Tracking moving object by Stereo Vision Head with Vergence for Humanoid Robot.

by

Junji Yamato

B. Eng., University of Tokyo(1988)

M. Eng., University of Tokyo(1990)

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

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May, 1998

Signature of Author ....
Department of Electrical Engineering and Computer Science
May, 1998

Certified by ....
Rodney A. Brooks
Fujitsu Professor of Computer Science and Engineering
Thesis Supervisor

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

Tracking moving object by Stereo Vision Head with Vergence for Humanoid Robot.

by

Junji Yamato

Submitted to the Department of Electrical Engineering and Computer Science on May 20, 1998 in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis presents a stereo active vision system that is designed for a humanoid robot. The task was decomposed into three layers as saccade, tracking, and vergence. We adopted the developmental approach, which is based on a human infants developmental process, for the task decomposition. These three tasks were realized by the layered network of modules which were implemented on the DSP network. Simple motion detection is used for the cue for the saccadic eye movements. For tracking, searching most similar region based on Sum of Absolute Difference (SAD) was utilized. Template updating mechanism in tracking module made the tracking more stable. The vergence module utilized minimum SAD search for each pixel to obtain figure-ground separation in the 3D space. The vergence mechanism made the search efficient. The figure-ground separation was used for vergence control and it enabled to keep vergence more robustly. As a result of the combination of these modules and techniques, the system demonstrated real time tracking, vergence, and figure-ground separation in the real world.

Thesis Supervisor: Rodney A. Brooks
Title: Fujitsu Professor of Computer Science and Engineering

Acknowledgments

This thesis would not have been possible without the support of many people. First of all, I want to specially thank Rodney A. Brooks, the supervisor, for his advice and patience. Many thanks to all members of Cog group, Brian Scassellati, Matthew Marjanović, Charles C. Kemp, Robert Irie, Matthew Williamson, and Cynthia Ferrell, for enabling my work here. I was very fortunate to be able to work with them. Especially, thank you very much, Scaz and Robert, for reviewing my ugly draft.

Thanks to Mr & Mrs Sugiyama and Hitoshi Ono, for friendship and support. They made my life in Boston more fun and easier to go through tough time.

Thanks to several people at Nippon Telegraph and Telephone Corp. for giving me a chance to come here at MIT, and support for two years. Akira Tomono and Sakuichi Otsuka, at NTT Human Interface laboratories, Kenichiro Ishii, at NTT Basic Research Laboratories, Yukio Okazaki, and Hiroshi Maejima at personnel section of NTT.

Contents

List of Figures

Chapter 1

Introduction

1.1 Stereo Vision for Humanoid Robots

Computer vision research has a long history among many fields of Artificial Intelligence. Moreover, stereo vision is one of the classical problems in the computer vision field and large amount of research approaches have been followed.

The main topic of this thesis is an attempt to utilize stereo cameras to get depth information, but it is different from usual machine vision problems which have pragmatic goal, such as general object recognition or 3D reconstruction.

This research is a part of a project to build a humanoid robot called Cog[BS94]. The Cog project has an ambitious goal to build a human level intelligent robot and a novel methodology[BBI+98] to achieve it. Having monolithic internal models of the world is denied in this approach; thus the role of vision is also changed from getting general representation of the outer world to a more purposive and qualitative one.

1.1.1 Purpose

The goal of this thesis is to get depth information using stereo cameras and utilize it for the visual behaviors of the humanoid.

For humanoid robots, interaction with objects or humans are quite important to learn behavior. For such interaction, vision plays an important role in gathering information or in directing attention or gaze on the object or human. In this role, depth information is very important in two ways.

First, depth information enables the robot to achieve interaction in 3-D space. One of the tasks implemented on Cog is visually-guided pointing at an object[MSW96]. It has been implemented successfully by learning maps between camera image coordinates, eye motors coordinates, and arm motor coordinates. As the image coordinates are two dimensional, Cog can reach in the direction of the object, but cannot reach the position in 3-D space. By adding depth information, the robot will be able to touch the objects in 3-D space.

Second, depth information is a good information source for recognition of stationary objects. Cog now can distinguish objects only from motion information of successive images. This is enough for directing attention to a moving object as a target for reaching, but figure-ground separation is necessary for recognizing stationary objects. Figure-ground separation should utilize many kinds of visual information, including brightness, color, edge, and depth. To obtain depth information of the object, vergence information is beneficial.

1.1.2 Approach

The motto of the Cog project as a whole is to “mimic the human”, in terms of mechanical configurations, control structure, and even in developmental ways.

Cog have two arms which have six degrees of freedom(DOF), a neck with three DOF, and two eyes with a total of three DOF. The vision system has right and left eyes, and each side has wide angle and narrow angle video cameras, which mimics the high resolution area of the human visual field.

In this thesis, we used an active vision head. It has mostly the same mechanism and controls as Cog's vision system, but implements only the head. The scope of this thesis does not include behavior of arm, like reaching or touching objects, but only tracking a moving object, verging to it, and obtaining figure-ground separation. The figure-ground separation is then used to refine the vergence.

"Developmental approach"[Sca98b] is another keyword. The task decomposition and modularity of software were inspired by the development of human infant stereopsis. This methodology, developmental approach, provides good criteria for decomposing the task in an incremental way.

The active vision head is controlled by a parallel network of digital signal processors(DSP). On each node of the network, one functional module is running as a task which might have multiple threads. The nodes can communicate with each other by high speed links. This architecture is suitable for developmental implementation of the behaviors. Because the network structure enables the addition of functional modules easily to the existing network, and by adding the modules, the behavior can be extended in a developmental way.

1.2 Overview of this Thesis

The rest of this thesis is organized as follows: Chapter 2 describes, as the background of this thesis, the new methodology of building intelligence used in the Cog project, including some key points such as Embodiment and Developmental approach. Chapter 3 describes the development of stereopsis in infants and how this knowledge is applied to think about robotic vision systems. In chapter 4, the hardware and software platform used in this project and algorithms of the modules and the networks implemented are presented. This chapter explains the details of the saccade, tracking and vergence algorithms. Chapter 5 covers the experimental results and evaluation of the implemented networks. Chapter 6 presents the conclusions and future works.

Chapter 2

Background

It is the ultimate goal of Artificial Intelligence research to realize and understand human level intelligence. AI has a long history of these trials, but the framework of AI has been so affected by implicit assumptions that the world can be represented in some way, typically in symbolic forms, and that the manipulation of those symbol is the way human intelligence solves the problems. However, the limit of these assumptions about representation has been recognized[Bro91], and alternative methodologies are being explored.

In[BBI+98], the false assumptions of so called "classical" AI, were summarized as, the presence of monolithic internal models, the presence of monolithic control, and the existence of general purpose processing. These assumptions have been refuted by recent cognitive science and neuroscience.

Instead of the classical approach, some alternative concepts have been proposed[BBI+98, PSed]. According to [BBI+98], the alternative methodology has to emphasize some aspects of human intelligence, including **Develpment**, **Embodiment**, **Social Interaction**, and **Integration**. An embodied human-like form creature can be a good platform for pursuing research focusing on these aspects. The humanoid project is described next as a background.

2.1 Cog, The Humanoid Project

2.1.1 Overview

Cog is the name of the upper-torso humanoid robot being developed at the MIT AI lab. The main motivation of the Cog project is the hypothesis that human-like intelligence requires human-like interaction with world[BS94]. Cog is equipped with arms, a torso, and a head with visual and auditory sensory system. The arms have six degrees-of-freedom to mimic human arms, the vision system has independent pan axis for right and left cameras, and a joint tilt axis. Each side consists of two video cameras, one with a wide angle view, another with a high resolution telephoto lens. This combination mimics the feature of human visual field, which has high resolution only at the center.

2.1.2 Embodiment

Why is the human-like form important? Why is computer simulation not enough? Because an embodied human-like creatures can interact with humans in the real world in a natural way. If the robot has a human-like form, it is easy for humans to interact with it naturally and it enables learning or adaptation through a large number of interactions. It is not only for pragmatic reasons, but also includes subjective effects that humans feel that this interaction is natural and easy to do.

The simulation of humanoid is less natural and easy than a physical existence of robot in terms of communication with humans. The simulation also assumes that the representation of the world for the robot is possible. Furthermore, it is easier to utilize real conditions like gravity or friction than to simulate them on the computer.

2.2 Developmental Approach

The importance of building a humanoid as an "embodied" creature was described. Then, what should we do with the humanoid? What tasks should we implement on the humanoid in what environment? The answer is again "as humans do". The humanoid should do the task that humans do, in the environment that humans live. That sounds reasonable as an ultimate goal, but it is not realistic to achieve that at once. How can we achieve it gradually? What sub-goals should we set for the ultimate goal? The answer is "as humans do", again. That is called the "developmental approach"[Sca98b].

The developmental approach decomposes tasks into computationally simpler ones which can be achieved step by step. In decomposing the task into such stages, the development of a human infant gives a good guideline. Infants acquire skills sequentially and the order of onsets of skills are fixed. We can study the literature on infant development and developmental psychology, to find which tasks should be implemented earlier, which tasks should based on other ones, and how complex tasks can be decomposed into simpler ones.

It should be noted that this approach, imitating human infant development, is made possible because Cog has a human-like configuration. If it were far from human structure, task decomposition according to human infant development may not be appropriate to implement on Cog.

2.2.1 Advantages of the Developmental Approach

There are some advantages in the developmental approach. Three themes pointed out in [Sca98b] are following:

- **Development gives a structured decomposition**

In the developmental process, behaviors arise from the context that earlier behaviors provide. So the developmental study provides an insight into an ordered decomposition of complex task. A good structured decomposition must consist of small, simple sub-skills. The sub-skills should overlap enough to allow integration, but should not be too similar to duplicate effort. The decomposition is not necessarily a minimal or optimal set for the complex behavior, but a sufficient one at least.

- **Development facilitates learning**

  Embodied robotic systems should have the ability to learn or adapt to the real world. Decomposing complex tasks into small pieces enables the learning at each stage to be simpler and easier. It also gives insights into how previously gained behaviors may bootstrap more complex behaviors.

- **Development provides a gradual increase in complexity**

  Here, the word complexity is used with two meanings. Internal complexity and external world's complexity. In infant developmental process, the complexity of the external world is controlled by a caregiver. This improves the robustness of the learning process.

