Animat Vision
Active Vision in Artificial Animals

by

Tamer F. Rabie

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

Animat Vision
Active Vision in Artificial Animals

Tamer F. Rabie

Doctor of Philosophy

Graduate Department of Electrical and Computer Engineering

University of Toronto

1999

We propose and demonstrate a new paradigm for active vision research which draws upon recent advances in the fields of artificial life and computer graphics. A software alternative to the prevailing hardware vision mindset, *animat vision* prescribes artificial animals, or animats, situated in physics-based virtual worlds as autonomous virtual robots with active perception systems. To be operative in its world, an animat must autonomously control its eyes and actuated body. Computer vision algorithms continuously analyze the retinal image streams acquired by the animat's eyes, enabling it to locomote purposefully through its world. We describe an initial animat vision implementation within lifelike artificial fishes inhabiting a physics-based, virtual marine world. Emulating the appearance, motion, and behavior of real fishes in their natural habitats, these animats are capable of spatially nonuniform retinal imaging, foveation, retinal image stabilization, color object recognition, color stereo obstacle avoidance, and perceptually-guided navigation. These capabilities allow them to foveate and pursue moving targets of interest, such as other artificial fishes, while exercising the sensorimotor control necessary to avoid collisions and predators. We demonstrate that the animat vision paradigm

extends to virtual environments inhabited by virtual humans. Animat vision offers a fertile approach to the development, implementation, and evaluation of computational theories that profess sensorimotor competence for animal or robotic situated agents.

Acknowledgements

All praise is due to Allah, and peace and blessings be upon the final Messenger of Allah.

I would like to thank my advisor Demetri Terzopoulos for providing a research topic that is unique and of intellectual interest. I learned a great deal from him. His modest personality and friendly attitude greatly enhanced our working relationship. I have also benefited from his experience and understanding of how to do interesting research that stands out and is recognized.

I would also like to thank all colleagues in the vision lab for interesting discussions which made the time spent very enjoyable. In particular, I would like to thank Xiaoyuan Tu who developed the artificial fish animat for her creativity, dedication, and cooperation. I thank her and Radek Grzeszczuk for their many important contributions to the artificial fishes project. I also thank the many persons who have discussed and debated the animat vision idea with us, especially John Tsotsos, Geoffrey Hinton, and Allan Jepson.

I cannot forget to thank Michael Swain and Michael Black for discussing their work with me and extending their help when we needed it.

I save my final thanks to my beloved Mother and Father for always encouraging me to be the best and for their many sacrifices. Without their continued guidance and caring support I would not have been able to reach this academic level. My thanks also goes to my dear wife for her endurance and support during days of work pressure and deadlines.

I dedicate this thesis to my beloved late grandparents; Mama Hagga and Giddo Hassan for taking me under their loving caring wing, and putting my feet on the path to knowledge. May Allah have mercy on their souls and accept them with the company of the righteous in the highest levels of paradise. Ameen.

This work was supported by a research grant from the Natural Sciences and Engineering Research Council of Canada and by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., Industry Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro Technologies, and Atomic Energy of Canada Limited.

Contents

List of Figures

Chapter 1

Introduction

Vision has been regarded in its early stages as the problem of determining "what is where by looking." Three decades ago computer vision research was primarily concerned with the passive inversion of the image formation process [Roberts, 1965]. Almost two decades ago David Marr [Marr, 1982] proposed a computational theory for vision which prescribes detailed three dimensional reconstruction to compute representations of scenes for the purpose of object identification. Until the mid-eighties, the limitations of computing power restricted experimentation in computer vision to the analysis of static scenes. In the last decade, powerful, general purpose processors have become widely available, as have special-purpose vision hardware such as video frame grabbers and pipelined low-level image analysis systems [Brown and Terzopoulos, 1994]. This has resulted in an increased interest by vision researchers in the study of perception from the point of view of an active observer or agent in a dynamic word. It has lead to a re-evaluation of the goals of computer vision itself and the emergence of a new dominant paradigm commonly known as *active vision.*

In contrast to the earlier (passive vision) approach, the new objective is to construct active vision systems that possess visual skills which allow them to interact with a dynamic environment. The agents in which active vision systems are usually embedded are typically mobile robots. Active vision research in most labs today is in reality the technologically driven pursuit of "hardware vision." To be sure, applications-minded

researchers have legitimate reasons for building robot vision systems, but the necessary hardware paraphernalia—CCD cameras, pan-tilt mounts, ocular heads, frame-rate image processors, mobile platforms, manipulators, controllers, interfaces, etc.—can be expensive to fabricate or acquire commercially and a burden to maintain in working order.

Advances in the emerging field of artificial life (ALife) make possible a fresh approach to computational vision.[1] A major theme in ALife research is the synthesis of artificial animals, or "animats" [Husbands, 1994]. Animats, a term coined by Wilson [Wilson, 1991], are computational models of real animals situated in their natural habitats. A recent breakthrough in animat research has produced situated virtual agents that realistically emulate animals of nontrivial complexity [Terzopoulos *et al.*, 1994]. This advance prompts us to propose *animat vision*, an approach which prescribes the use of artificial animals as autonomous virtual robots for active vision research.

1.1 The Animat Vision Concept

The animat vision concept in a nutshell is to implement, entirely in software, realistic artificial animals and to give them the ability to locomote, perceive, and in some sense understand the realistic virtual worlds in which they are situated so that they may achieve individual and social functionality within these worlds. To this end, each animat is an autonomous agent possessing a muscle-actuated body capable of locomotion and a mind with perception, motor, and behavior centers. The animat is endowed with functional eyes that can image the dynamic 3D virtual world onto 2D virtual retinas. The perceptual center of the animat's brain exploits active vision algorithms to process continually the incoming stream of dynamic retinal images in order to make sense of what the animat sees so that it can purposefully navigate its world.

[1] For an engaging introduction to the ALife field, see, e.g., S. Levy, *Artificial Life* (Pantheon, 1992).

1.2 Benefits of Animat Vision

The animat vision methodology that we propose in this thesis can potentially liberate a significant segment of the computer vision research community from the tyranny of robot hardware. It addresses the needs of scientists who are motivated to understand and ultimately reverse engineer the powerful vision systems found in higher animals. These researchers are well aware that animals do not have CCD chip eyes, electric motor muscles, and wheel legs. That is to say, they realize that readily available hardware systems can be poor models of biological animals. For lack of a better alternative, however, they have been struggling with inappropriate hardware in their ambition to understand the complex sensorimotor functions of real animals. Moreover, their mobile robots typically lacked the compute power to achieve real-time response within a fully dynamic world while running active vision algorithms of much sophistication. Yet in their ambition to understand the complex sensorimotor functions of real animals, active vision researchers have been forced to struggle with whatever hardware is available to them, for lack of a better research strategy; that is, until now.

Animat vision offers an alternative research strategy for developing biologically inspired active vision systems. It circumvents the aforementioned problems of hardware vision. The animat vision concept is realized with realistic artificial animals and active vision algorithms implemented entirely in software on readily available 3D graphics workstations. Animat vision offers several additional advantages:

- One can arbitrarily slow down the "cosmic clock" of the virtual world relative to the cycle time of the CPU on which it is being simulated. This increases the amount of computation that each agent can consume between clock ticks without retarding the agent's responses relative to the temporal evolution of its virtual world. This in turn permits the development and evaluation of sophisticated new active vision algorithms that are not presently implementable in real-time hardware.

- The quantitative photometric, geometric, and dynamic information that is needed

to render the virtual world is available explicitly. Generally, the animats are privy to no environmental ground truth data, but must glean visual information "the hard way"—from retinal image streams. However, the readily available ground truth can be extremely useful in assaying the effective accuracy of the vision algorithms or modules under development.[2]

We will argue in this thesis that animat vision can offer a fertile approach to the development, implementation, and evaluation of computational theories that profess sensorimotor competence for animal or robotic situated agents.

1.3 Examples

The reader may doubt the possibility of implementing artificial animals rich enough to support serious active vision research. Fortunately, this hurdle has already been cleared. Recent animat theory encompasses the physics of the animal and its world, its ability to locomote using internal muscles, its adaptive, sensorimotor behavior, and its ability to learn. In particular, an animat with these essential capabilities has been implemented that emulates animals as complex as teleost fishes in their marine habitats [Terzopoulos *et al.*, 1994, Tu and Terzopoulos, 1994a].

Imagine a virtual marine world inhabited by a variety of realistic fishes (Fig. 1.1). In the presence of underwater currents, the fishes employ their muscles and fins to swim gracefully around immobile obstacles and among moving aquatic plants and other fishes. They autonomously explore their dynamic world in search of food. Large, hungry predator fishes stalk smaller prey fishes in the deceptively peaceful habitat. The sight of predators compels prey fishes to take evasive action. When a dangerous predator appears, similar species of prey form schools to improve their chances of survival (Fig. 1.2). As the predator nears a school, the fishes scatter in terror. A chase ensues in which the predator selects victims and consumes them until satiated. Some species of fishes seem

[2]It is often convenient to represent ground truth data iconically in the form of retinocentric intrinsic images, including intensity, range, illumination, reflectance, and object identity images, and these can be computed easily and quickly by the rendering pipelines of 3D graphics workstations.

Figure 1.1: Artificial fishes in their physics-based virtual world as it appears to an underwater observer. The 3 reddish fish (center) are engaged in mating behavior, the greenish fish (upper right) is a predator hunting for small prey, the remaining 3 fishes are feeding on plankton (white dots). Seaweeds grow from the ocean bed and sway in the current.

untroubled by predators. They find comfortable niches and feed on floating plankton when they get hungry.

A challenge undertaken in this thesis is to synthesize an active vision system for realistic artificial animals which is based solely on retinal image analysis. The vision system should be extensible so that it will eventually support the broad repertoire of individual and group behaviors described above. It is important to realize that we need not restrict ourselves to modeling the perceptual mechanisms of real fishes. In fact the animat vision paradigm applies to any animat that models an animal to a reasonable level of fidelity. The animat vision system developed in this thesis does not model fish vision. Rather, the fish animat serves as a virtual piscatorial robot that is an active observer of its world.

The basic functionality of the animat vision system starts with binocular perspective projection of the 3D world onto the animat's 2D retinas. Retinal imaging is accomplished by photorealistic, color graphics rendering of the world from the animat's viewpoint. This

Figure 1.2: A predator shark is stalking a school of prey fish in the background.

projection respects occlusion relationships among objects. It forms spatially variant visual fields with high resolution foveas and low resolution peripheries. Fig. 1.3 shows a stereo image pair rendered from the point of views of the animat's left and right eyes. It is clear from the figure that the center of the left and right retinal images have higher resolution that decreases gradually towards the image borders to simulate the high resolution fovea and lower resolution periphery that characterises primate retinas.

Based on an analysis of the incoming color retinal image stream, the dynamic visual center of the animat's brain supplies saccade control signals to the eyes in order to stabilize the visual fields during locomotion through compensatory eye movements (an optokinetic reflex), and also supplies alerting signals to the animat's motor controler when dangerous obstacles or predators are recognized in the low-resolution visual periphery. This optokinetic and sensorimotor control reflex allows the animat to attend to interesting colored targets, and to keep these dynamic targets fixated. The artificial fish is thus able to approach and track other artificial fishes under visual guidance while exercising the sensorimotor control necessary to avoid collision and predators. Fig. 1.4 show an overhead view of the animat tracking a reddish fish which it has fixated in its high resolution fovea

Figure 1.3: Stereo retinal images acquired by the eyes of the fish animat.

of Fig. 1.3. The white lines emanating from the animat's eyes indicate the gaze direction. They show that the eyes are properly fixated on the target in the fovea. Fig. 1.5 shows the gaze geometry from the animat's eyes to the fixated target.

The proposed animat vision paradigm is flexible enough to be implanted into animats other than virtual fish. Fig. 1.6 shows the animat vision system in action in a human soldier animation API called *DI-Guy* developed by Boston Dynamics, Inc. The figure shows a DI-Guy soldier visually tracking and pursuing an enemy soldier. The green lines emanating from the soldier's eyes indicate the gaze direction. Fig. 1.7 shows stereo retinal images captured by the eyes of the observer. Fig. 1.8 demonstrates an animat vision system controlling a version of the famous interactive computer game *DOOM* developed by id Software, Inc. In the figure, the doom warrior is in the process of visually recognizing a hostile enemy.

