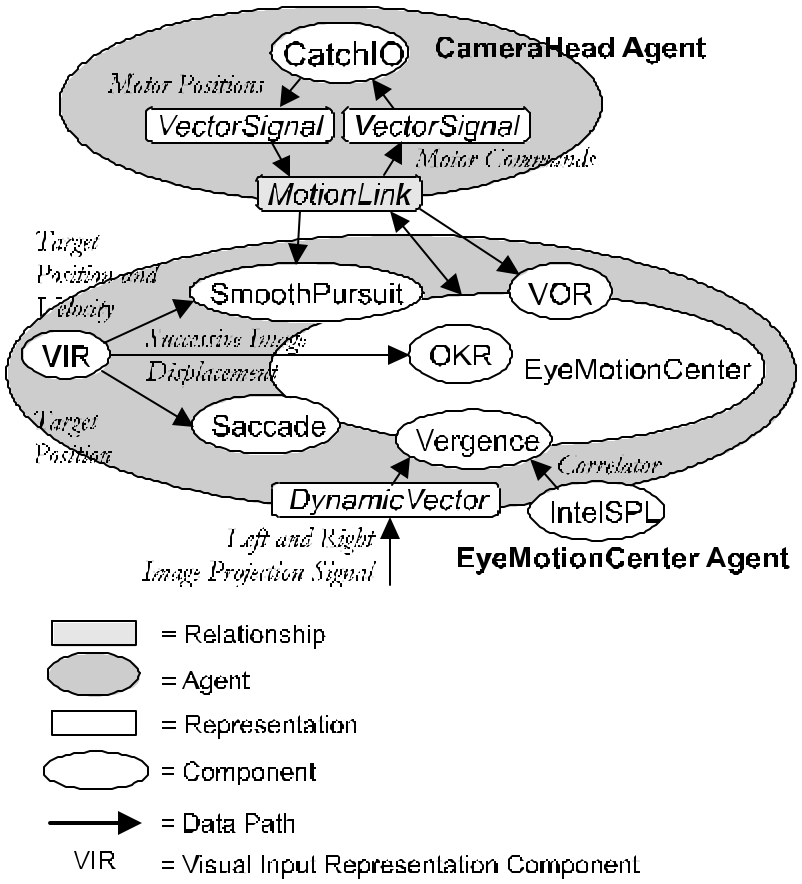


Figure 123: EyeMotionCenter and Head Agents

EyeMotionCenter Agent

The EyeMotionCenter component ties all ¯ve human eye movement components together. Those components are Saccade, SmoothPursuit, Vergence, OKR, and VOR. The EyeMotionCenter agent is depicted in Figure 123. It works closely to a CameraHead agent which provides current motor positions to and receives motor commands from other agents. Details of the CameraHead agent will be discussed later in this chapter.

The Saccade, SmoothPursuit, and OKR acquire target information from a Visual

Input Representation(VIR) component. This component gathers all target information from the VisualAttention agent into one place for convenience (i.e. once the agent run on separated machine). Originally, these information are represented by VectorSignal representation.

The Saccade component obtains the target's position from the VIR component to perform saccades. The SmoothPursuit component requires the target's position and velocity from the VIR component. It also acquires motor positions from CameraHead agent to smooth motor movements. The OKR component needs successive image displacement to calculate a compensation of the head movement.

The Vergence component calculates disparity of left and right images from left and right image projection signal. It perform correlation on both signals using a correlation function from the library of an IntelSPL component.

The VOR component monitor the pan motor from the CameraHead agent and generates verge motor commands to compensate for the pan movement.

Each of these eye movement components calculate and provide motor commands for the EyeMotionCenter component. It then interprets these motor commands into the final motor commands needed to send to CameraHead agent in order to actually move the camera head.

Camera Head Controller Agent

The CameraHead agent manipulates the camera head hardware. The CatchIO component physically communicates with the camera head controller via RS-232. It provides the current reading of each motor to the VectorSignal representation. It also receives motor commands from other components/agents via the VectorSignal

representation. Because many components/agents can simultaneously access the camera head resource , The CameraHead agent is then designed to communicate with other components/agents through the MotionLink relationship. Figure 123 shows the CameraHead agent.