These points make the developmental approach quite suitable and attractive for building embodied and situated robotic systems in the real world. An example of application of this approach is described in the next section.

2.2.2 Example: Reaching Behavior

Visually guided reaching behavior is observed in infants at about 5 months of age. At this stage, infant can not not fully control their arm in its visual space. Infants always begin reaching from a rest position near their head. When they miss the target, they cannot move to the correct position from the missed position. They can only return to the rest position and try again.

This reaching behavior was implemented on Cog[MSW96]. Cog's arm has a six degree of freedom and the analytical solution for this behavior based on kinematics and dynamics of the arm becomes very complicated. Furthermore, such a solution is completely dependent on the configuration of the arm, camera, and head, and cannot adapt to changes in these configurations.

Instead of a classical robotics approach, the developmental approach provides an alternative task decomposition; first learn to foveate the visual target, then learn to orient to that target, and finally to reach for that target. The learning was decomposed into two stages. First, learn a saccade map that relates 2D coordinates in the camera image to eye motor control, and second, a ballistic map which relates gaze direction and arm control command to that location. If the robot has a computed map from 2D image coordinates to arm control command computed from kinematics and dynamics, it can directly move its hand to the arbitrary position of a visual stimulus. But this is not flexible, and unnecessary. A ballistic reach from a rest position along the direction of the gaze is sufficient to mimic infant reaching behavior at the first developmental stage.

Furthermore, this decomposition of learning maps allows the learning process to be autonomous. In learning the saccade map, Cog picks random positions in the visual field and saccades to the position using its current saccade map which is initialized linearly at the beginning. After the saccade, if the mapping is perfect, the center of the new visual field should be the target location. But there is usually some error. To measure this error, the best correlated postion to the original target image patch around the new center is located. The search is based on the dot product of the image patches as vectors. The offset of the most similar patch and the center of the new visual field is used as an error signal to update the map. This learning process can be repeated auotonomously until the error becomes small enough. In the experiment of[MSW96], an accurate saccade map was obtained after 2000 iterations. The second mapping, a ballistic map, which relates gaze direction to arm control, is also learned autonomously. Again, Cog can obtain an error signals by itself. After each attempt to reach, the robot can examine how close the hand is located to the center of the visual field through its own vision. This offset gives an error signal to update the ballistic map. To simplify the six DOF arm, the arm configuration is represented by the combination of three primitive position[Wil96]. This dimensionality reduction makes the mapping problem simpler.

It is complicated and inflexible to make an accurate and complete hand-eye system based on classical robotic approaches, computing kinematics and dynamics. It is also more than sufficient, because it enables general description of 3D space and direct correction of reaching which human infants do not do. Adopting a developmental approach, instead of it, gives the decomposition of the task into small stages and the learning at each stage can be autonomous.

In this chapter, the background of this research was described and the advantage of the developmental approach was explained. In the next chapter, we show how the developmental approach can be applied to stereo vision.

Chapter 3

A Developmental Approach to Stereo Vision

We adopt the developmental approach to study the task and modularity of stereo active vision. To decide what configuration of stereo tracking system should be used for the humanoid, one good criteria is to mimic the human visual system, because the purpose of vision system of humanoid is to achieve human level behavior. To get knowledge of the configuration of human stereo or tracking system, we examine the development of stereo vision of infants. This chapter describes some facts about the development of visual perception in infants, and the criteria derived from these facts for implementing a stereo tracking system.

3.1 Infant Development of Eye Alignment and Stereopsis

3.1.1 Depth Perception

How are two dimensional stimuli on the retina transformed into a three dimensional perception? One theoretical approach, called cue theory[Gol96], explains this mechanism. The cue approach looks for the connections between stimuli in the environment, the images these stimuli create on the retina, and perception of depth. According to cue theory, we learn the connection between the cue and depth through our experience with the environment. After this learning has occurred, the connection becomes automatic; after the association has been established, the depth cue causes three dimensional experience directly and solely. There are four types of cues:

- **Oculomotor cues**
  They are based on our ability to sense the position of our eyes, for example, convergence angle and accommodation.

- **Pictorial cues**
  What can be depicted in a still picture, for example, overlap, texture gradient, and linear perspective.

- **Movement-produced cues**
  They are created by the movement of the observer or object, for example, motion parallax.

- **Binocular disparity**
  Depth perception created by the difference of images formed on the left and right retinas.

In the developmental process of the human infant, binocular disparity becomes effective early and the pictorial cues becomes effective later. For stereopsis of the humanoid robot, the first step is to implement binocularity cues. Later pictorial cues should be associated, hopefully through learning. The next section focuses on the development of binocular disparity.

3.1.2 Development of Eye Alignment, Vergence, and Sensory Binocularity

For the operation of binocular disparity, binocular fixation must be accomplished. In the developmental process of the human infants, binocular disparity is decomposed into three different aspects: alignment of eyes, convergence, and sensory binocularity[Hel91, TGC+94]. Development of these features are measured and compared in [TGC+94]. Ocular alignment and convergence can be measured by a human examiner using a standard test procedures. Sensory binocularity can be measured using a fusion-versus-rivalry preferential looking technique. This technique uses two sets of gratings, fusible and rivalry pairs. The fusible pair are both vertical gratings and rivalry pair consist of one horizontal and one vertical gratings. Infants who have sensory binocularity are supposed to look at the fusible pair more often. The number of trials in which infants looked fusible pair is considered to be the measure of the development of the sensory binocularity

The result of their experiment showed that the development of these three aspects occurs rather suddenly at specific ages. For the relation of ocular alignment and convergence, at less than 6 weeks of age, most infants showed orthotropia, but none showed full convergence. The age at which 50% of the infants demonstrated full convergence was 11.9 weeks. These two developmental processes showed no specific relation. On the other hand, both sensory binocularity and convergence, in other words, the motor binocularity, develop at a similar age. The onset of sensory binocularity occurred at 12.8 weeks, and full convergence 13.7 weeks. Furthermore, the cumulative proportion of infants showing the onset of sensory binocularity and convergence were almost on the same curve.

From these experimental results, [TGC+94] concluded that ocular alignment did not require the development of sensory binocularity and the development of sensory binocularity did not wait for the onset of good ocular alignment. In other words, the onset of convergence and sensory binocularity were neither the cause nor the result of good eye alignment. On the other hand, the close link between convergence and sensory binocularity onsets indicates a common causal factor or common mechanisms.

These observations give some hints about the coordination of single eye tracking and binocular vergence.

3.2 Task Decomposition

The overall task in this thesis is tracking an object using a stereo active vision head. In the implementation of this task on the active vision head, ocular alignment corresponds to a single eye tracking of the object or attention to the object. Sensory binocularity corresponds to the stereo matching of left and right images to generate depth information. Convergence is the vergence angle control of right and left cameras. Though each feature of the active vision head are closely related, the relation to infant development gives some guidelines for designing a modular active vision controller.

- monocular alignment should be achieved without vergence

  This suggests that the attention and tracking capabilities have to work without vergence control or stereo depth perception modules. This does not mean depth perception does not help the tracking or alignment, but only that they should work even roughly without stereopsis.

- binocular disparity module should be independent from monocular alignment

  This suggests that the stereo module have an independent mechanism from alignment of right and left eyes. If right and left eyes can independently align to the same object, and if the target points are the same, then vergence is realized just as a result of the alignment of the eyes. This is mechanically possible but does not match observed infant development. Therefore an independent binocular disparity module should be implemented on the active vision system.

- vergence control and stereo vision are closely connected

  Sensory binocularity and the motor binocularity, which is vergence control, share much information and the implementation in which one software module deals with both feature can be justified.

Based on the considerations above, we first decompose the overall task into two classes. One is to track and focus attention on an object with one eye, and the second is tracking of the object with two eyes which have vergence. The first task is also decomposed into two stages. Saccadic motion to some cue of interesting object and tracking of the object, because both types of eye movement are observed in infants. Then, there are three levels of the task, as a result. Each task should be implemented in the form of an incremental network of modules. The task and its controller should be decomposed as follows:

- **saccade to cues**

  This level of controller first detects a cue, and then directs the gaze to that location. This change in gaze direction requires some time, and corresponds to a saccadic motion of human eyes. At this level, the control system does not need to have any memory, it needs only to react to the environment.

- **saccade and tracking**

  In this level, the system has a sort of memory for the target and can maintain its gaze on it. This level corresponds to ocular alignment but without any depth cues. Only 2D information is utilized and both eyes of the vision head move in the same way.

- **saccade, tracking, and vergence**

  Now the system has both vergence and sensory stereopsis. At this level, it can detect the differences in depth between the target and background. The output of the system is not only its behavior to track the target, but also the figure-ground separation of the target. This separation should be re-used to refine the vergence control. This feature corresponds to the close connection between sensory and motor binocularity of infants.

All three levels of the task and the corresponding control network are designed incrementally. Higher levels are added to the lower levels, though added layers can also be tested in isolation. This structure is expected to have robustness and manageability[Bro86]. The details of the proposed layered tasks and controller implemented on the active vision head are described in the next chapter.

3.3 Comparison to Related Work

3.3.1 Vision Related to Cog

There have been some computer vision research related to the Cog project. Wessler[Wes95] built *Reubens* with three main visual modules: tracking, saccades, and calibration. This modularity and their connections enabled saccadic tracking behavior. When it failed to maintain an attention window, it realized the failure and saccade module found another target region for attention. The important point is to recognize when the tracking fails. On *Reubens*, stereo and vergence were not implemented, but this modularity of tracking and saccade can be utilized also for the stereo head in this thesis.

Kemp[Kem97] implemented a loosely connected tracking algorithm on stereo head and it naturally achieved vergence. It does not have an independent binocularity module, but instead maintains a tracking template for both eyes mostly common. Each template is updated by weighted averaging of each new image patches with the previous template, and the template of the dominant eye is propagated to the secondary eye. The propagated template is used to keep the template similar to the one for the dominant eye, by averaging it and the template being kept for secondary eye. This process maintains the similarity of the two templates, and assures that the two eyes look at the same object.

Marjanović et.al. demonstrated a self-taught reaching behavior on Cog[MSW96], as described in the previous chapter. In the learning of the behavior, motion detection played an important role. This feature was implemented using a simple image difference and region growing technique. The motion detection module was based on the same algorithm was used for this thesis.