1.4 Contributions

We have proposed and developed our animat vision concept in the following articles: [Terzopoulos and Rabie, 1995, Rabie and Terzopoulos, 1996, Terzopoulos and Rabie, 1997, Rabie and Terzopoulos, 1998]. The major contributions of this thesis are as follows:

Figure 1.4: Top- and Side-view of the animat tracking another fish which it has fixated in its fovea.

- We introduce the animat vision approach which prescribes the use of artificial animals for active vision research. Artificial animals, which are implemented entirely in software, have the ability to locomote, perceive, and understand the virtual worlds in which they are situated. Our claim is that the animat vision approach is a fruitful approach which is complementary to the development of active vision systems based on hardware implementations.

- We demonstrate the animat vision approach by building a prototype active vision system in the artificial fish animat. The vision system makes exclusive use of the stream of retinal images, which are acquired by the agent's mobile eyes, to analyse the surrounding environment and interact accordingly.

- We further demonstrate the animat vision approach in two other virtual environments. We implement a similar animat vision prototype for the DI-Guy animat: a realistic human model with life-like human motions and actions. Furthermore, a simplified version of the algorithm is implemented for the warrior animat in the popular video game called Doom.

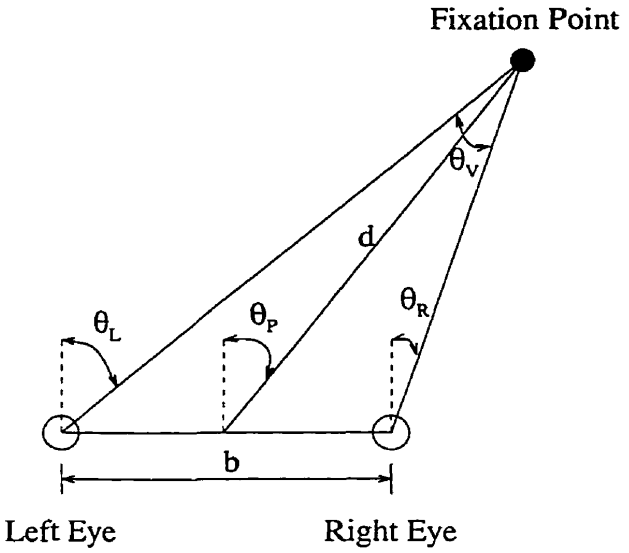


Figure 1.5: Gaze angles and range to target geometry.

- We adapt and integrate a suite of active vision algorithms into the working prototype animat vision system. We integrate motion and color-based gaze control algorithms to enhance the prototype and to support more robust vision-guided navigation abilities in the animat. We further enhance the animat's navigation and perception abilities by combining stereo and color-based motor control algorithms to extend the animat's functionality by supporting obstacle recognition and avoidance.

- We make improvements to the color histogram intersection methods originally introduced by Swain [Swain and Ballard, 1991]. We develop a more robust intersection measure that is invariant to scale changes. We adapt it to foveated systems to make use of the information present in the lower resolution periphery.

- We adapt Black's [Black and Anandan, 1993] robust incremental optical flow approach to the animat vision system. We develop an incremental motion segmentation technique that makes use of the robust optical flow estimate at each time instant to refine an initial segmentation over time as the animat navigates and acquires more retinal image frames. Also, motion and color are integrated to increase the robustness of the animat's recognition senses.

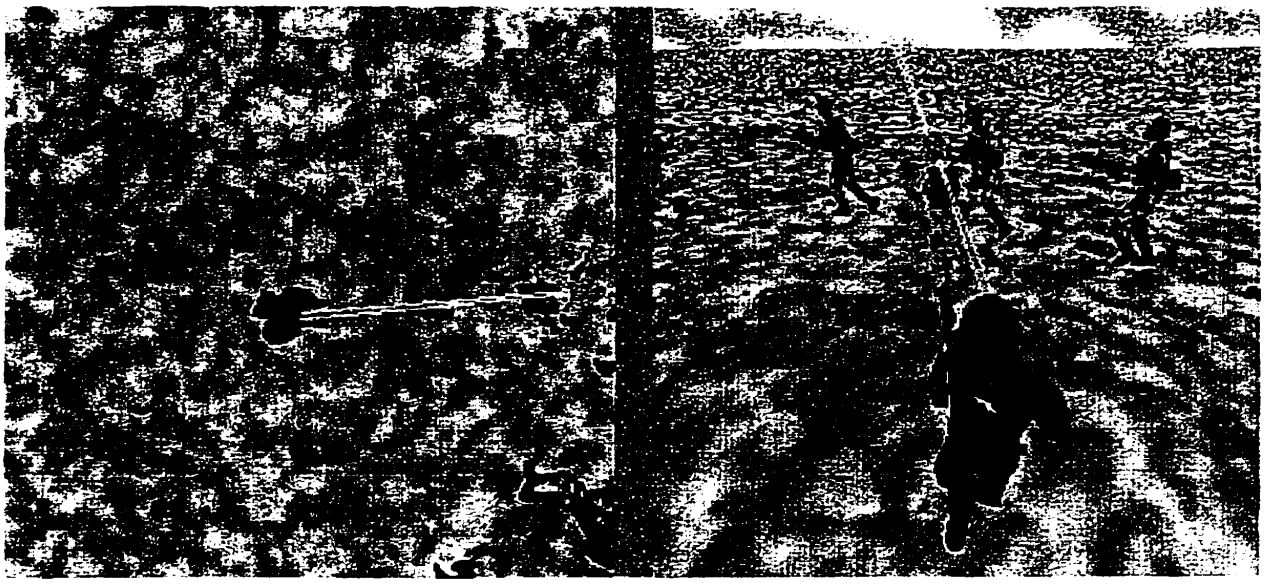


Figure 1.6: Top and side view of a DI-Guy soldier animat visually tracking another soldier.

- We exploit in the animat vision system stereo and color cues for enhanced dynamic obstacle recognition and avoidance. We develop stereo disparity algorithms based on steerable filters that make use of the color signals available naturally from the photorealistic images acquired by the animat to improve the matching process and to obtain more accurate disparity estimates. We show that this method is very effective and, when combined with color cues, it gives the animat the abilities required to avoid obstacles.

1.5 Thesis Overview

This thesis is presented in 8 chapters. Chapters 3 to 6 present theoretical and practical aspects of system implementation, while the last chapter gives a discussion of the work presented and some concluding remarks.

Chapter 2 briefly motivates the animat vision approach vis-a-vis conventional active vision based on robot hardware and discusses the background of our work. Chapter 3 reviews the fish animat in more detail describing its physical model implementation. Chapter 4 presents an initial functional implementation of the animat vision system, with

Figure 1.7: Stereoscopic retinal images captured by the virtual soldier's eyes.

Figure 1.8: Stereoscopic retinal images captured by the Doom animat as it recognizes an adversary.

a detailed description of its development.

Chapters 5 and 6 present active vision additions to the animat vision system enhancing the animat's functionality in its virtual environment. In Chapter 5 motion and color cues are integrated to increase the robustness of the animat's perceptual functions. Chapter 6 adds dynamic obstacle recognition and avoidance capabilities by exploiting stereo and color cues naturally available from the photorealistic images acquired by the animat's binocular eyes. These chapters also give an example of how vision algorithms can be evaluated within the proposed animat vision framework.

Chapter 7 shows that the proposed animat vision paradigm is flexible enough to be

implanted into animats other than virtual fish by integrating the vision system into two different virtual environments inhabited by humans characters. This demonstrates the versatility of the paradigm and that it is a general active vision framework applicable to robotic systems of varying degrees of complexity.

Chapter 8 draws conclusions from our animat vision work and suggests possible directions for future research.

Chapter 2

Motivation and Background

Our zoomimetic approach to computer vision is made possible by the confluence of three recent trends: 1) advanced physics-based artificial life modeling of natural animals, 2) photorealistic computer graphics rendering and its efficient implementation in modern 3D graphics workstations, and 3) active computer vision algorithms. In this chapter we review approaches related to our animat vision approach, and give some background on areas of active vision that currently interest researchers.

2.1 Related Work

J.J. Gibson [Gibson, 1979], in a sense the grandfather of active vision, stresses in pre-computational terms the importance of modeling the active observer situated in the dynamic environment. In his theory of direct perception, the environment is to be regarded as the repository for information. Thus, no internal representation is needed from this point of view since all the information required for action is already out there in the world and it is the responsibility of the active observer to choose what is needed from the images to carry out a particular task. Versions of this paradigm suitable for mainstream computer vision were introduced in the seminal papers of Bajcsy [Bajcsy, 1988] and Ballard [Ballard, 1991] under the names of active perception and animate vision, respectively. The active vision approach was further developed by Aloimonos *et al.* [Aloimonos *et al.*, 1987] and many others (see, e.g., [Ballard and Brown, 1992, Blake and Yuille, 1992, Swain and Stricker, 1993]) into the prevalent paradigm that it is today.

The artificial animals that we advocate in this thesis are active "vehicles" in the sense of Braitenberg [Braitenberg, 1984]. We believe that they are as appropriate for grounding active vision systems as are hardware mobile robots. The first biologically inspired mobile robots were developed by Grey Walter in the 1950's [Walter, 1953]. His "turtle" robots, which were equipped with a simple controller using light sensors, exhibited interesting phototropic navigation trajectories. This illustrated well that even simple control functions can generate quite complex behaviors when interacting with a dynamic environment. Modern versions of these mobile robots have emerged from the situated robotics work of Brooks and his group [Brooks, 1991] and they have been an inspiration to numerous other robotics groups (see, e.g., the compilation [Maes, 1991]). Undeniably, however, efforts to equip real-time mobile robots with general-purpose biologically inspired active vision systems have been hampered by the hardware and the relatively modest abilities of on-board processors [Prokopowicz and Cooper, 1995, Grosso *et al.*, 1995, Murray *et al.*, 1995, Seelen *et al.*, 1995].

Artificial fishes are animats of unprecedented sophistication. They are autonomous virtual robots situated in a 3D continuous virtual world governed by physical dynamics. This makes them suitable for grounding active vision systems. By contrast, Wilson's original animat [Wilson, 1991], which was proposed for exploring the acquisition of simple behavior rules, is a point marker on a non-physical 2D grid world that can move between squares containing food or obstacles. Other simple animats include the 2D cockroaches of Beer [Beer, 1990]. A more sophisticated animat is the kinematic dog described by Blumberg and Galyean [Blumberg and Galyean, 1995]. Prior animats make use of "perceptual oracles"—schemes for directly interrogating the virtual world models to extract sensory information about the environment as needed by the animat. One can also find several instances of "oracle vision" in the behavioral animation literature [Reynolds, 1987, Renault *et al.*, 1990, Tu and Terzopoulos, 1994b]. Of these, the "synthetic vision" work of Renault *et al.* is most relevant to animat vision. The animat vision approach is also related to that of Maes [Maes *et al.*, 1994], but there are major differences. Her ALIVE

system, which is inhabited by simple animats, images the outside world through a CCD camera. Image processing hardware executing region tracking algorithms analyzes the camera image in real time. This enables a person to interact with the animats through body gestures. Unlike oracle vision, for the animat vision approach to make sense, it is absolutely necessary that the animat and its world attain a high standard of visual fidelity.

2.2 Background on Active Vision

Looking at vision from the point of view of an active, autonomous agent changes our view of what the vision problem is and how it should be approached. Computer vision research carried out from this point of view is variously called *active, animate or purposive* vision. A number of researchers have argued that it constitutes a new paradigm for computer vision, and may lead to significant advances in robotics and to a better understanding of vision in general.

Active vision systems have mechanisms that can actively control camera parameters such as orientation, focus, zoom, and vergence in response to the requirements of the task and external stimuli. They may also have anthropomorphic features such as spatially variant (foveal) sensors, binocularity, and high speed gaze control [Ballard, 1991, Prokopowicz and Cooper, 1995, Grosso *et al.*, 1995]. More broadly, active vision encompasses attention (selectively sensing in space, resolution and time), whether it is achieved by modifying physical camera parameters or the way data is processed after leaving the camera [Tsotsos *et al.*, 1995, Swain and Stricker, 1993].