Appendix B

SOFTWARE AGENT SETUP FOR SACCADE EXPERIMENTS

In order to perform saccade experiments, the system needs to be rearranged such that it provides a proper environment for the saccade experiments. The current software architecture used in this work, however, allows the system to recon¯g itself to perform di®erent tasks without any software modi¯cation. The system uses state machine engine to execute each components' mechanisms. The saccade experiments are setup by just changing the state and state transition in the state machine engine. Using the available color segmenter component, three di®erent color models are created. The state machine system diagram is shown in Figure 124.

Single Step Pattern

The process starts from state 1 where the camera ¯xates on the maroon dot. After some delay time (i.e. timeout), the state of the system then changes to state 2 where the camera moves from the maroon dot to another target (green dot or orange dot). Finally, the state is reset to the initial state and the process starts over again.

Stair Case Pattern

The process starts from state 1 where the camera ¯xates on the maroon dot. Then the system changes to state 2 where the target moves to the green dot. After the time out which is set to be equal to an interstep time (¿), the target moves to its ¯nal position on the orange dot at state 3. Finally, the state is reset to the initial state

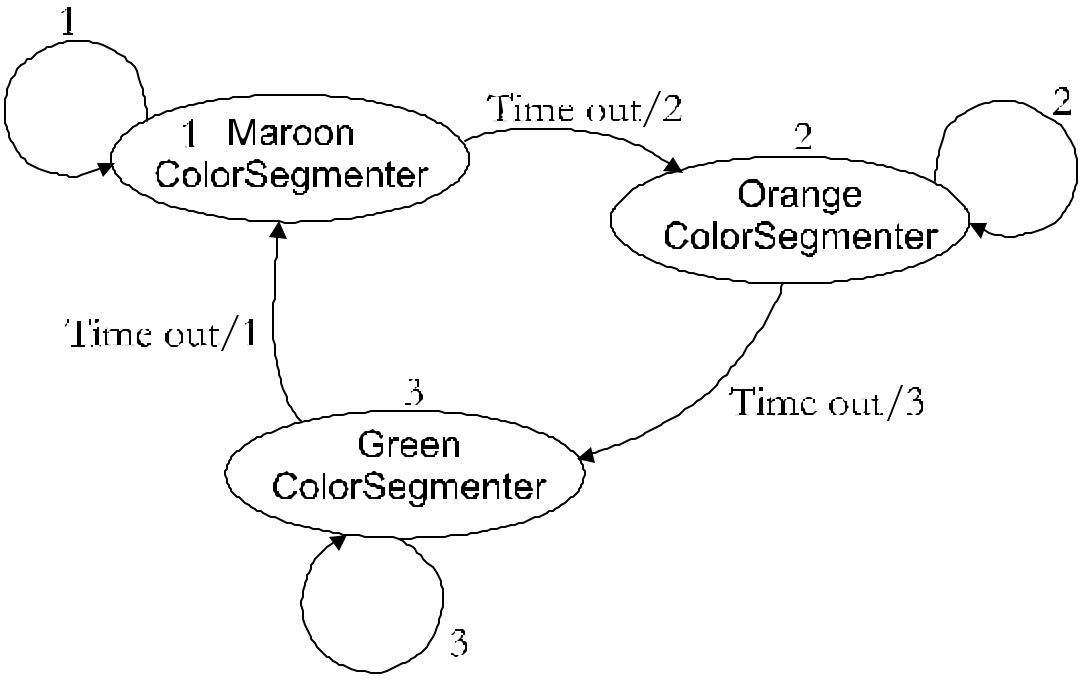


Figure 124: State machine diagram used in saccade experiments

and the process starts over again.

Pulse Undershoot Pattern

The process starts from state 1 where the camera ¯xates on the maroon dot. Then the system changes to state 3 where the target moves to the orange dot. After the time out which is set to be equal to an interstep time (¿), the target moves to its ¯nal position on the green dot at state 2. Finally, the state is reset to the initial state and the process starts over again.

Symmetrical Pulse Pattern

The process starts from state 1 where the camera ¯xates on the maroon dot. Then the system changes to state 3 where the target moves to the orange dot. After the

time out which is set to be equal to an interstep time (¿), the target moves back to its ¯nal position on the maroon dot at state 1. Finally, the state is reset to the initial state and the process starts over again.

Stair Case Pattern

The process starts from state 2 where the camera ¯xates on the green dot. Then the system changes to state 3 where the target moves to the orange dot. After the time out which is set to be equal to an interstep time (¿), the target moves to its ¯nal position on the maroon dot at state 1. Finally, the state is reset to the initial state and the process starts over again.