One of the most important skills acquired by infants in the developmental process is shared attention[Sca98d]. To realize it on Cog, Scassellati implemented human face and eye detection on the active vision head[Sca98c]. This thesis use the software environment from his research, including some modules.

3.3.2 Comparison to Other Approaches

The approach described above is quite efficient, compared to usual stereo vision systems which have neither attention nor vergence.

Kanade et al. developed a video-rate stereo machine equipped with six fixed(no vergence) video cameras[KKK+95]. The key algorithm is "multi-base line stereo" which combines the output of similarity from pairs of the cameras along depth. This reduces the ambiguity of the stereo matching. The purpose of the CMU stereo machine is to obtain a dense depth map, which is different from our purpose. It does not have attention to any point in the view and obtains the depth over the image field. In our case, we do not need such a dense depth map, for the behavior of the humanoid. Instead, the stereopsis should assist in directing attention to the object whose depth is different from the background, and the vergence should make it more efficient.

Because of vergence, we can restrict the search for stereo matching to a small range. The search range can be only a few pixels around the point of vergence distance in order to distinguish the object from the background. This small range of search also reduces the ambiguity of matching because matching candidates out of that range are eliminated. This illustrates the merit of having vergence in stereo vision systems.

Having vergence is good. Then, how should it relate to other features? Coombs's binocular head[CB93], which has vergence control, realized a smooth pursuit of object. The head can follow a moving object by smooth pursuit, utilizing an edge-based zero disparity filter(ZDF) and PID control of velocity and position. Its structure is the reverse of our proposed layered controller. Images obtained by the right and left cameras are smoothed by a Gaussian filter and then vertical edges are detected. The ZDF output of the two edge images is thresholded and the target for pursuit is selected. Here, the ZDF is a logical AND operation for vertical edge images, which assumes good vergence maintenance. As a result of this configuration, the system does not track an image from a single camera, instead it tracks the output of the ZDF. This means each single eye does not have any tracking ability, and tracking is dependent on the vergence control. When vergence control fails, it also loses tracking. Although the algorithm was sufficient to realize a smooth pursuit ability, it is not suitable for implementing tracking and vergence developmentally.

Another concern is symmetry between the right and left eyes. Some stereo vision heads with vergence have a completely symmetric implementations[CB93, PUE96]. However, from the viewpoint of mimicking human visual system, having a dominant eye might be helpful to extend the visual behavior.

Kemp's implementation[Kem97] has a dominant eye but it does not have a complete master-slave relation. The template for tracking with a secondary eye is kept similar to the one from the dominant eye, by blending the template from the master with the one for the slave eye. This sounds like a good mid-point between a symmetric implementation and a master-slave relation. However, in this thesis, right and left eyes have separate roles. The right side tracks the target and the left side is concentrated on vergence, because it makes adding vergence layer to tracking easy. This problem, how right and left eyes should share the roles remains for future work.

Chapter 4

Implemented Modules and Networks

This chapter describes the hardware and software environment in which the networks of modules described in the previous chapter were implemented. Then, the algorithms for each module and the structure of the networks are presented.

The hardware consists of mechanical, electrical, and computer components. The software is written in parallel C, a multitasking version of C produced by 3L, running on a network of TI C40 DSPs. Detailed information is given in the following section.

4.1 Active Vision System

4.1.1 Hardware

The the active vision head is shown in **Fig.4.1**. The system was designed to mimic the human visual system[Sca98a]. It was constructed by Brian Scassellati, Cynthia Ferrell, Milton Wong and Elmer Lee as a part of the Cog project.

The active vision head has four degrees of freedom(DOF). Each eye can pan independently, and they tilt together. The head also can rotate around a neck, but the neck was fixed in this research and only three degrees of freedom were used. The range of pan is approximtely 120 degrees, and tilt is approximately 60 degrees. The size of the head is about 25 cm width and 10 cm height for the eye part.

The system has four cameras. A pair of wide angle camera and fovea camera is mounted on each side. This provides both wide angle view and high resolution area on the center of the view. This is to mimic the human visual system, which has low resolution peripheral view and high resolution fovea. Each camera is a Chinon CX-062 color CCD. It has 1/3 inch color CCD, and small camera head board which weighs only 20 grams and is connected to main board by 3 inch cable. For wide view, a 3mm lens was used, and for the fovea camera, an 11mm lens was used.

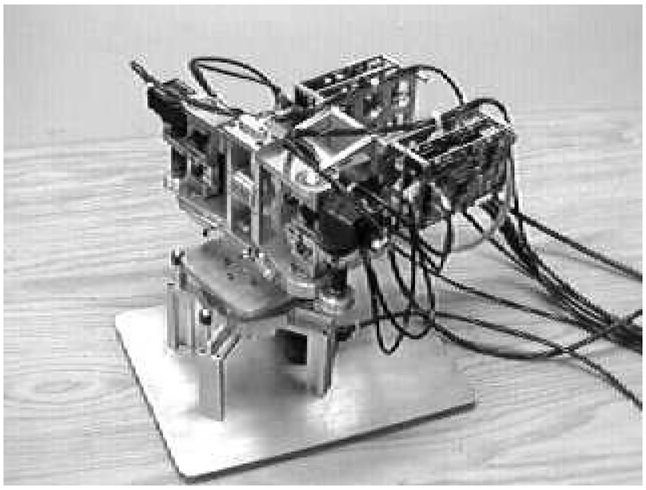


Figure 4.1: Active vision head used in this thesis

As the purpose of the active vision system is to mimic the human system, the requirements of movement are also defined by human eye motion speed. The motor system is designed to achieve three 120 degree pan saccades per second and three 60 degree tilt saccades per seconds. To meet these requirements, Maxon 12 Volt, 2.5 Watt motors with 16.58:1 reduction planetary gearboxes were selected for the pan axes. To control the each motor, an HP HEDS-5500 optical shaft encoder was fitted on the axis. It outputs 1024 counts per revolution, and the resolution was modified for each axis by changing the size of spindle on the cable, into 8.5 encoder ticks/degree for the pan axis, and 17 encoder ticks/degree for the tilt axis. All motors are controlled by a Motion Engineering Inc.'s LC/DSP-400 motor controller board which is attached to the host PC via the ISA bus. The controller maintained a 1.25 kHz servo loop at 16 bits of resolution.

The video streams from four cameras are captured and processed by the DSP network. It is parallel network architecture base on the TIM-40 standard for the Texas Instruments TMS320C40 digital signal processor. There are several commercial DSP modules based on the TIM-40 standard, including video capture and image processing. Each module has high-speed bi-directional hardware links which are called "comports". The DSP modules are connected by a commercial backplane, and the entire network is connected to a host PC through an ISA interface card.

We used four types of DSP module. One is a general C40 processor with no special feature. This type was used as a "ROOT" processor and additional general processor which is labeled as "P2" in **Fig.4.2**. The "ROOT" processor is the only module which can communicate with host PC. The second type of module is called VIPTIM, Visual Information Processor, which has a hardware convolution feature. This module is used for the *Motion detection module* and the *Tracking module* which are described in following sections. The third type is called AGD, Accelerated Graphics Display, which has hardware to display images on a VGA monitor and is used for the *Display module* . The fourth type is called Grabber. It has hardware to capture composite or a Y/C separate video signal and digitize it. One grabber DSP module is used for each *Grabber module* in the network.

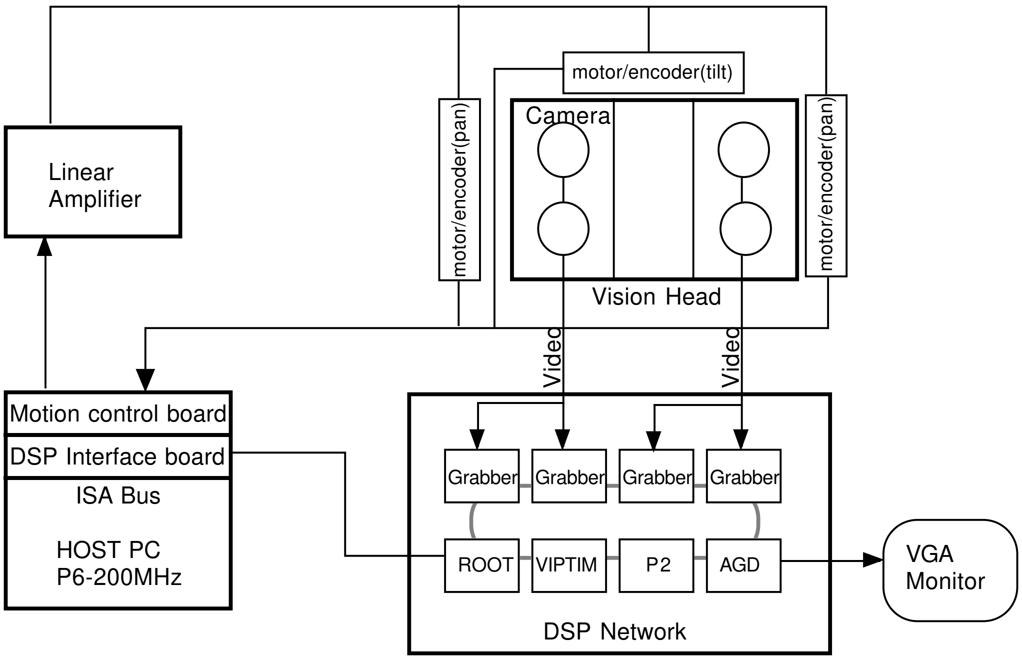


Figure 4.2: Active Vision System configuration

The host is a PentiumPro 200MHz PC which has an ISA interface card for motor control and for communication with the DSP network. The outline of the configuration of the system is shown in **Fig.4.2**.

4.1.2 Software

The software for controlling the vision head, including capturing video, processing video image frames, and sending control signals for motor control, is written in *Parallel C* by 3L. It is a multi-threading C library and runtime system. All source code is written on the host PC and compiled, and then configured for the C40s and is downloaded to the network to be executed. *Parallel C* provides "virtual channels" for inter-module communication. This means modules which do not have physical connections can communicate in the same manner as those are physically connected. Software can treat them identically.