Active systems can dramatically simplify the computations of early vision:

- Enabling areas of interest to be examined at the desired resolution (inside the fovea) without the costs of uniformly high resolution sensing.

- Simplifying segmentation of an object of interest from its background, using controlled motions to disambiguate solutions that are otherwise underconstrained.

- Examining unseen areas of the scene, and simplifying the transition from image to world coordinates.

The tight coupling between perception and action proposed in the active-vision paradigm does not end with camera movements. In this paradigm, visual processing is tied closely with the activities it supports (navigation, manipulation, signaling danger or opportunity, etc.) allowing simplified control algorithms and scene representations, quick response time, and increased success at supporting the goals of the activities. In addition, integration provides new metrics with which to judge vision algorithms that are better tuned to applications of computer vision.

The important research areas in active vision can be briefly summarized as follows:

1. **Attention**:

   Visual attention in the most general sense consists of tuning visual processing to those aspects of the visual signal that are relevant to the task at hand. It can often be viewed as signal selection. That is, it often consists of suppressing irrelevant information so that it does not consume resources or interfere with higher level processing. The assumption underlying this is that the visual signal contains far more information than it is possible to analyze in a reasonable amount of time, and (usually) far more than is needed to exhibit useful behavior [Tsotsos *et al.*, 1995].

   Signal selection can take place along various signal dimensions: space, velocity, and distance.

   - Focal Selection: For a given camera position, interest is spatially restricted to a small region within the camera's field of view. This (focal) region is observed at high spatial resolution, while the remaining (peripheral) field is observed at much reduced resolution.
   - Motion Selection: Selection can be achieved in motion by tracking an object moving at a particular velocity. The tracked object will be rendered stationary while objects or patterns moving at other velocities are nonstationary and

perhaps blurred. Tracking is an effective means for isolating one pattern at a time in a complex world containing many differently moving patterns.

- Selection in Depth: A system can select objects at a particular distance from the camera using focus, stereo, or motion parallax. Selection in stereo and motion can be realized by transformations that align regions of interest between images from which the disparity can be estimated and used to infer depth.

2. **Foveated Sensing:**

Biological systems make extensive use of foveated visual systems. In primates, the "fovea" of the retina defines the visual axis of the eye and is responsible for highly detailed and exact vision. The fovea is quite small, consisting of about a $1.5mm$ diameter depression (corresponding to 5.2° of the visual field of view) in the retina situated near the optic axis. The center of the fovea contains only cone receptors (responsible for color vision in bright light) that are much longer and thinner than those on the periphery. This rod free area is about $0.33mm$ in diameter and corresponds to only 1° of the visual field of view. Beyond the fovea is the peripheral retina which constitutes about 97.25% of the retinal concave surface and consists largely of rod receptors (responsible for night viewing) with a sparse distribution of cones in between. There are about an order of magnitude more rods than cones in the human retina: about 120×10^6 rods compared with 6.5×10^6 cones in each eye [Levine, 1985].

While most current machine-vision architectures are uniform in spatial resolution, there are clear advantages to using a multiresolution system. It can provide a wide angle of field with a high resolution in its center (the fovea) and decreasing resolution towards its periphery. If we assume a foveated sensor roughly analogous to the human retina, then the ratio of the sample points needed for this sensor to the sample points needed to provide uniformly high resolution is approximately 1:1000 to 1:10,000 [Swain and Stricker, 1993].

To realize these possibilities, several engineering and algorithmic research problems arise:

- Space-variant systems must be active in order to utilize the higher-resolution parts of the sensor. Actuators must be provided to fixate the sensor on task-specific regions of interest.
- Space-variant systems must be able to effectively detect features that fall on lower resolution areas of the sensor in order to make a rapid fixation to the center (fovea).
- Image-processing and pattern-recognition methods appropriate to space-variant sensors must be developed. Even a simple act such as convolution is "different" in the space-variant context.

3. **Gaze Control**:

One of the central themes of active vision research is *gaze control.* Gaze control is the purposeful alteration of the imaging parameters (viewpoint, viewing directions, vergence angle, focus, etc.) to facilitate the performance of visual tasks. The parameters are typically controlled so as to produce *fixation*, a condition in which the optic axes of the cameras remain fixated on a common point on some world surface. Visual attention and fixation are obviously closely related. It is generally assumed that agents fixate features and objects in the scene to which they are attending [Coombs and Brown, 1991, Coombs and Brown, 1993].

The problem of gaze control can be partitioned into two primary categories: gaze stabilization and gaze change. The former, known as fixation, consists of controlling the available degrees of freedom to maintain clear images of some world object that may be stationary or in motion with respect to the camera. For moving targets, this typically involves target tracking. The latter category, also known as foveation, is more involved: in general, foveation may be directly motivated by the need to reduce the computational complexity of visual tasks. Foveation may be used

to transfer stabilized gaze to new fixation points, similar to human saccadic eye movements, in order to assist in solving low-level or high-level visual tasks.

Gaze control is motivated by the fact that vision in an unrestricted environment must involve the control of imaging parameters. Imagine, for example, a person trying to pilot a vehicle without making any head or eye movements. With only "passive" imaging, these and many other tasks are extremely difficult. In contrast, an active observer will seek an advantageous configuration of camera parameters for a given situation. The ability to control imaging parameters in such a manner facilitates the close interaction of perception and action that is necessary for autonomous interaction with the environment.

For practical, real-world situations, the problems of gaze stabilization and foveation can only be accomplished using active control mechanisms. These capabilities are important even within static environments. In order to perform 3-D reconstruction of a general scene, for example, the cameras must be moved so that different portions of the scene come into view, and lens parameters must be tuned so that sharp, focused images are acquired with appropriate brightness values. For moving objects, the need to control gaze is even more compelling. The ability to stabilize the image of a moving object as the agent is moving also may not only be necessary for robust processing, but can also provide significant computational advantages over the passive case [Ballard, 1991, Aloimonos *et al.*, 1987].

The advantages of gaze control for both static and dynamic situations are as follows:

- *Image Stabilization*: For an object that moves relative to a camera, translational image blur may result unless the object is tracked. Because tracking prevents the object's image from translating, fixation permits simpler schemes for control and for processing of image sequences. Furthermore, tracking objects of interest keeps them in the fovea for a longer period of time, permitting accurate high-resolution analysis.

- *Overcome Limited Field of View*: Any given camera system provides only a limited field of view. Gaze control is needed so that new portions of the scene can be brought into the image, and camera movements are often needed to overcome problems of occlusion. A special case of limited field of view involves sensors that have a central high-resolution fovea and a lower-resolution periphery. Although the overall field of view is very wide, the foveal region is typically very small and must be aimed at locations of interest. Without the ability to fixate, these sensors are severely limited. To state this another way, gaze control makes possible all of the advantages offered by foveal sensors.

- *Optimize Camera/Scene Relationship*: With an active vision system, it is possible to shift the operating point of the system so that imaging parameters are well matched to the object being fixated. For example, when an object lies near the optical axis of a camera, the image may be closely approximated by orthographic projection. This simplifies many computations. An active vision system can also manipulate the level of detail in an image by moving a camera or changing focal length to control the level of detail present in the image.

- *Figure/Background Separation*: If a moving object is tracked, the optical flow in the image for that object is reduced to very small values relative to the rest of the scene. The large differences in the flow fields can be used to segment the image, so that the object being tracked is extracted from the background.

- *Range Calculations*: If a stationary scene point is fixated by a camera mounted on a moving vehicle, the image flow for that point will have zero magnitude. Direction of flow at other points of the flow field gives relative range from the fixation point. Range data fusion is also possible; for example, a scene point predicted by stereo disparity can be examined by directing camera gaze toward the predicted location. Focus or vergence information can then be used to accept, reject, or improve the position estimate [Olson and Lockwood, 1992].

4. **Where to Look Next:**

   Active vision requires a system that purposefully gathers information from the visual world. Information gathering is a dynamic process that responds at once to events in the visual world, to the system's evolving understanding of that world, and to changing requirements of the vision task [Ye and Tsotsos, 1995].

   Vision must be understood as a spatiotemporal process. This is true because events in the world are distributed in time as well as space. It is also true because cost and complexity considerations require that system resources be focused on restricted regions of the scene. Hence a sequence of such focal probes is needed to explore the visual world. Finally, it is true because sequential processing provides for high efficiency through directed analysis: results of each step direct subsequent steps to most important regions of a scene.

   Sequential purposive eye movements address the difficult problem of simultaneously associating many models to many parts of the image. To make this problem computationally tractable within a single fixation, it must be simplified either into a location task (try to find a known object) or an identification task (try to identify an object whose location can be fixated). A simplified model of visual organization consists of a center and a surround. The center is *where-I'm-looking* and the surround is a source of possible new gaze points. A location task is to find the image coordinates of a single model in the presence of many alternatives. In this task the image periphery must be searched; one can assume that the model has been chosen a priori. An identification task is to associate the foveated part of the image with one of many possible models. In this task one can assume that the location of the model to be established is at the fixation point.

   In order for these suggestions to be reasonable, there must be some way of changing gaze (foveation) to fixate a specific object. One difficulty is that it is unreasonable to assume that the location of the object is known precisely, at least on the initial gaze change. Another is that whatever visual mechanism is posited, it must require

only very low spatial resolution, since just prior to gaze change, the target object is typically at the periphery of the visual field. Features that work under these circumstances include motion and color [Swain and Stricker, 1993].

2.3 Summary

Interest in active vision research has increased in recent years with the advent of lightweight cameras and advances in real-time image-processing hardware. However, for the reasons mentioned earlier, active vision research in most labs today is in reality the technologically driven pursuit of hardware vision. The animat vision methodology, proposed herein, with all its advantages over hardware vision is another alternative that can help expedite the development, implementation, and evaluation of computational theories that profess sensorimotor competence for animal or robotic situated agents.

Chapter 3

Review of the Fish Animat

The artificial fish model is developed elsewhere [Terzopoulos *et al.*, 1994, Tu and Terzopoulos, 1994a, Tu and Terzopoulos, 1994b]. This chapter reviews the fish animat to a level of detail sufficient to comprehend the animat vision system developed in subsequent chapters.

Each artificial fish is an autonomous agent with a deformable body comprising a graphical display model and a biomechanical model actuated by internal muscles. As Fig. 3.1 illustrates, the body also includes eyes (among other on-board sensors) and a brain with motor, perception, behavior, and learning centers. Through controlled muscle actions, artificial fishes are able to swim in simulated water in accordance with simplified hydrodynamics. Their functional fins enable them to locomote, maintain balance, and maneuver in the water. Thus the artificial fish model captures not just 3D form and appearance, but also the basic physics of the animal and its environment. Though rudimentary compared to real animals, the minds of artificial fishes are nonetheless able to learn some basic motor functions and carry out perceptually guided motor tasks within a repertoire of piscine relevant behaviors, including collision avoidance, foraging, preying, schooling, and mating.

3.1 Motor System

The motor system (see Fig. 3.1) comprises the fish biomechanical model, including muscle actuators and a set of motor controllers (MCs). Fig. 3.2 illustrates the mechanical body

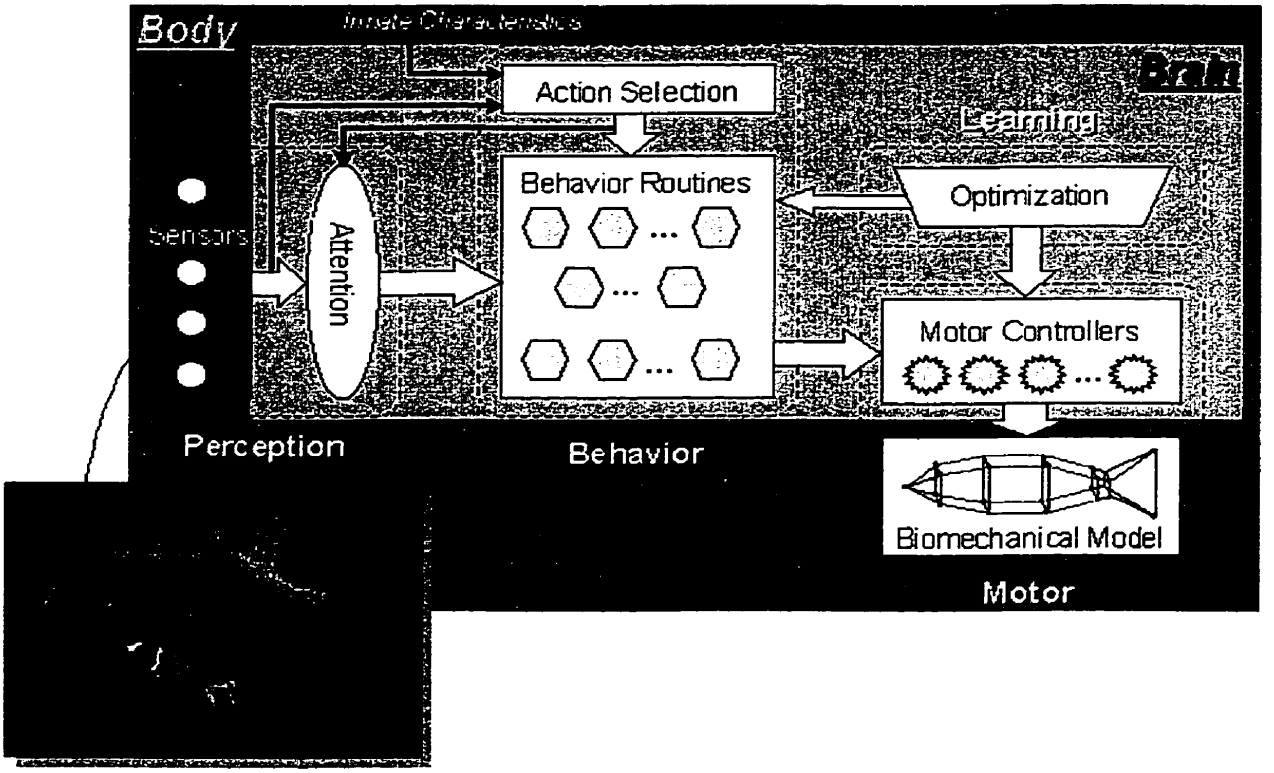


Figure 3.1: The body of an artificial fish comprises a muscle-actuated biomechanical model, perceptual sensors, and a brain with motor, perception, behavior, and learning centers. To the lower left is an artificial fish graphical display model.

model which produces realistic piscine locomotion using only 23 lumped masses and 91 uniaxial viscoelastic elements, 12 of which are actively contractile muscle elements. These mechanical components are interconnected so as to maintain the structural integrity of the body as it flexes due to the muscle actions.

Artificial fishes locomote like real fishes, by autonomously contracting their muscles in a coordinated fashion. As the body flexes it displaces virtual fluid which induces local reaction forces normal to the body. These hydrodynamic forces generate thrust that propels the fish forward. The model mechanics are governed by Lagrange equations of motion driven by the hydrodynamic forces. The system of coupled second-order ordinary differential equations is continually integrated through time by a numerical simulator.[1]

[1]The artificial fish model achieves a good compromise between realism and computational efficiency. To give an example simulation rate, the implementation can simulate a scenario with 10 fishes, 15 food particles, and 5 static obstacles at about 4 frames/sec (with wireframe rendering) on a Silicon Graphics R4400 Indigo^2 Extreme workstation. More complex scenarios with large schools of fish, dynamic plants,

Fig. 3.2 shows an equation of motion used to model the spring mechanics of the artificial fish. For each fish, 69 of these equations (23 nodes ×3 degrees of freedom) are integrated at each time instant to produce locomotion.

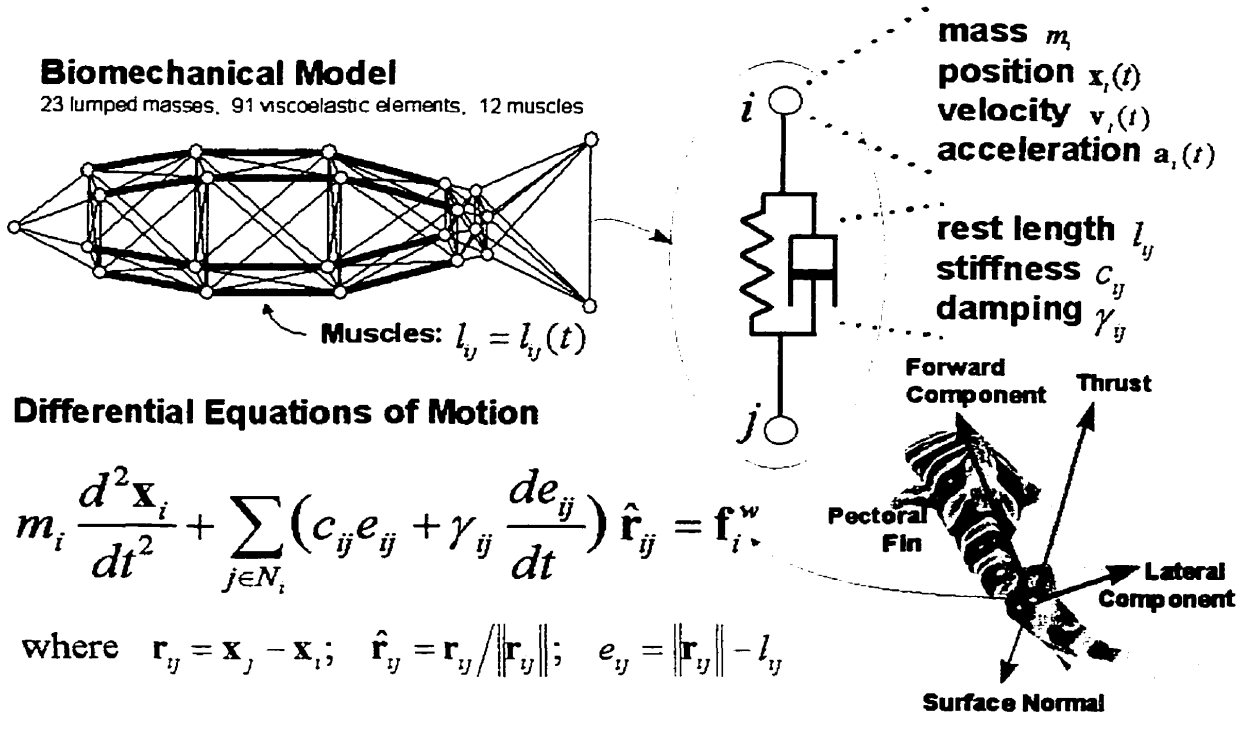


Figure 3.2: Equations of motion that govern the biomechanical fish model.

The model is sufficiently rich to enable the design of motor controllers by gleaning information from the fish biomechanics literature. The motor controllers coordinate muscle actions to carry out specific motor functions, such as swimming forward (`swim-MC`), turning left (`left-turn-MC`), and turning right (`right-turn-MC`). They translate natural control parameters such as the forward speed or angle of the turn into detailed muscle actions that execute the function. The artificial fish is neutrally buoyant in the virtual water and has a pair of pectoral fins which enable it to navigate freely in its 3D world by pitching, rolling, and yawing its body. Additional motor controllers coordinate the fin actions.

---

and full color texture mapped GL rendering at video resolution can take 5 seconds or more per frame.

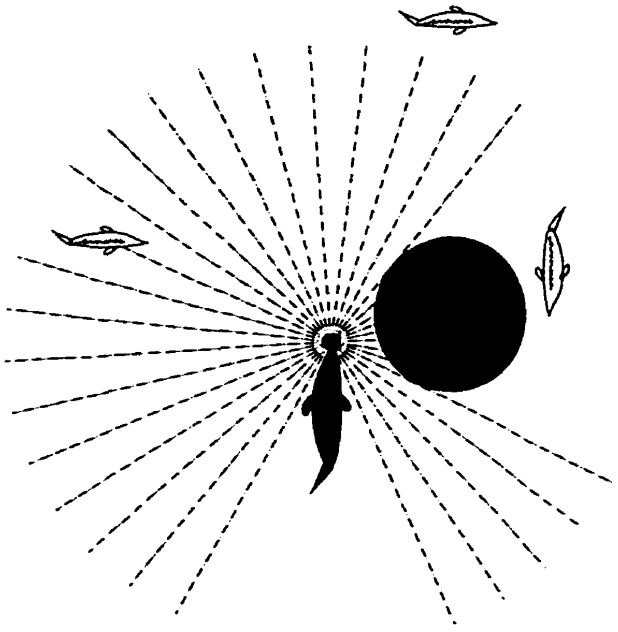


Figure 3.3: Artificial fishes perceive objects within a limited field of view if objects are close enough and not occluded by other opaque objects (only the fish towards the left is visible to the animat at the center).

3.2 Perception System

Artificial fishes gain awareness of their world through sensory perception. As Fig. 3.3 suggests, it is necessary to model not only the abilities but also the limitations of animal perception systems in order to achieve natural sensorimotor behaviors. Hence, the artificial fish has a limited field of view extending frontally and laterally to an effective radius consistent with visibility in the translucent water (Fig. 3.3). An object may be detected only if some visible portion of it (i.e., not occluded behind some other opaque object) enters the fish's field of view (Fig. 3.3). The perception center of the artificial fish's brain (see Fig. 3.1) includes a perceptual attention mechanism which allows the animat to attend to the world in a task-specific way, hence filtering out sensory information superfluous to its immediate behavioral needs. For example, the artificial fish attends to sensory information about nearby food sources when foraging. The animats in our previous ALife simulations (described in [Terzopoulos *et al.*, 1994, Tu and Terzopoulos, 1994a, Tu and Terzopoulos, 1994b]) employ a *perceptual oracle* scheme according to which the

artificial fish may satisfy its perceptual needs via direct interrogation of the 3D world model. In particular, subject to the appropriate perceptual limitations, the animat's on-board sensors query the geometric and photometric information that is available to the graphics rendering engine, as well as object identity and dynamic state information within the physics-based virtual world.

We emphasize that our goal in this thesis is to replace the perceptual oracle with an artificial fish active vision system that elicits visual information from retinal images, as will be described in chapter 4.

3.3 Behavior System

The behavior center of the artificial fish's brain mediates between its perception system and its motor system (Fig. 3.1). A set of innate characteristics determines the (static) genetic legacy, which dictates whether the fish is male or female, predator or prey, etc. A (dynamic) mental state comprises variables representing hunger, fear, and libido, whose values depend on sensory inputs. The fish's cognitive faculty resides in the action selection component of its behavior center. At each simulation time step, action selection entails combining the innate characteristics, the mental state, and the incoming stream of sensory information to generate sensible, survival sustaining goals for the fish, such as to avoid an obstacle, to avoid predators, to hunt and feed on prey, or to court a potential mate. The action selector ensures that goals have some persistence by exploiting a single-item memory. The behavior memory reduces dithering, thereby improving the robustness of prolonged behaviors such as foraging, schooling, and mating. The action selector also controls the perceptual attention mechanism. At every simulation time step, the action selector activates behavior routines that attend to sensory information and compute the appropriate motor control parameters to carry the fish a step closer to fulfilling its immediate goals. The behavioral repertoire of the artificial fish includes primitive, reflexive behavior routines, such as obstacle avoidance, as well as more sophisticated motivational behavior routines such as schooling and mating whose activation is dependent on the

mental state.