4.2 Description of Modules

This section describes the basic algorithms and interfaces of each module. Each module is implemented as a task of a DSP node. Each task has one or more threads running on one C40 module. The whole control system implemented in this thesis is shown in **Fig.4.3**. Each level corresponds to the decomposed task. Level 1 is for saccadic motion, level 2 is for tracking, and level 3 is for vergence. As shown in this figure, two cameras were used in the network. We used two wide view cameras for stereo vision, although there are two narrow view cameras on both right and left side. The coordination of wide and narrow view cameras is an important future work. We describe the modules used in the network and layers of network which is a part of the whole system.

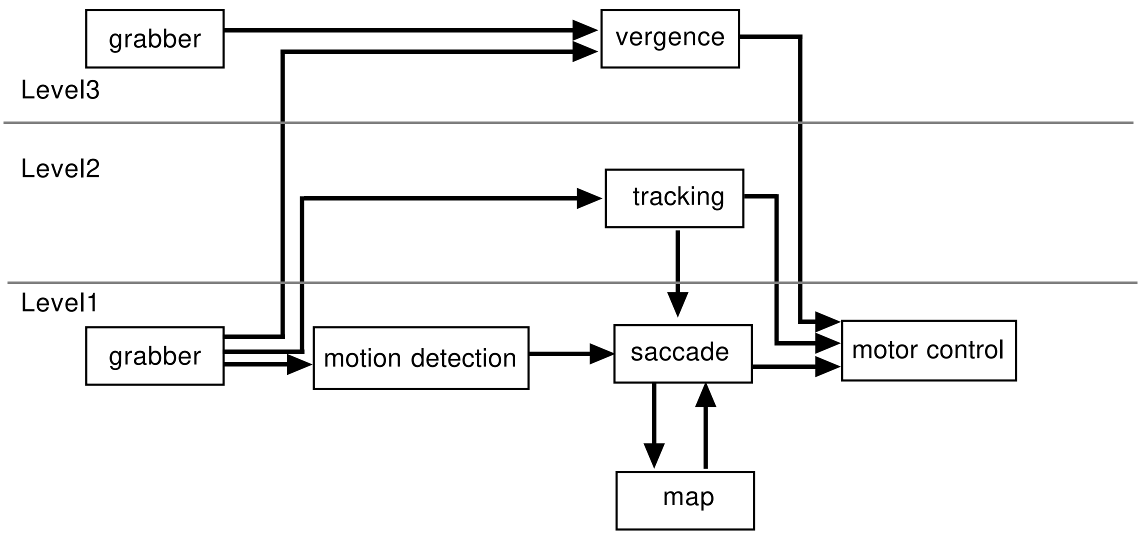


Figure 4.3: Layered network for controlling active vision head

4.2.1 Motion Detection Module

This module is one of the two key modules used to implement saccadic eye movement. The input of this module is one video stream, and the output is a position where the largest moving region exists. The algorithm is based on simple image difference. First, it takes the difference between two successive frames of the video. Then the difference image is thresholded, and the motion region is extracted. The motion region is then labeled and the centroid of the largest one is sent out as a detected motion region. Though the input video signal is NTSC color, the calculation of image difference is based on the gray scale image because that is what human motion detection operates on primarily.

4.2.2 Saccade to Motion Module

This module transforms two dimensional location information from motion detection module into a motor command. The transformation is based on the mapping obtained by learning[MSW96]. The learned mapping is kept by the mapping module and it communicates with this saccade to motion module. This module has another input, a suppression signal from the upper layer module. When suppression is ON, the output to motor control command is prohibited to make the control from upper layer valid. This input port is used only when there is an upper level in the network.

Fig.4.4 shows the configuration of the network for saccade to motion. This illustrates how these modules are used, and it is a implementation for level1 task. The motion detection module and saccade to motion module were built by Scassellati[MSW96] for detecting hand in learning process of Cog's reaching behavior.

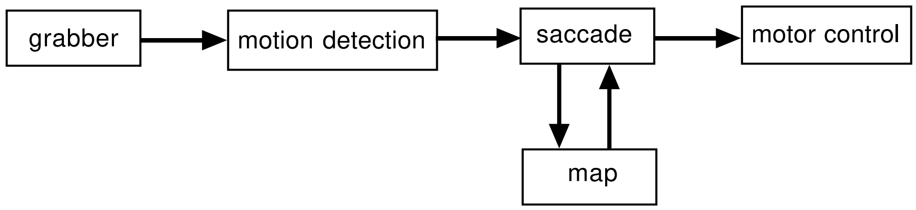


Figure 4.4: Network for saccade motion

4.2.3 Tracking Module

The tracking module takes video as input, keeps a template for matching, compares the input video with the template, and sends out motor control commands to keep the most correlated region at the center of the view(**Fig.4.5**). The velocity of motor is controlled proportionally to the distance to the most correlated region.

The matching is based on the sum of the absolute difference(SAD). It finds the x_0 and y_0 which minimize:

$$\min_{x_0,y_0} \sum_{x,y} |I(x_0 + x, y_0 + y) - T(x, y)|$$

as a new target location, where $I(x, y)$ is new frame of image, and T(x,y) is the template. This main portion of tracking module was built by Scassellati.

Updating of the template is an issue for this module. If the template is fixed, the tracking module cannot adapt to changes of environment and target object, for example, illumination change during the tracking, change of posture or size of target object. However, if the template is updated every time it gets a new image frame, the error is easily accumulated and the template loses the target quickly. To keep the consistency to some extent, and at the same time, to adapt to the change of the object and environment, template updating control is important. Simple linear interpolation is used to update the template[Kem97]. The ratio for blending an old template with a newer acquired target region is empirically determined. In

$$T_{new}(x, y) = \frac{\alpha T_{old}(x, y) + I(x_0 + x, y_0 + y)}{\alpha + 1}$$

where T_{new} is the new template and T_{old} is the old template, 10 was used for α in this thesis. It depends on the speed of the change of target and frame rate of the system, and was empirically decided through some trials.

An important feature of the target module is self detection of its status. When tracking module loses the target, it should be aware of it. The GOOD or LOST signal is used for coordination with other modules which also send motor commands. The tracker module detects its failure using the SAD value and its change. The conditions are following:

- SAD value is more than threshold
- SAD became large rapidly

When one of the conditions above is true, the tracker module thinks that the target is LOST and it sends out signal to other module.

It is also important to initialize the template, at the beginning of the tracking, and during recovery from the LOST status. It is hard to define an initialization process suitable for general usage, but in the network which the tracker is used in this thesis, it captures an image patch at the center of the image field as a new target region, after a fixed delay, allowing the lower level to successfully locate something moving during that period. These algorithms of the tracking module are illustrated in in **Fig.4.6**.

When the tracking module fails to detect LOST status or fails to capture moving region as a target, it mostly continue to look at somewhere in the background. To escape from this status, the “bored” signal was introduced. The bored counter runs while the target is not moving and when the duration becomes more than pre-determined threshold, the module sends out bored signal which is treated the same as a lost signal.

Figure 4.5: Network for tracking

4.2.4 Vergence Module

The vergence module takes video from the right and left cameras, compares them, and sends out motor control for the vergence angle. It controls only the slave side eye motor. The original image SAD based vergence module was built by Scassellati, and it was used as a base of development. The captured images are convolved by Laplacian of Gaussian (LoG) filter and transformed into edge images. The size of filter was decided empirically. The σ of the Gaussian was 1.7(pixels) for 64x64 pixel images. This is suitable size for handy size objects at an arm distance from the vision head. The kernel used for convolution was 9x9 which is large enough to cover the region in which the kernel has a values large enough. After LoG-filtering, the target region which is small image patch of master camera, is compared to the search region in the slave camera image(**Fig.4.8**). This is to find:

$$\min_{x_0} \sum_{x,y} |LoG(Targ(x,y)) - LoG(S(x+x_0,y))|,$$

where $Targ(x, y)$ is target image and $S(x, y)$ is search image. This matching is done by 1/4 pixel wise by linear interpolation. For the y-axis, the fixed offset value is pre-determined to make the matching possible within one pixel, however, less than 1/4 pixel size offset is calculated and used in the disparity map described below. The x_0 is used to calculate motor control signal to the slave motor.

To detect the qualitative depth in the target region, the map of disparity is built using minimum of sum of absolute difference between two image patches. The disparity map $D(x, y)$ is calculated by

$$\begin{aligned} D(x,y) = arg\min_{x'} \quad \textstyle\sum_{\Delta x \Delta y \in patch} \quad & |LoG(Targ(x + \Delta x, y + \Delta y)) \\ & -LoG(S(x + x_0 + x' + \Delta x, y + \Delta y + y_{offset}))|, \end{aligned}$$

where the patch is (x, y) centered 5x5 pixel area, and y_{offset} is less than one pixel offset, which is same for one map $D(x, y)$ but adjusted every frame. The y_{offset} is measured every time by 1/4 size of pixel to minimize the SAD of whole target region. This disparity map has zero value at the depth of verged point, and smaller value indicates nearer. Because the vergence is controlled and the purpose is basic figure-ground separation, the range of x' can be small, as mentioned in the previous chapter. In this implementation, ± 2 pixels was used. This map is transformed into a weight map for vergence using this formula:

$$Weight(x, y) = 1 - \beta |D(x, y)|,$$

where β should determined so that the weight does not become minus even for largest value of $|D(x, y)|$. In the implemented vergence module, β was 0.4, while maximum $|D(x, y)|$ was 2. At the verged point, the weight has largest value. Using this weight, the vergence function is modified as:

$$\min_{x_0} \sum_{x,y} Weight(x,y)|LoG(Targ(x,y)) - LoG(S(x+x_0,y))|,$$

The weight map helps to keep attention on the verged target object by ignoring background region. When the background area in the target region has a strong contrast, especially vertical edges, the minimum SAD happens to drop at the depth for the background, not for a object. This effect is demonstrated in the next chapter, using examples. This feature corresponds to the close connection between sensory and motor binocularity mentioned in the infant developmental process. Sensory and motor binocularity work together and both onset happens relatively suddenly at the same age.

In this module, LoG filtered images are used to avoid the effect of intensity differences between images. In the tracking module, the sum of absolute difference of intensity images worked, because those are captured by the same camera, but here, the two images are from different cameras and intensity values are hard to calibrate between cameras. For the vergence control, ZDF using logical AND of two vertical edge[CB93] was also tested, but it was easy to match false edges in a cluttered environment. SAD of LoG filtered image was more robust in such environments.