3.4 Modeling Form and Appearance

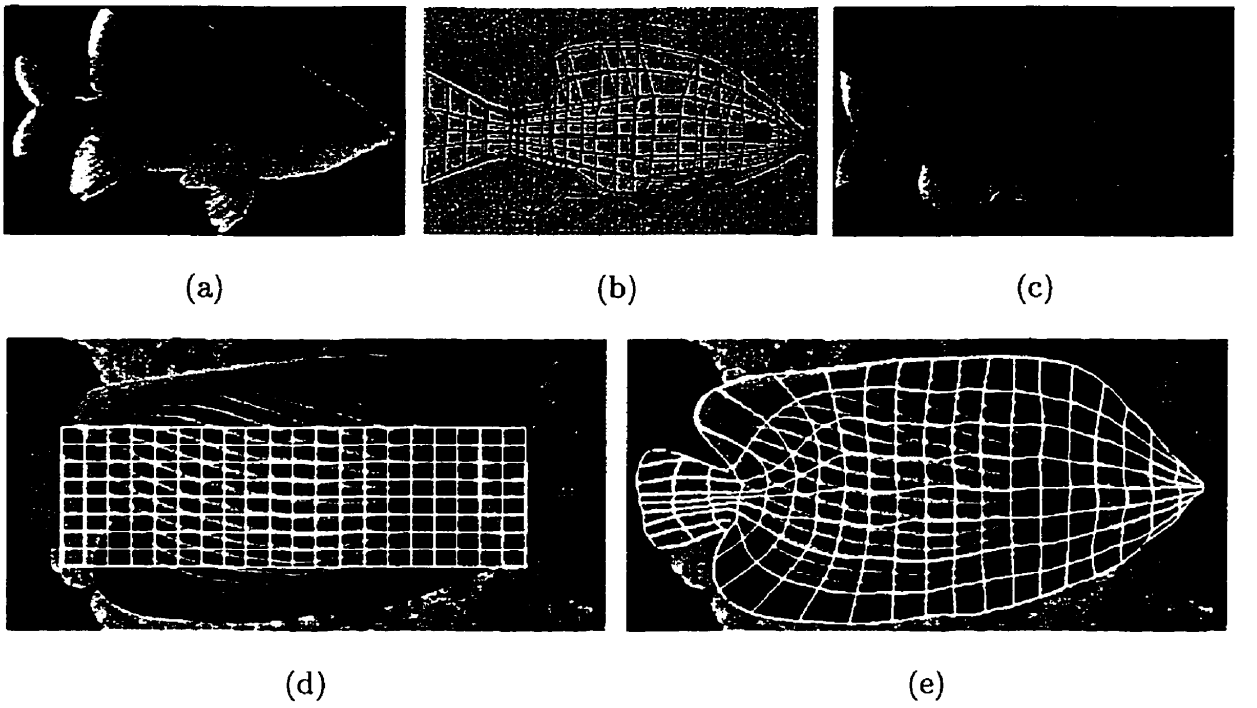


Figure 3.4: (a) Digitized color image of a fish photo. (b) 3D NURBS surface fish body. (c) Color texture mapped 3D fish model. Initial (d) and final (e) snake-mesh on an image of a different fish.

Active vision research currently employs mobile robots situated in natural real world environments. For active vision research to be appropriately applied to the animat vision framework, the artificial animal and its world must capture the form and appearance of real animals and their physical environment with considerable visual fidelity. To this end, photographs of real fishes, such as the one shown in Fig. 3.4(a), are converted into 3D spline (NURBS) surface body models (Fig. 3.4(b)). The digitized photographs are analyzed semi-automatically using deformable models [Terzopoulos *et al.*, 1988], in particular, a "snake-mesh" tool which is demonstrated in Fig. 3.4(d–e) on a different fish image. The snake mesh slides freely over the image and can be manipulated using the mouse. The border snakes adhere to intensity edges demarcating the fish from the background, and the remaining snakes relax elastically to cover the imaged fish body with

a smooth, nonuniform coordinate system (Fig. 3.4(e)). The coordinate system serves to map the appropriate image texture onto the spline surface to produce the final texture mapped fish body model (Fig. 3.4(c)).

Chapter 4

The Animat Vision System

In this chapter we present the animat vision system that we have developed for the artificial fish, which makes exclusive use of color retinal images.

4.1 Eyes and Retinal Imaging

The artificial fish has binocular vision. The movements of each eye are controlled through two gaze angles (θ, ϕ) which specify the horizontal and vertical rotation of the eyeball, respectively. The angles are measured with respect to the head coordinate frame, such that the eye is looking straight ahead when $\theta = \phi = 0°$.

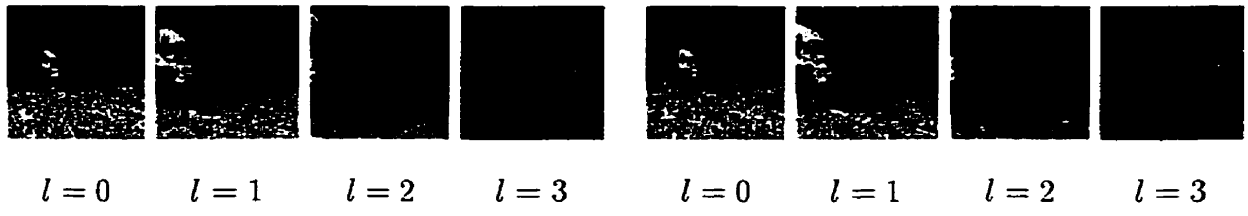
Each eye is implemented as four coaxial virtual cameras to approximate the spatially nonuniform, foveal/peripheral imaging capabilities typical of biological eyes. The level $l = 0$ camera has the widest field of view (about 120°) and the horizontal and vertical fields of view for the level l camera are related by

$$f_x^l = 2\tan^{-1}\left(\frac{d_x/2}{2^l f_c^0}\right); \qquad f_y^l = 2\tan^{-1}\left(\frac{d_y/2}{2^l f_c^0}\right), \tag{4.1}$$

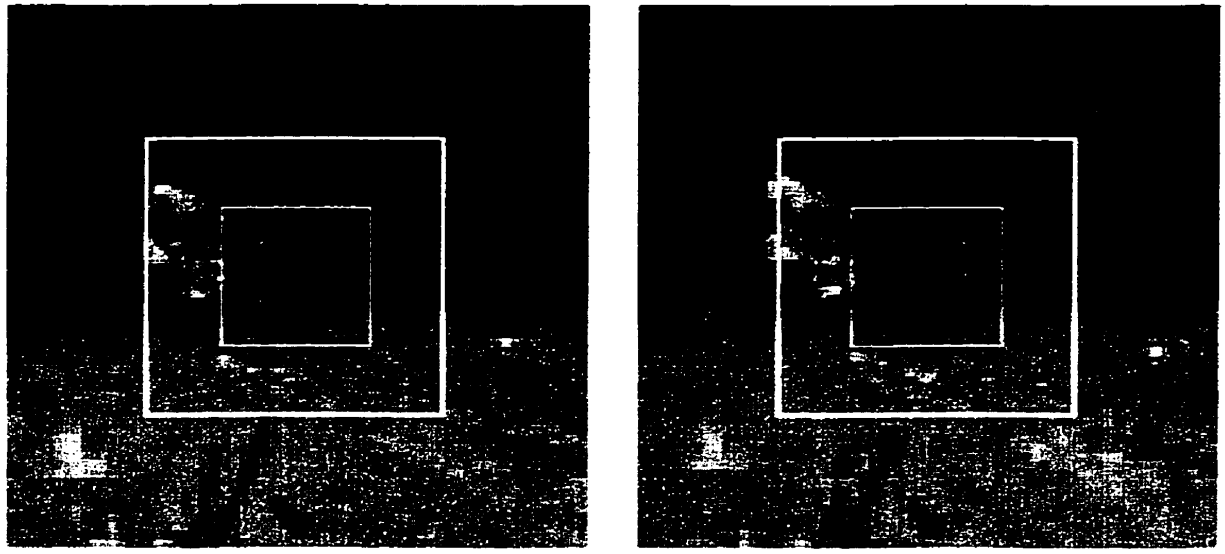
where d_x and d_y are the horizontal and vertical image dimensions and f_c^0 is the focal length of the wide field of view camera ($l = 0$).[1]

Fig. 4.1(a) shows an example of the 64×64 images that are rendered by the four coaxial cameras (using the OpenGL library and SGI graphics pipeline) of the left and right eye. Since the field of view decreases with increasing l, image l is a zoomed version

[1] If f_c^0 is unknown, but the $l = 0$ field of view is known, then f_c^0 is first computed using (4.1) with $l = 0$ and this value is used to specify the field of view at the other levels.



(a)



Left eye Right eye

(b)

Figure 4.1: Binocular retinal imaging. (a) 4 component images; $l = 0, 1, 2$, are peripheral images; $l = 3$ is foveal image. (b) Composited retinal images (borders of composited component images are shown in white).

of the central part of image $l-1$. We refer to the image at level $l = 3$ as the fovea and the others as peripheral images. We magnify the level l image by a factor of 2^{3-l} and overlay in sequence the four images coincident on their centers starting with the $l = 0$ image at the bottom (to form an (incomplete) pyramid), thus compositing a 512×512 retinal image with a 64×64 fovea at the center of a periphery with radially decreasing resolution (and increasing smoothing) in 3 steps. Fig. 4.1(b) shows the binocular retinal images composited from the coaxial images at the top of the figure. To reveal the retinal image structure in the figure, we have placed a white border around each magnified component image.

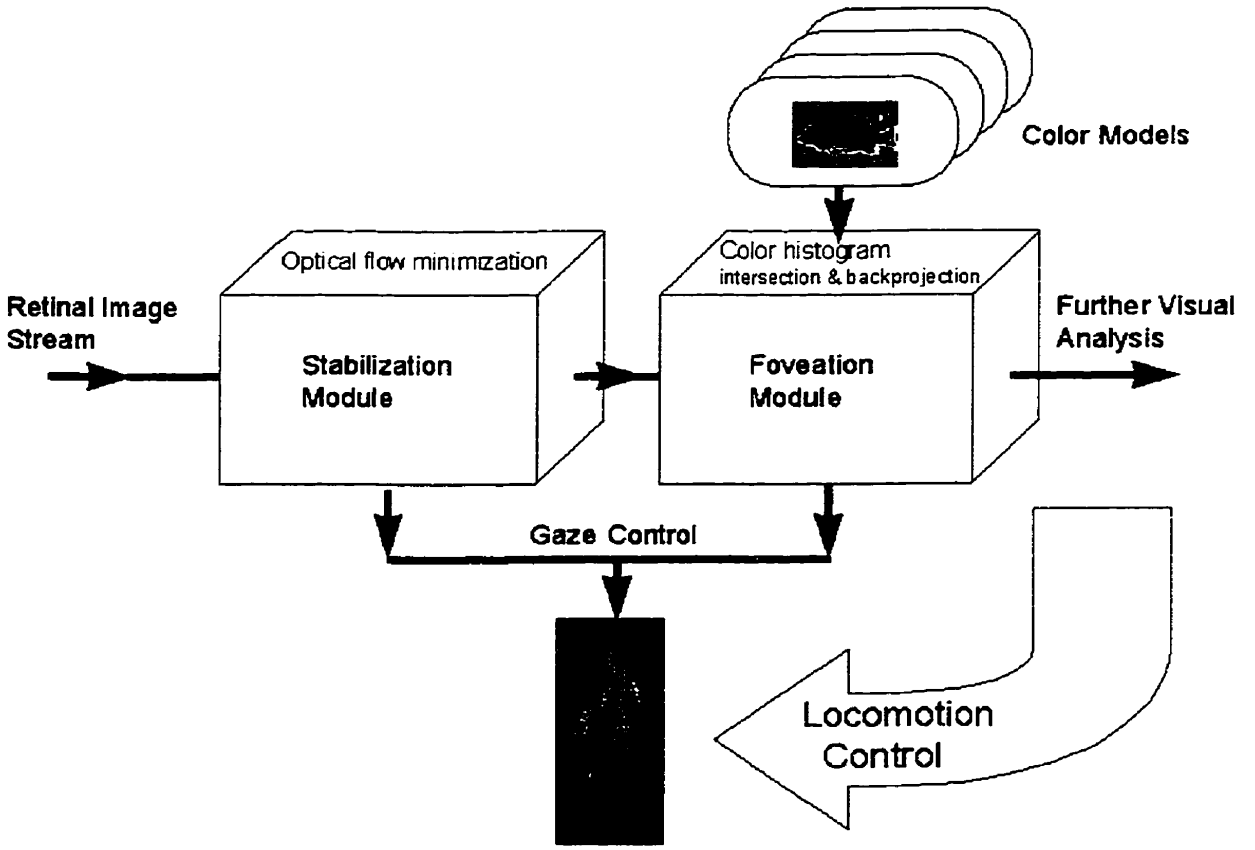


Figure 4.2: Gaze control for the animat vision system.

The advantages of the multiresolution retina are significant. Vision algorithms which process the four 64×64 component images could be 16 times more efficient than those that would have to process a uniform 512×512 retinal image.

4.2 Active Vision System Overview

Fig. 4.2 illustrates a block diagram of one ocular channel of the binocular animat vision system. The system consists of two main modules—a *foveation* module and *stabilization* module. Together they implement a gaze control capability that enables the artificial fish to stabilize the visual world as it locomotes, as well as to detect a visual target in its field of view, foveate the target, and visually navigate towards the target. If the target is in motion, the artificial fish can track it visually and swim in pursuit.

4.3 Foveation using Color Object Detection

The mind of the fish stores a set of color models of objects that are of interest to it. For instance, if the fish is a predator, it would possess models of prey fish. The models are stored as a list of 64×64 color images in the fish's memory.

We have adopted into the active vision system the color histogram methods of Swain [Swain and Ballard, 1991]. The fish employs modified versions of these methods to detect and localize any target that may be imaged in the low resolution periphery of its retinas. Since each model object has a unique color histogram, a target with a similar color histogram can be detected in the periphery by histogram intersection and localized by histogram backprojection.

4.3.1 Modified Color Histogram Intersection Method

Swain [Swain and Ballard, 1991] developed a technique called color indexing that efficiently identifies objects in a database in the presence of occlusion and over changes in viewpoint. He demonstrated that object color distributions without geometric information can provide a powerful cue for recognition.

The effectiveness of the algorithm degrades badly if the area of the object in the model image differs substantially from the area of the target object appearing in the image. Swain suggests scaling the initial model histogram by d_M^2/d^2 where d_M is the known range of the model initially and d is the computed range of the target object at the time of backprojection. We will show later that estimating range is straightforward with the eyes verged on a target. Unfortunately, this scaling technique did not work well for the artificial fish, apparently because of noisy depth measurements and the perspective nonlinearity associated with the wide field of view cameras.

We developed a more robust intersection measure that is invariant to scale changes. This new method iteratively scales down an initially large model color histogram to the approximate size of the target object appearing in the image.

Following Swain's notation in [Swain and Ballard, 1991], his original histogram inter-

section measure is

$$H = \frac{\sum_{j=1}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} M_j}, \tag{4.2}$$

where I is the image color histogram (I_j is the number of image pixels classified into histogram bin j) , M is the model color histogram, and n is the number of histogram bins used. This measure is effective only if the model histogram is properly scaled. To overcome this limitation, note that the match value H gives the percentage of pixels from the model that have corresponding pixels of the same color in the image (hence, $H = 0.9$ means that there is 90% chance the model appeared in the image; however, the value of H can drop significantly; e.g. $H = 0.1$ if there is a scale difference between target and model). This suggests that we can use H itself to scale the model histogram M. Our experiments revealed that this is not always effective, but that it can be improved by scaling the histogram iteratively; i.e., recomputing H after every scaling of M until the value of H increases above a set threshold, say 0.9. This technique may be expressed as

$$M_k^{i+1} = M_k^i H^i = M_k^i \frac{\sum_{j=1}^{n} \min(I_j, M_j^i)}{\sum_{j=1}^{n} M_j^i}; \quad k = 1, \ldots, n, \tag{4.3}$$

where i is the iteration number. Equation 4.3 is iterated until the value of H^i either exceeds the threshold, indicating the presence of the model in the image, or remains constant below threshold (or decreases), indicating that the model matches nothing in the image. The equation converges after a few iterations (usually 2 to 4 if the target size is not too much smaller than the model).

The iterative technique may degenerate in cases when the model is not present in the image, while a similar color combination is. The problem is that the model histogram gets scaled to the size of the false target to yield a large intersection match value, hence a false alarm.

To overcome this problem, we employ a new intersection measure after scaling the model histogram using (4.3). Our measure makes use of a weighted histogram intersection method inspired by the local histogram method proposed by Ennesser and Medioni

[Ennesser and Medioni, 1993]. Our measure is

$$H_N = \frac{\sum_{j=1}^{n} W_j \min(I_j, M_j)}{\sum_{j=1}^{n} M_j}, \tag{4.4}$$

where the weighting histogram W is given by $W_j = M_j/2^l P_j$. Here P is the color histogram of the peripheral image (at level $l = 0$). As is noted by Ennesser and Medioni, the weighted intersection technique increases the selectivity of the method by placing more importance on colors that are specific to the model. In our experiments, H_N provided very good separation between intersection match values for false targets ($H_N < 0.2$) and true targets ($H_N > 0.8$).

An alternative method, which also gives good results is to incorporate the weighting histogram inside the iteration of equation (4.3) as follows:

$$M_k^{i+1} = M_k^i \frac{\sum_{j=1}^{n} (M_j^i/2^l P_j) \min(I_j, M_j^i)}{\sum_{j=1}^{n} M_j^i}. \tag{4.5}$$

The scaled M is then used iteratively to compute the intersection match value H_N as before.

4.3.2 Localization using Color Histogram Backprojection

Once the model histogram has been properly scaled as described above, Swain's backprojection algorithm works well in localizing the pixel position of the center of the detected target in the foveal image.

Histogram backprojection gives large weights to pixel locations in the image whose color histogram closely resembles the color histogram of the model. It thus answers the question: Where are the colors in the image that belong to the target object being observed? The answer is given such that colors appearing in other objects besides the target are deemphasized so that the search may be focused on finding the actual target.

In histogram backprojection the ratio histogram R_j is defined as

$$R_j = \frac{M_j}{I_j}, \quad j = 1, 2, ..., n. \tag{4.6}$$

It is this histogram R that is backprojected onto the image. The image values are replaced by the values of R that they index. This can be represented as

$$c_{x,y} = R(\text{map}(c_{x,y})), \tag{4.7}$$

where $c_{x,y}$ is the image color value at pixel location (x, y), and map(c) is a histogram function that maps a three dimensional color value c to a three dimensional histogram bin. The backprojected image is then convolved with a circular disk of area equal to the expected area of the target in the image as

$$c_{x,y} = D^r * c_{x,y}, \tag{4.8}$$

where D^r is the disk of radius r. The peak in the convolved image gives the expected (x, y) location of the target in the image. A thorough description of this algorithm is available in [Swain and Ballard, 1991].

The color space that we used in the animat vision system's color recognition algorithms was the HSV (Hue, Saturation, Intensity) color space. It was chosen empirically to allow the intensity axis V to be more coarsely sampled than the H and S axes. The intensity axis V is more sensitive to lighting variations from shadows and distance from light source than the H, S axes, so blurring V more improves the robustness of the system to lighting changes in the acquired scene. In our experiments, The H and S axes were divided into 32 sections, while the V axis was divided into only 16 sections for a total of 16384 bins.

4.3.3 Saccadic Eye Movements

When a target is detected in the visual periphery, the eyes will saccade to the angular offset of the target to bring it within the fovea. With the object in the high resolution fovea, a more accurate foveation is obtained by a second pass of histogram backprojection. A second saccade typically centers the object accurately in both left and right foveas, thus achieving vergence.

When the fish is executing a rapid turn, however, the target could partially exit the fovea ($l = 3$). Part of it will appear in the next coarser image ($l = 2$). Three saccades are then typically used to regain a foveal fix on the target. The first saccade detects the portion of the target still in the fovea and makes an initial attempt to foveate the target, on the second saccade the target is brought closer to the center of the fovea, and finally the third saccade accurately centers the target in both foveas to verge the eyes.

If the turn is too rapid such that the target leaves the fovea entirely, the histogram intersection method will fail to detect the target in the fovea. The algorithm automatically reapplies the intersection of the model with the image at a lower level with a wider field of view. Thus, the algorithm will continuously move down the pyramid trying to detect the target. When it is eventually detected and localized, it is foveated at multiple saccades of the eyes to center it inside the high resolution fovea at $l = 3$.

Saccadic eye movements are performed by incrementing the gaze angles (θ, ϕ) with differential angles $(\Delta\theta, \Delta\phi)$ in order to rotate the eyes to the required gaze direction. When the pixel location of the target computed from the left or right images at level l is (x_c, y_c), the correction gaze angles for the eye are given by

$$\Delta\theta = \tan^{-1}\left(\frac{x_c}{2^l f_c}\right); \qquad \Delta\phi = \tan^{-1}\left(\frac{y_c}{2^l f_c}\right). \tag{4.9}$$

If the target object comes too near the eyes and fills the entire fovea, the algorithm detects this condition and foveates the target at the next coarser level (e.g., $l = 2$), where the field of view is broader and the target has a more reasonable size for detection and localization. Note that (4.9) computes the correction angles at level l, but the same corrected (θ, ϕ) are used to render all the other levels.

4.4 Visual Field Stabilization using Optical Flow

It is necessary to stabilize the visual field of the artificial fish because its body undulates as it swims. The optokinetic reflex in animals stabilizes vision by measuring image motion and producing compensatory eye movements. Once a target is verged in both

foveas, the stabilization process assumes the task of keeping the target foveated as the fish locomotes.

Stabilization is achieved by computing the displacement (u, v) between the current and previous foveal images and updating the gaze angles to $(\theta + \Delta\theta, \phi + \Delta\phi)$. The displacement is computed as a translational offset in the retinotopic coordinate system by a least squares minimization of the optical flow constraint equation between image frames at times t and $t - 1$ [Burt *et al.*, 1989, Irani *et al.*, 1994]. Given a sequence of time-varying images, points imaged on the retina appear to move because of the relative motion between the eye and objects in the scene [Gibson, 1979]. The instantaneous velocity vector field of this apparent motion is usually called optical flow [Ballard and Brown, 1982]. The optical flow constraint equation is given by [Horn, 1986]

$$uI_x + vI_y + I_t = 0, \tag{4.10}$$

where $(I_x, I_y, I_t)^T$ is the spatiotemporal intensity gradient given as $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t})^T$. Values of (u, v) satisfying this constraint equation lie on a straight line in velocity space. The image intensity is computed as

$$I(x, y, t) = \frac{1}{3}[R(x, y, t) + G(x, y, t) + B(x, y, t)], \tag{4.11}$$

where R, G, and B denote the color component channels. The error function

$$E(u, v) = \sum_{x,y \in \text{fovea}} (uI_x + vI_y + I_t)^2 \tag{4.12}$$

is minimized by simultaneously solving the two equations $\partial E/\partial u = 0$ and $\partial E/\partial v = 0$ for the image displacement (u, v).

The correction angles $(\Delta\theta, \Delta\phi)$ for a displacement of (u, v) between the images at level l are computed using (4.9) by replacing (x_c, y_c) with (u, v). If the displacement computed from the foveal images (at level $l = 3$) is too small (indicating the target is close enough to fill the fovea), the algorithm stabilizes at the next lower level where the target does not fill the entire image area.

The flow constraint displacement estimation method is accurate only for small displacements between frames. Consequently, when the displacement of the target between frames is large enough that the method is likely to produce bad estimates, the fix is regained by invoking the foveation module to re-detect and re-foveate the target as described earlier.

Each eye is controlled independently during foveation and stabilization of a target. Hence, the two eyes must be correlated to keep them verged accurately on the target and not drifting in different directions. The correlation is performed by computing the displacement (u, v) between the left and right foveal images (at $l = 3$), and correcting the gaze angles of the right eye to $(\theta_R + \Delta\theta_R, \phi_R + \Delta\phi_R)$ using (4.9).