4.2.5 Other Modules

There are some other modules, including a *Grabber module* and *Display module* which are important for the implementation of the active vision system. The *Grabber module* captures video and averages the signal into an appropriate size and color. Digitized video is transmitted to connected modules via comports. In this research, all images are scaled to 64 by 64 pixel 24bit color images, in order to maintain real time performance. One module can capture one stream of video signal, and we employed four *Grabber modules* in the DSP network. However, only two *Grabber modules* were used to capture two wide angle video cameras. *Display module* drives the VGA monitor to display image, graphics, and text from other modules. This is quite useful for debugging in development.

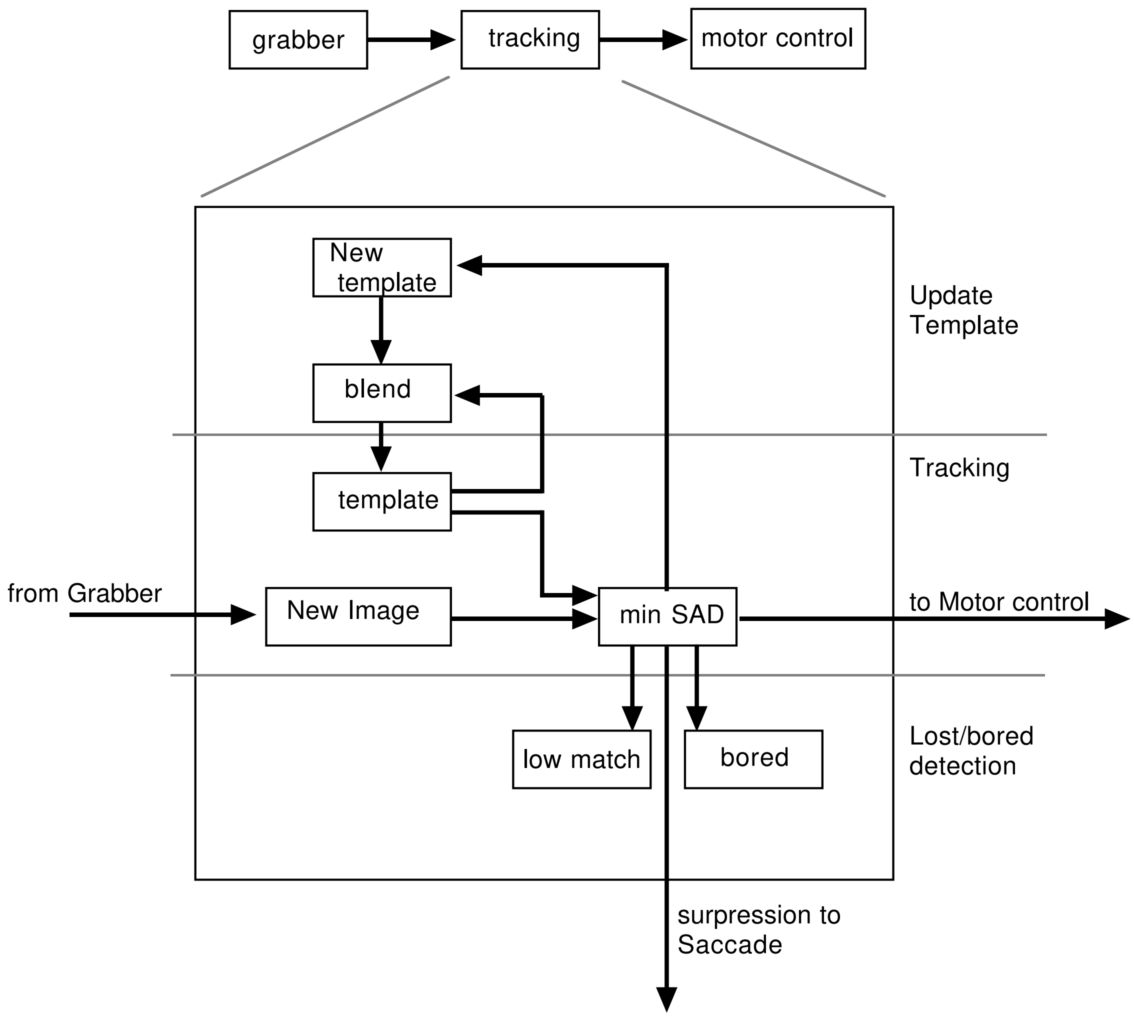


Figure 4.6: Tracking module

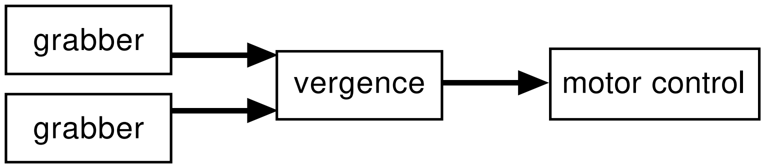


Figure 4.7: Network for vergence

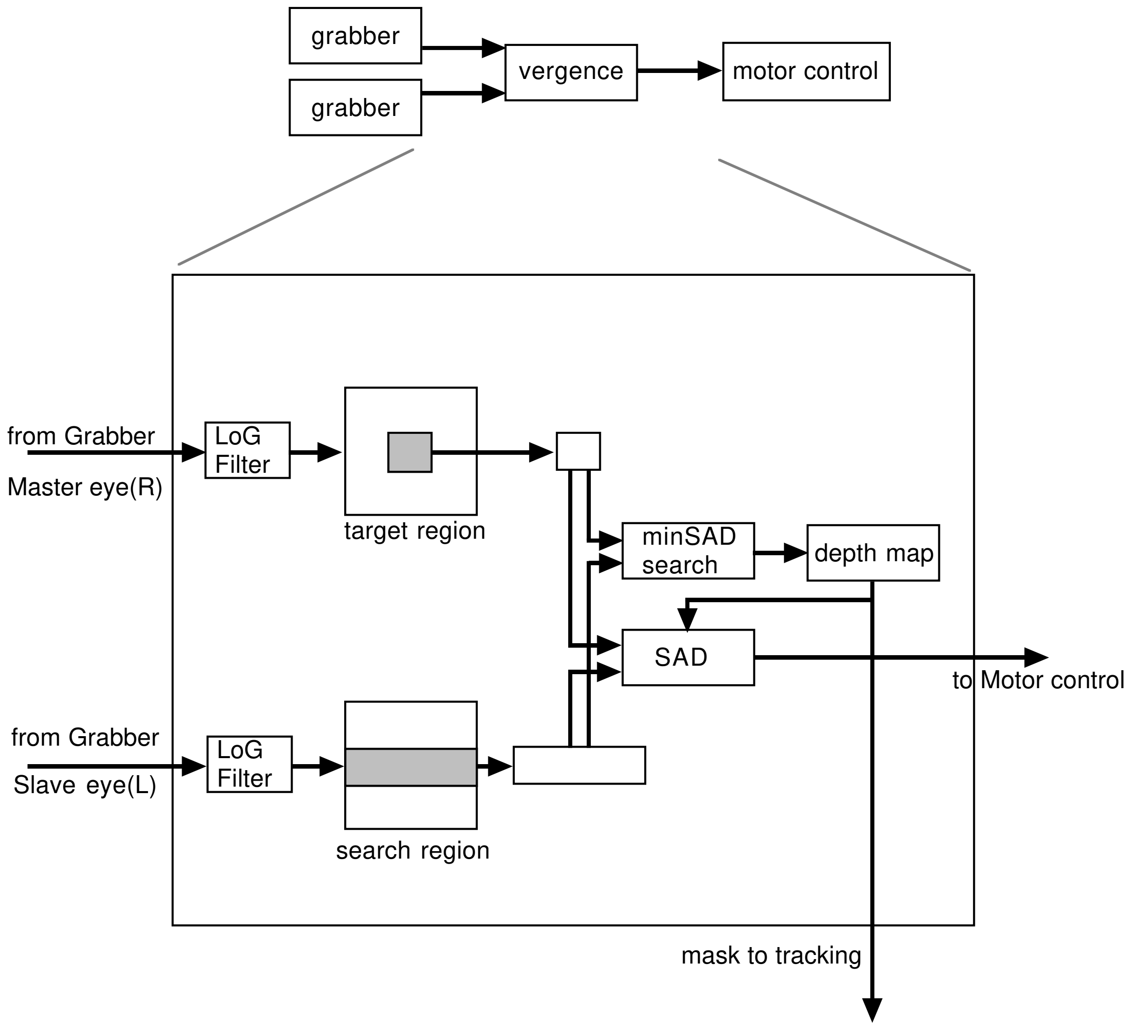


Figure 4.8: Vergence module

4.3 Description of Networks

This section illustrates networks for each level of task, from level 1 to level 3. Each network is incrementally built on the level 1 network which was described in the Saccade to Motion Module section.

4.3.1 Level1:S Saccade to Motion

As described before, it consists of grabber, motion detection, saccade, map, and motor control modules(**Fig.4.4**) .

4.3.2 Level2:ST Saccade and Tracking

This network is built by adding a tracking module to the level 1 network, as shown in **Fig.4.9**. While the tracking is working well, the tracking module suppresses the motor control from saccade to motor control module, and tracking takes control. When the tracker decides it has lost the target, it stops suppressing, and saccade takes control. After the eyes successfully capture the moving region, which is considered to be something interesting to track, the tracker regains control. This network sometimes slips to the background during tracking and did not detect LOST, and sometimes captures the background after recovering from LOST status. To avoid these situations, a control timer is introduced. Timer counts a no movement period, and if there is no movement over a pre-determined threshold, it becomes "bored" with the current target and behave as if it were LOST.