Once the eyes are verged on a target, it is straightforward for the active vision system to estimate the range to the target from the gaze angles. Referring to Fig. 4.3, the range is [Horn, 1986]

$$d = b\frac{\cos(\theta_R)\cos(\theta_L)}{\sin(\theta_L - \theta_R)\cos(\theta_P)}, \tag{4.13}$$

where b is the baseline between the two eyes, and $\theta_P = \frac{1}{2}(\theta_R + \theta_L)$ is the left/right turn angle. When the eyes are verged on a target the vergence angle is $\theta_V = (\theta_L - \theta_R)$ and its magnitude increases as the observer comes closer to the target [Brown *et al.*, 1992].

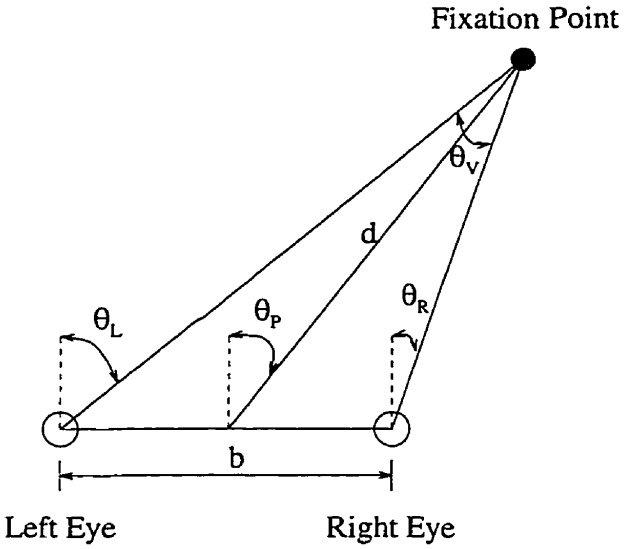


Figure 4.3: Gaze angles and range to target geometry.

4.5 Vision-Guided Navigation

The artificial fish can employ the direction of gaze of its eyes to navigate effectively in its world. In particular, it is natural to use the gaze angles as the eyes are fixated on a target to navigate towards the target. The θ angles are used to compute the left/right turn angle θ_P shown in Fig. 4.3, and the ϕ angles are similarly used to compute an up/down turn angle $\phi_P = \frac{1}{2}(\phi_R + \phi_L)$. The fish's turn motor controllers are invoked to execute a left/right turn—`left-turn-MC` for negative θ_P and `right-turn-MC` for positive θ_P (see Section 3)—with $|\theta_P|$ as parameter when $|\theta_P| > 30°$. An up/down turn command is issued to the fish's pectoral fins if $|\phi_P| > 5°$, with a positive ϕ_P interpreted as up and negative as down.

4.6 Pursuit of Nonrigid Targets in Motion

The problem of pursuing a moving target that has been fixated in the foveas of the artificial fish's eyes is simplified by the gaze control mechanism described above. The fish can robustly foveate a moving target and chase it by using the turn angles (θ_P, ϕ_P) computed from the gaze angles that are continuously updated by the foveation/stabilization algorithms.

We have carried out numerous experiments in which the moving target is a reddish fish whose color histogram model is stored in the memory of a predator fish equipped with the active vision system. Fig. 4.4 shows plots of the gaze angles and the turn angles obtained over the course of 100 frames in a typical experiment as the predator fixates on and actively pursues a prey target. Fig. 4.5 shows a sequence of image frames acquired by the observer fish during its navigation (only the left retinal images are shown). Frame 0 shows the target visible in the low resolution periphery of the fish's eyes (middle right). Frame 1 shows the view after the target has been detected and the eyes have saccaded to foveate the target (note that the size decrease of the target after foveation is a perspective effect). The subsequent frames show the target remaining fixated in the fovea despite the

side-to-side motion of the fish's body as it swims towards the target. Fixation is achieved by stabilizing the eyes with compensating optokinetic signals. The signals are indicated in Fig. 4.4 by the undulatory responses of the θ angles.

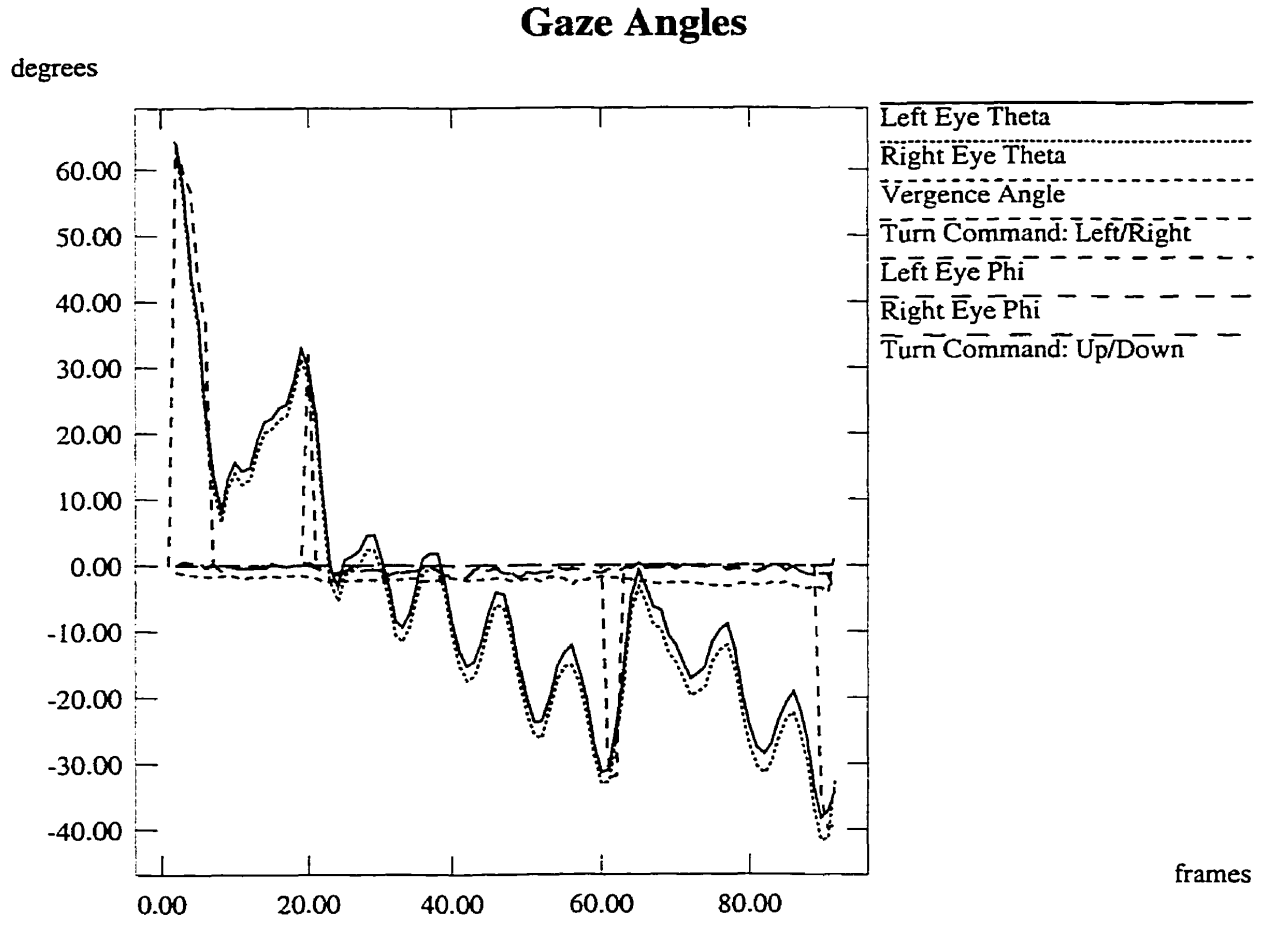


Figure 4.4: Gaze angles (saccade signals) vs time (frames) of the observer fish while pursuing the reddish target fish.

Fig. 4.4 shows that the vergence angle tends to increase in magnitude as the fish moves closer to the target (around frame 100). In comparison to the θ angles, the ϕ angles show little variation, because the fish does not undulate vertically very much as it swims forward. It is apparent from the graphs that the gaze directions of the two eyes are nicely correlated; that is, (θ_R, ϕ_R) follows (θ_L, ϕ_L), with $\theta_L - \theta_R$ indicating the reciprocal of range to the target.

Notice that in frames 87–117 of Fig. 4.5, a yellow fish whose size is similar to the target

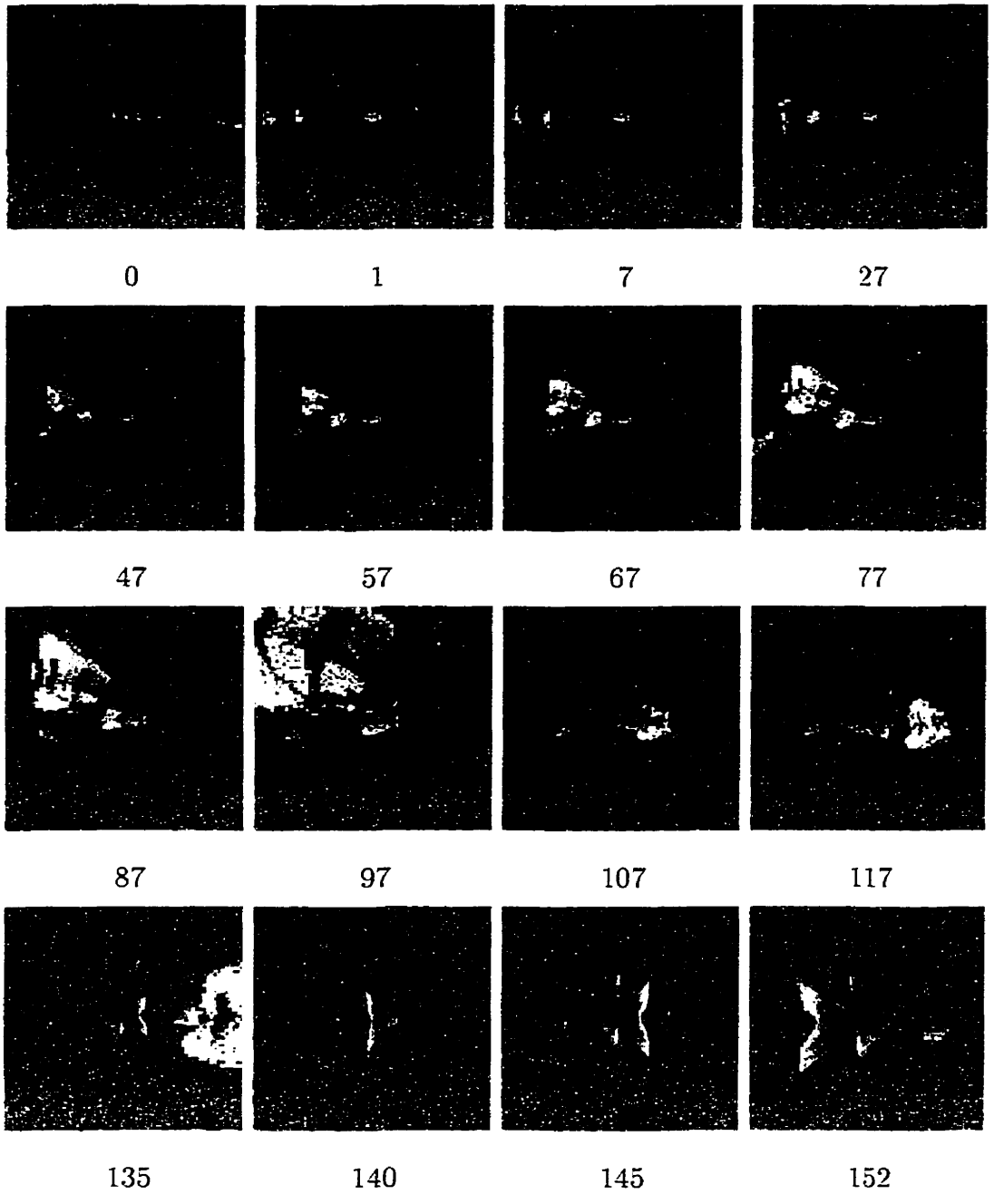


Figure 4.5: Retinal image sequence from the left eye of the active vision fish as it detects and foveates on a reddish fish target and swims in pursuit of the target. The target appears in the periphery (middle right) in frame 0 and is foveated in frame 1. The target remains fixated in the center of the fovea as the fish uses the gaze direction to swim towards it (frames 7–117). The target fish turns and swims away with the observer fish in visually guided pursuit (frames 135–152).

fish passes behind the target. In this experiment the fish with active vision was instructed to treat all non-reddish objects as totally uninteresting and not worth foveating. Because of the color difference, the yellow object does not distract the fish's gaze from its reddish target. This demonstrates the robustness of the color-based fixation algorithm.