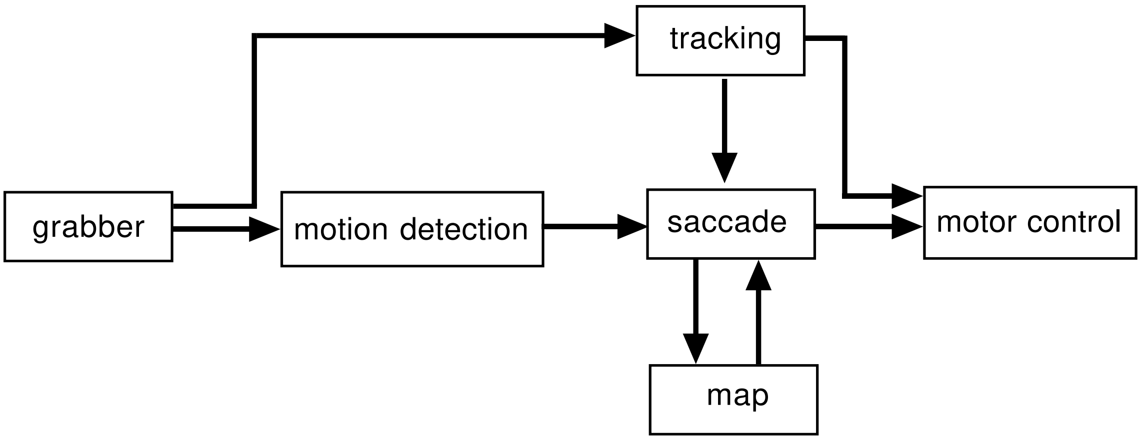


Figure 4.9: Network for Saccade and Tracking

4.3.3 Level3:STV Saccade and Tracking with Vergence

At the third level, another grabber module for the slave eye and vergence module were added on the level 2 network, as illustrated in **Fig.4.10**. This network enables the vision head to verge to an object in 3D space and to separate it from the background. The vergence module controls only the slave eye's pan motor and does not influence the master eye's pan motor or tilt motor, so no additional arbitration is needed. The depth map used as a weighted mask for SAD calculation can also be applied to the matching of the tracking template. However, this connection was not implemented because the transmission between modules takes times and reduces system speed. The utilization of the figure-ground separation remains for future work.

In this chapter, the hardware and software environments and the algorithms of modules and networks implemented in these environments were described. The modularity of tasks and their implementation inspired from infant development proved a good organization to implement on the DSP network. The evaluation

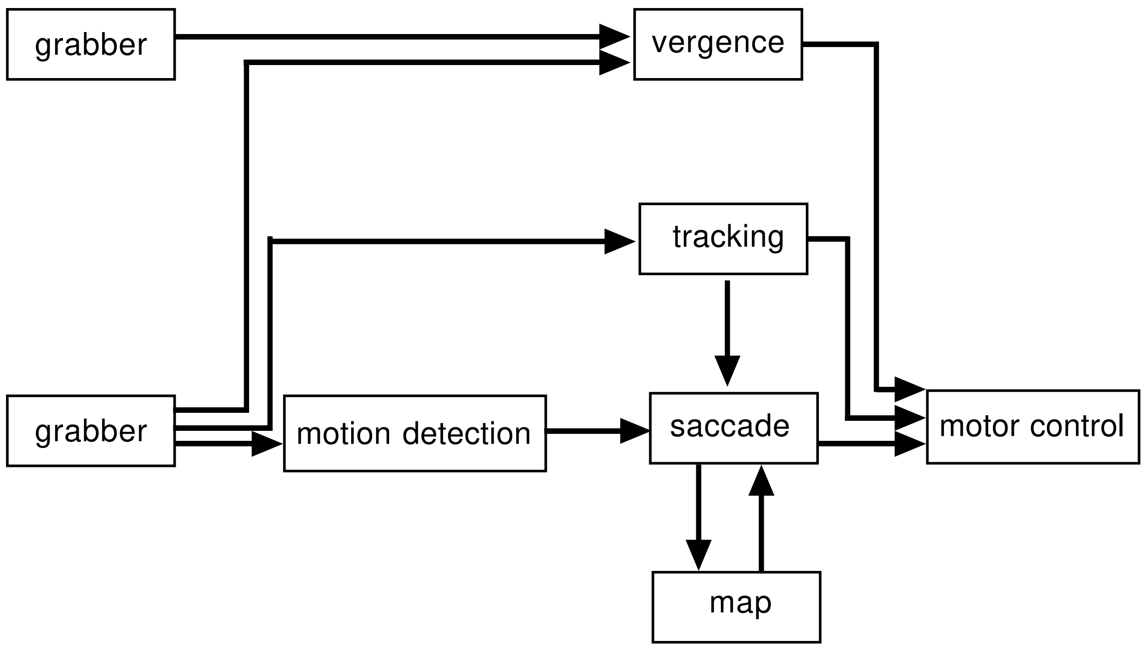


Figure 4.10: Network for Saccade and Tracking with vergence

of these networks are in the next chapter.

Chapter 5

Evaluation

This chapter describes the tracking and vergence behavior of the tracking system implemented in this thesis, showing some images taken by the active vision system. All images and figure-ground separation map were written by an additional routine of the system and it caused a little slow down to the system. So the behavior is possibly slightly different from the one without such image writing.

5.1 Level1:S

The Level 1 network for saccadic eye movement detected the motion cue and then directed the gaze to the location. During the saccadic eye movements, the motion detection module still sends out the location of the motion, however the motion is caused by the eye movement and the saccade module has to ignore it to avoid detecting false motion cues. The interval time is adjusted to 300msec to ensure no detection of such a false motion cue.

5.2 Level2:ST

The Level 2 network for saccade plus tracking successfully tracked the target object, for example a white mug, a human hand, and a textured circular disk.

When the system loses the target because of a sudden change of the target appearance, it locates to something moving. This process sometimes fails to capture the target and the gaze gets stuck somewhere in the background. Getting the gaze on the target again requires some intentional movement, and it is not so naturally done to switch saccade and tracking. However, even when the gaze fixes on to the background region, it soon becomes bored and tries to find another moving object, because the background does not move.

The template updating mechanism successfully worked. **Fig.5.1** shows the images captured while the system was tracking the textured circular disk. During the tracking, the disk rotates slightly, but the system could follow it. The template used at each time is shown in the lower row of the figure. The frame rate of the system was about 15 frame/sec and the frames shown in the figure are one from every five frames. The effect of the template blending is shown in **Fig.5.2**. The target of object is rotated by the hand. The target regions taken at each new frame are shown in the lowest row in the figure, and the resulting templates generated by the blending are in the middle row. The template follows the change of the target slowly by blending the new image to the previous one.

The template updating mechanism is suitable for objects which change their appearance during a change in orientation, like human faces. As demonstrated in **Fig.5.3**, the system successfully followed a face while it changed orientation and appearance.

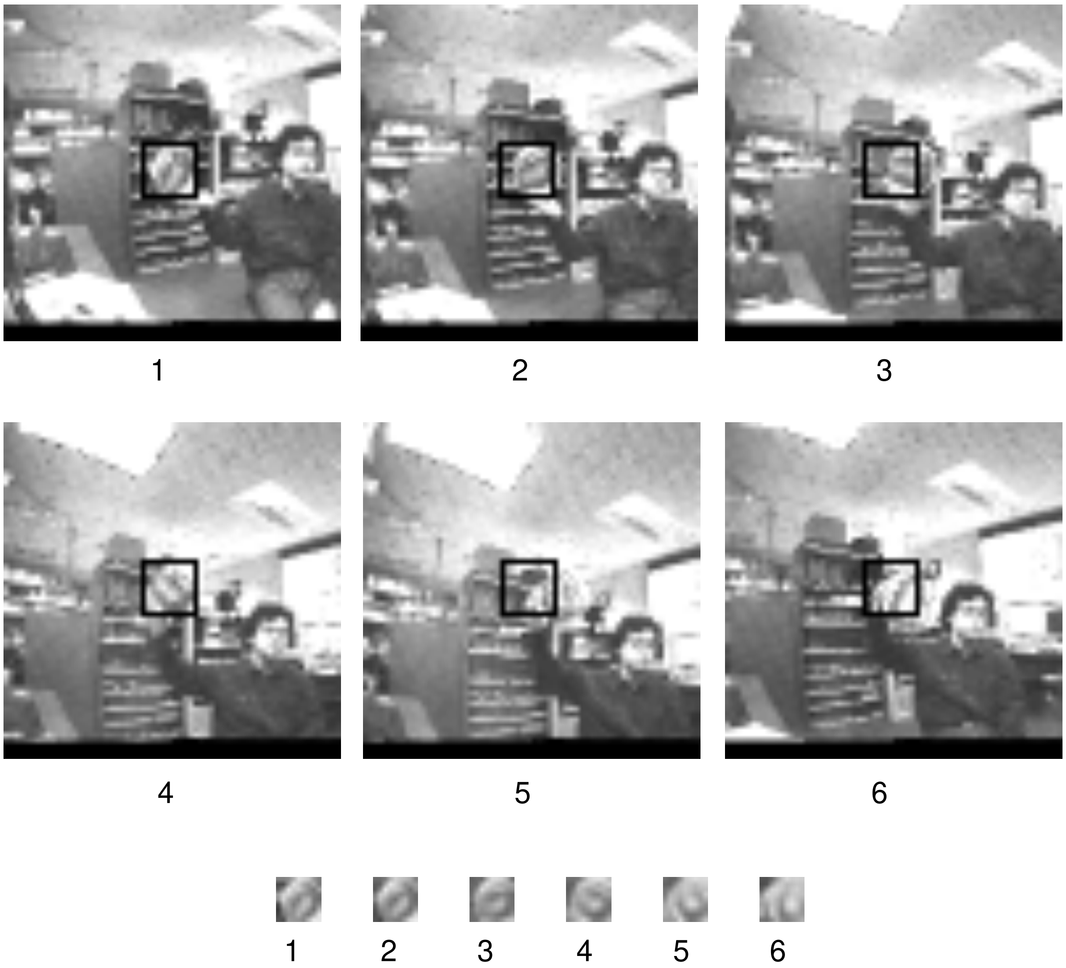
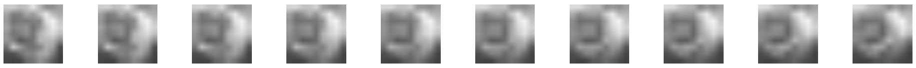
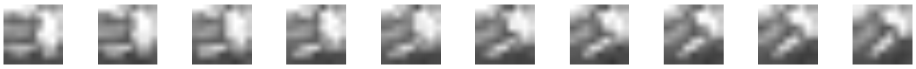


Figure 5.1: Tracking an object.

a) result of traking (every five frames)



b) blended template (every frame)



c) new template (every frame)

Figure 5.2: Template updating for rotating target.

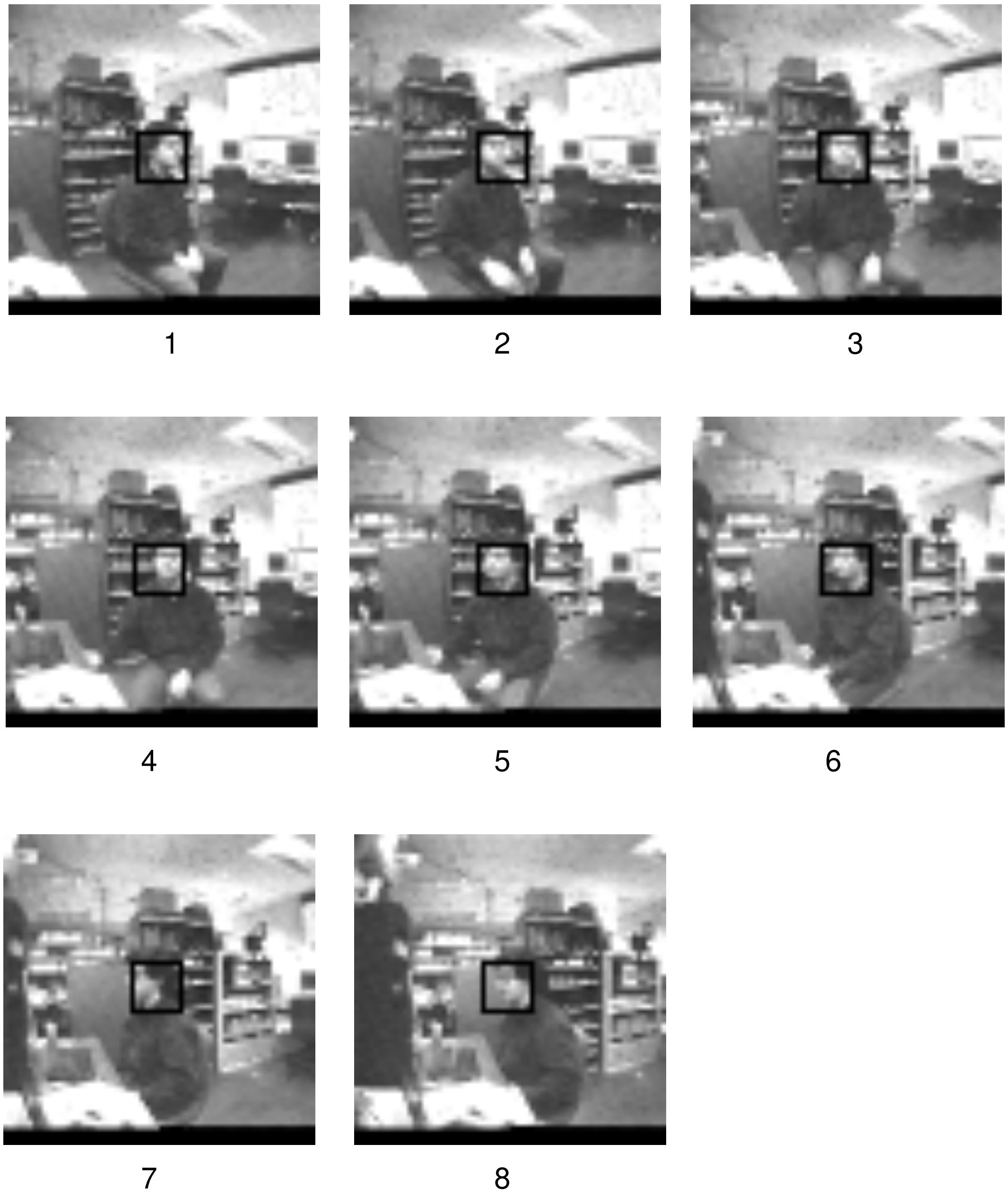


Figure 5.3: Tracking a face.

5.3 Level3:STV

The level 3 network achieved tracking with vergence. The second eye, left side, was controlled by the vergence module, and followed the target in its center of the view. This demonstrated motor binocularity. The vergence module also output the result of figure-ground separation, as a sensory binocularity. Then the separation was used for vergence control as feedback, and it made it possible to keep the vergence more robust than without feedback.

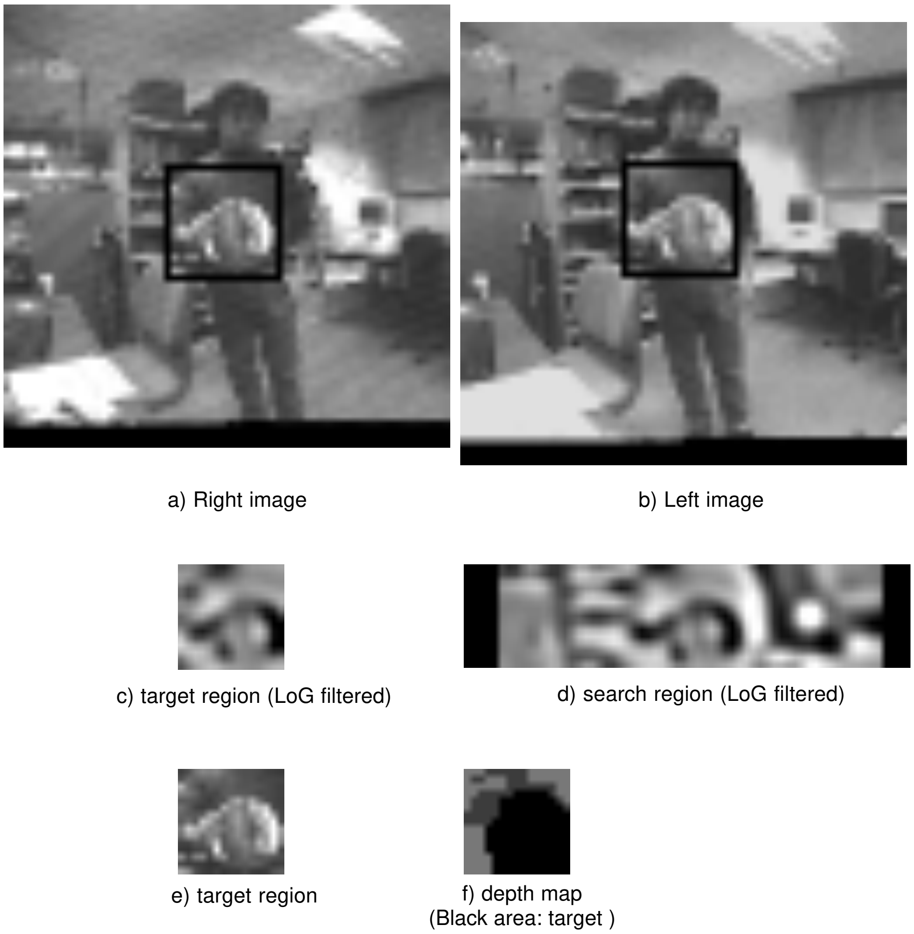


Figure 5.4: Figure-ground separation.

Fig.5.4 illustrates the input images and LoG filtered images used for the matching, and the resulting depth map[1]. In the depth map, the darker pixels are nearer to the verged depth. Here only ± 2 pixels were searched for the matching and the two eyes were verged on the target, which was nearest to the eyes in the region, only three values were shown in the depth map. The black pixels, which is in the verged depth, correspond to the circular target shape.

When the background is clean and flat textured, like **Fig.5.4**, the figure-ground separation seems rather easy. **Fig.5.5** is a cluttered background case. Even with such a background that has a lot of edges, the separation worked. **Fig.5.6** demonstrate the ability to keep vergence on the target by feedback of the separation to the vergence matching, even when it is moving out from the region of attention. Especially, when the background has a strong vertical edges, the eyes tend to verge on the background. However, as shown in **Fig.5.6**, the system could keep the vergence on the target, even when half of the target was going out from the region of attention. This is the effect of feedback from the figure-ground separation. In contrast, if eyes verged on the edge of the shelf in background at first and then the target came in, the eyes could keep the vergence on the background.

When the target was going out more than the case shown in **Fig.5.6**, the eyes lost the vergence on the target in spite of the feedback. It is shown in **Fig.5.7**. Now the vergence point is moving from the target to the background. It is not on the background yet because of the effect of the median filter in the vergence control.

For comparison, **Fig.5.8** demonstrates the case without feedback of the depth map. Here, the eyes tend to lose the vergence on the target more easily than the case with feedback.

[1] In the figures in this chapter, right and left images are so located that they can be seen as a stereogram.

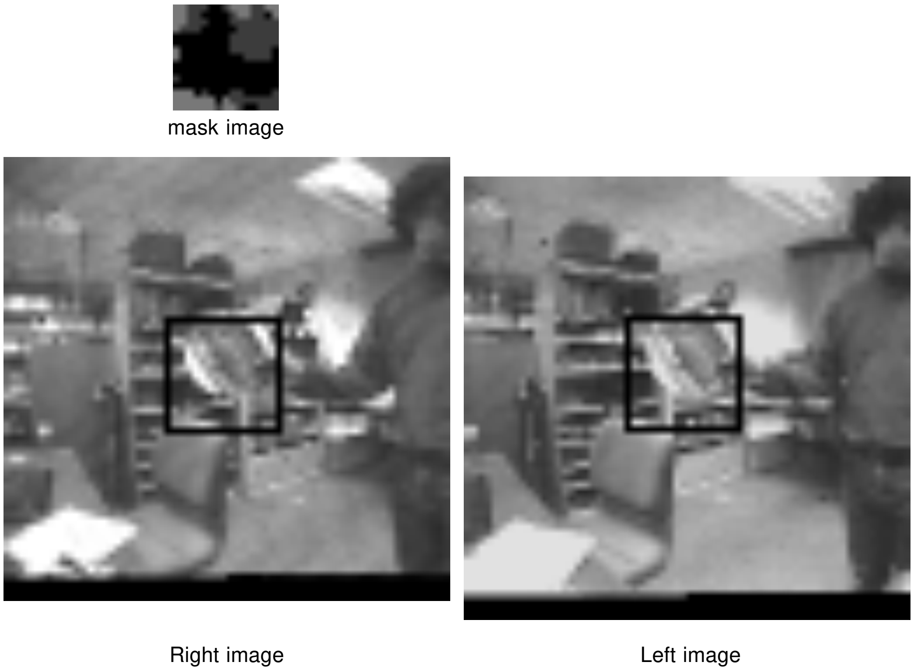


Figure 5.5: Figure-ground separation (with cluttered background).

The implemented networks showed ability to track the object with vergence and figure-ground separation. It also indicated the effect of template updating and vergence control with the figure-ground separation. However, the switching mechanism for tracking and saccade needs more effort to achieve natural switching, and the overall system requires a higher frame rate to work more stably and naturally. Possible refinement and future work is discussed in the next chapter.

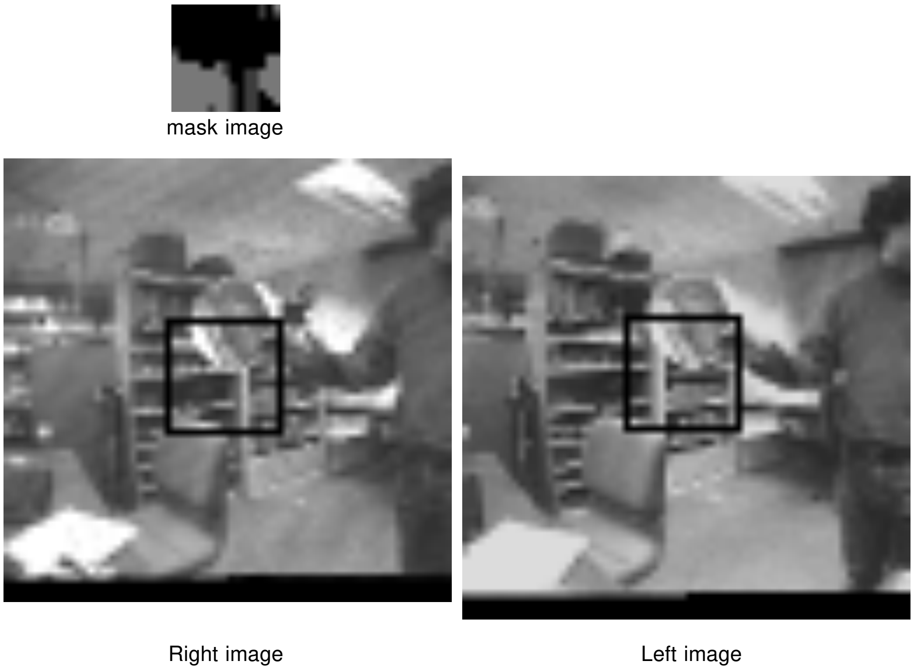


Figure 5.6: Figure-ground separation (keeping vergence to the object).

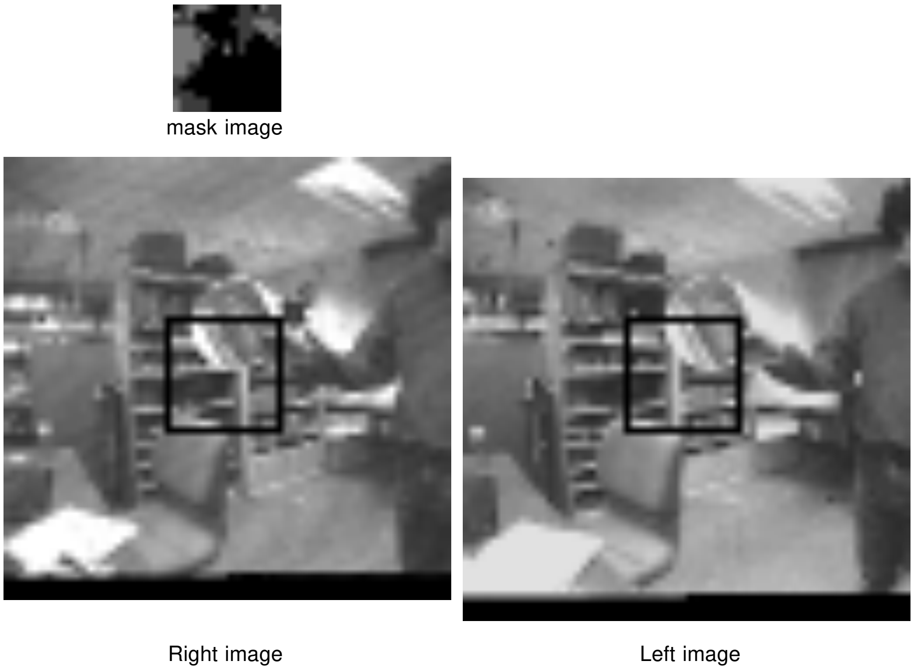


Figure 5.7: Figure-ground separation (lost vergence).

a) verged to object

b) verged to background

Figure 5.8: Figure-ground separation without feedback of mask.

Chapter 6

Conclusion

This chapter describes some improvements and extensions that can be made to the system, and final conclusions.

6.1 Futurework

Capturing a New Target

In the level 2 network, the switching from tracking to saccade was good, but from saccade to tracking was not done naturally. The problem was in the process of capturing a new target for tracking. The system sometimes failed to capture the target and the gaze was located somewhere in the background. The main reason was that it takes time to saccade and get a new target region after detecting motion. The moving target changes its position during the time and the system sometimes directs the gaze to the place where the target was a little before, then continues to watch the background. Although watching the background is terminated by the *Bored* signal, this problem needs to be solved. One possible improvement in the new target capture is to obtain a image of target region before the saccade. This might be good to make the process more stable, however this solution needs more communication between the saccade and tracking modules. It needs to transfer the template between the modules and it makes the task decomposition less simple and clear.

There are some features which are good to be implemented on the system but have not been done so far.

Using figure-ground separation for tracking

The obtained figure-ground separation was used only for vergence control within the same module. But it can also be used as a mask for tracking template to focus the target in the region of attention and to ignore the background in the template. It can be effective to make the tracking more stable, especially for clutter background or small target. However, this cause the transmission of the depth map from the vergence module to the tracking module and it might effect the speed.

Fovea cameras

In this thesis, only two wide angle cameras were used. Narrow angle cameras were not used in the implemented network. Some possibilities of usage of the narrow angle cameras and the cooperation with wide angle cameras were considered, for example, rough vergence by wide angle cameras and then precise vergence by narrow angle camera, and refinement of figure-ground separation using edge from narrow angle camera, and so on. However the full implementation remains to be future work.

Other cues than motion

The only cue used for capturing a new target is motion information in the implemented system. But there are a lot of possible cues for the purpose. Brightness, color, and some special patterns like the face and hand of humans. The face detection is especially important for humanoid robot to interact with humans. The integration of these many cues for attention is an interesting topic.

Learning

One advantage of developmental approach is, as described in chapter 2, that it facilitates learning. However, learning ability was not implemented on the networks described in this thesis. One possible learning scenario is about learning rough absolute distance through the hand and vergence coordination. While the humanoid robot looks at the hand, the humanoid can relate the vergence angle to its hand position. This mapping from vergence angle to the arm control can be used for touching a target in three dimensional space.

6.2 Conclusion

In this thesis, the developmental approach in building embodied robotic system is introduced, and applied to the stereo tracking vision system, to give a task decomposition. The networks of the modules are proposed based on the task decomposition, and implemented on the DSP network. The networks demonstrated the ability to track the object with vergence and achieve rough figure-ground separation. The result of figure-ground separation is utilized for keeping the vergence on the target more robustly.

Bibliography

[BBI+98] Rodney A. Brooks, Cynthia Breazeal(Ferrell), Robert Irie, Charles C. Kemp, Matthew Marjanović, Brian Scassellati, and Matthew Williamson. Alternative essences of intelligence. In *AAAI 98*, 1998.

[Bro86] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[Bro91] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–160, 1991.

[BS94] Rodney A. Brooks and Lynn A. Stein. Building brains for bodies. *Autonomous Robots*, 1(1):7–25, 1994.

[CB93] David Coombs and Christopher Brown. Real-time binocular smooth pursuit. *International Journal of Computer Vision*, 11:2:147–164, 1993.

[Gol96] E. Bruce Goldstein. *Sensation and Perception 4th Ed.* Brooks/Cole Publishing, 1996.

[Hel91] Richard Held. *Binocular Vision*, chapter "Development of binocular vision and stereopsis", pages 170–178. Macmillan, 1991.

[Kem97] Charles Kemp. A platform for visual learning. Master's thesis, MIT Department of Electrical Engineering and Computer Science, 1997.

[KKK+95] Takeo Kanade, Hiroshi Kano, Shigeru Kimura, Atsushi Yoshida, and Kazuo Oda. Development of a video rate stereo machine. In *International Robotics and Systems Conference(IROS-95)*, 1995.

[MSW96] Matthew Marjanović, Brian Scassellati, and Matthew Williamson. Self-taught visually-guided pointing for a humanoid robot. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 1996.

[PSed] Rolf Pfeifer and Christian Scheier. *Understanding Intelligence: The"New AI" Approach to Cognition.* MIT Press, 1998(to be published).

[PUE96] Kourosh Pahlavan, Tomas Uhlin, and Jan-Olof Eklundh. Dynamic fixation and active perception. *International Journal of Computer Vision*, 17(2):113–135, 1996.

[Sca98a] Brian Scassellati. A binocular, foveated active vision system. Memo 1628, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts, January 1998.

[Sca98b] Brian Scassellati. Building behaviors developmentally: A new formalism. In *AAAI Spring Symposium: Integrating Robotics Research*, 1998.

[Sca98c] Brian Scassellati. Finding eyes and faces with a foveated vision system. In *Proceedings of the American Asociation of Artificial Intelligence AAAI-98*, 1998.

[Sca98d] Brian Scassellati. Imitation and mechanisms of shared attention: A developmental structure for building social skills. In *Agents in Interaction - Acquiring Competence through Imitation: Papers from a Work-*

shop at the Second International Conference on Autonomous Agents, 1998.

[TGC+94] Frank Thorn, Jane Gwiazda, Antonio A. V. Cruz, Joseph A. Bauer, and Richard Held. The development of eye alignment, convergence, and sensory binocularity in young infants. *Investigative Ophthalmology and Visual Science*, 35:544–553, 1994.

[Wes95] Mike Wessler. A modular visual traking system. Master's thesis, MIT Department of Electrical Engineering and Computer Science, 1995.

[Wil96] Matthew M. Williamson. Postural primitives: interactive behavior for a humanoid robot arm. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Society of Adaptive Behavior, 1996.