4.7 Summary

This chapter described the implementation of a prototype animat vision system within lifelike artificial fishes inhabiting a physics-based, virtual marine world. The fishes emulate the appearance, motion, and behavior of natural fishes in their physical habitats. In a relatively short period of time we were able to implement successfully within the framework of the artificial fish animat a set of active vision algorithms for foveation and vergence of interesting targets, for retinal image stabilization, and for pursuit of moving targets through visually-guided navigation. Note, however, that the automated analysis of the class of retinal images that confront our vision algorithms is by no means easy.

The next two chapters extend the animat vision system thus enhancing the animat's functionality in its virtual environment. In Chapter 5, motion and color cues are integrated to increase the robustness of the animat's perceptual functions. Chapter 6 adds dynamic obstacle recognition and avoidance capabilities by exploiting stereo and color cues naturally available from the photorealistic images acquired by the animat's binocular eyes. These chapters will also present an example of how vision algorithms may be evaluated within the proposed animat vision framework.

Chapter 5

Motion and Color Analysis for Animat Perception

The high-acuity fovea in many biological eyes covers only a small fraction of a visual field whose resolution decreases monotonically towards the periphery. Spatially nonuniform retinal imaging provides opportunities for increased computational efficiency through economization of photoreceptors and focus of attention, but it forces the visual system to solve problems that do not generally arise with a uniform field of view. A key problem is determining where to redirect the fovea when a target of interest appears in the periphery. In this chapter we present a solution to this problem through the exploitation of motion and color information.

Motion and color play an important role in animal perception. Birds and insects exploit optical flow for obstacle avoidance and to control their ego-motion [Gibson, 1979, Horridge, 1993]. Some species of fish are able to recognize the color signatures of other fish and use this information in certain piscene behaviors [Adler, 1975]. The human visual system is highly sensitive to motion and color. We tend to focus our attention on moving colorful objects. Motionless objects whose colors blend in to the background are not as easily detectable, and several camouflage strategies in the animal kingdom rely on this fact [Cedras and Shah, 1995].

Following the animat vision paradigm, the motion and color based gaze control algorithms that we present in this chapter are implemented and evaluated within the animat vision framework. Our new gaze control algorithms significantly enhance the prototype

animat vision system implemented in Chapter 4 and they support more robust vision-guided navigation abilities in the artificial fish.

5.1 Integrating Motion and Color for Attention

Selective attention is an important mechanism for dealing with the combinatorial aspects of search in vision [Tsotsos *et al.*, 1995]. Deciding where to redirect the fovea can involve a complex search process [Tsotsos *et al.*, 1995, Ye and Tsotsos, 1995, Rimey and Brown, 1994, Maver and Bajcsy, 1990]. In this section we propose an efficient solution which integrates motion and color to increase the robustness of our animat's perceptual functions.

Motion and color have been considered extensively in the literature in a variety of passive vision systems [Wixson and Ballard, 1989, Dubuisson and Jain, 1994b, Wang and Adelson, 1993, Weber and Malik, 1993, Campani *et al.*, 1995, Cedras and Shah, 1995], but rarely have they been integrated for use in dynamic perception systems. The conjunction of color and motion cues has recently been exploited to produce more exact segmentations and for the extraction of object contours from natural scenes [Dubuisson and Jain, 1993, Dubuisson and Jain, 1994a]. Color and motion features of video images have been used for color video image classification and understanding [Gong and Sakauchi, 1992].

Integrating motion and color for object recognition can improve the robustness of moving colored object recognition. Motion may be considered a bottom-up *alerting* cue, while color can be used as a top-down cue for model-based recognition [Swain *et al.*, 1992]. Therefore, integrating motion and color can increase the robustness of the recognition problem by bridging the gap between bottom-up and top-down processes, thus, improving the selective attention of dynamic perceptual systems such as the animat vision system that we have developed.

5.1.1 Where to Look Next

Redirecting gaze when a target of interest appears in the periphery can be a complex problem. One solution would be to section the peripheral image into smaller patches or *focal probes* [Burt *et al.*, 1989] and search all the probes. The strategy will work well for sufficiently small images, but for dynamic vision systems that must process natural or photorealistic images the approach is not effective.

We choose a simple method based on motion cues to help narrow down the search for a suitable gaze direction [Campani *et al.*, 1995]. We create a saliency image by initially computing a reduced optical flow field between two stabilized peripheral image frames (an advantage of the multiresolution retina is the small 64×64 peripheral image). Then an affine motion model is fitted to the optical flow using a robust regression method that will be described momentarily. The affine motion parameters are fitted to the dominant background motion. A saliency map is determined by computing an error measure between the affine motion parameters and the estimated optical flow as follows:

$$S(x,y) = \sqrt{[u_a(x,y) - u(x,y)]^2 + [v_a(x,y) - v(x,y)]^2}, \tag{5.1}$$

where (u, v) is the computed optical flow and

$$\begin{aligned} u_a(x,y) &= a + bx + cy, \\ v_a(x,y) &= d + ex + fy \end{aligned} \tag{5.2}$$

is the affine flow at retinal image position (x, y) which is used to describe the motion of a planar surface relative to the camera. The saliency image S is then convolved with a circular disk of area equal to the expected area of the model object of interest as it appears in the peripheral image.[1]

The blurring of the saliency image emphasizes the model object in the image. The maximum in S is taken as the location of the image probe. The image patches that serve

[1]Reasonably small areas suffice, since objects in the 64×64 peripheral image are typically small at peripheral resolution. Methods for estimating appropriate areas for the object, such as Jagersand's information theoretic approach [Jagersand, 1995], may be applicable.

283 284 285 286 Saliency Image

Figure 5.1: Four consecutive peripheral images with image probes outlined by white squares. Saliency image (right), with bright areas indicating large motions.

as focal probes in consecutive peripheral frames form the image sequence that is processed by the motion segmentation module described later. Fig. 5.1 shows four consecutive peripheral images with the image probes outlined by white boxes. The blurred saliency image is shown at the end of the sequence in Fig. 5.1. The maximum (brightness) corresponds to a fast moving blue fish in the middle right portion of the peripheral image (inside the white borders).

5.1.2 Robust Optical Flow

A key component of the selective attention algorithm is the use of optical flow. Optical flow can provide important information about the spatial arrangement of objects viewed and the rate of change of this arrangement [Horn, 1986]. Various techniques for determining optical flow from a sequence of two or more frames have been proposed in the literature [Horn and Schunck, 1981, Anandan, 1989, Lucas and Kanade, 1981, Horn and Jr., 1988, Bergen *et al.*, 1992, Fleet and Jepson, 1990, Black and Anandan, 1993, Jepson and Black, 1993, Spetsakis, 1994]. Optical flow, once reliably estimated can be very useful in various computer vision applications. Discontinuities in the optical flow can be used in segmenting images into moving objects [Adiv, 1985, Burt *et al.*, 1989, Peleg and Rom, 1990, Irani and Peleg, 1992]. Navigation using optical flow and estimation of time-to-collision maps have been discussed in [Hagen and Heyerdahl, 1992, Campani *et al.*, 1995] and [Meyer and Bouthemy, 1992].

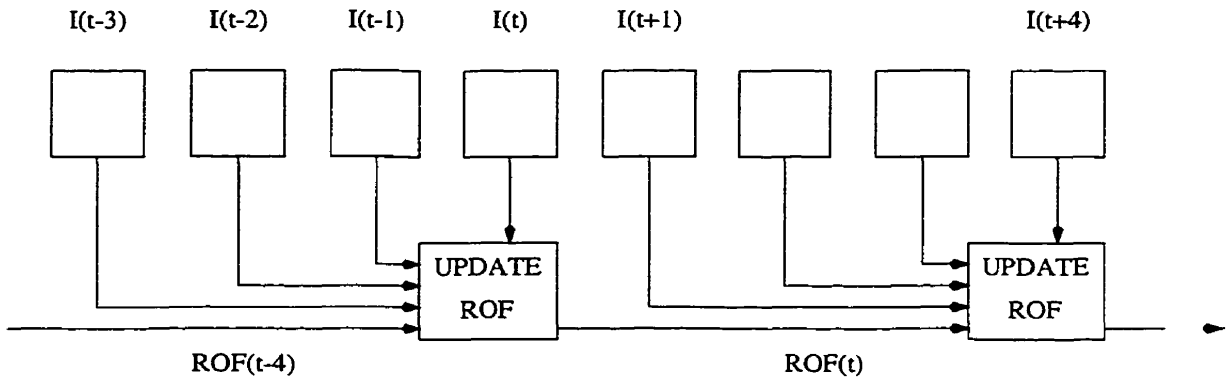


Figure 5.2: Incremental estimation of robust optical flow (ROF) over time.

For our specific application, however, we require efficiency, robustness to outliers, and an optical flow estimate at all times. Recent work by Black and Anandan [Black and Anandan, 1990, Black and Anandan, 1993] satisfies our requirements. They propose incremental minimization approaches using robust statistics for the estimation of optical flow which are geared towards dynamic environments. As is noted by Black, the goal is incrementally to integrate motion information from new images with previous optical flow estimates to obtain more accurate information about the motion in the scene over time. A detailed description of this method can be found in [Black, 1992]. Here we describe our adaptation of the algorithm to the animat vision system.

Ideally optical flow is computed continuously[2] as the animat navigates in its world, but to reduce computational cost and to allow for new scene features to appear when no interesting objects have attracted the attention of the animat, we choose to update the current estimate of the optical flow every four frames. The algorithm is however still "continuous" because it computes the current estimate of the optical flow at time t using image frames at $t-3, t-2, t-1$, and t in a short-time batch process. Fig. 5.2 shows this more clearly. This arrangement requires storage of the previous three frames for use by the estimation module.

The flow at $t+1$ is initialized with a predicted flow computed by forward warp of the optical flow estimate at t by itself [3] and then the optical flow at $t+4$ is estimated

[2]By continuously, we mean that there is an estimate of the optical flow at every time instant.

[3]The optical flow estimate is being used to warp itself, thus predicting what the motion will be in

by spatiotemporal regression over the four frames.

We compute our optical flow estimate by incrementally minimizing the cost function

$$E(u,v) = \lambda_D E_D(u,v) + \lambda_S E_S(u,v) + \lambda_T E_T(u,v), \tag{5.3}$$

where E_D is a data conservation constraint, E_S is a spatial coherence constraint, and E_T is a temporal continuity constraint.

The data conservation constraint is derived from the observation that surfaces generally persist in time and, therefore, the intensity structure of a small region in one image remains constant over time, although its position may change [Horn, 1986]. We formulate our data conservation constraint based on this assumption of intensity constancy within a small region, in terms of the optical flow constraint equation for a given pixel location as

$$E_D(u,v) = \rho(uI_x + vI_y + I_t, \sigma_D). \tag{5.4}$$

The spatial coherence constraint embodies the assumption that surfaces have spatial extent and, hence, neighboring pixels in an image are likely to belong to the same surface. Since the motion of neighboring points on a smooth rigid body changes gradually, a smoothness constraint can be enforced on the motion of neighboring points in the image plane [Horn and Schunck, 1981] and E_S can be given as

$$E_S(u,v) = \sum_{m,n \in N} [\rho(u - u(m,n), \sigma_S) + \rho(v - v(m,n), \sigma_S)], \tag{5.5}$$

where N is the local neighborhood to the current pixel position (typically taken to be the 4-connected neighbors).

We formulate our temporal continuity constraint E_T by imposing some coherence between the current flow estimate and its previous and next estimate:

$$E_T(u,v) = \rho(\mathbf{u} - \mathbf{u}_{BW}, \sigma_T) + \rho(\mathbf{u} - \mathbf{u}_{FW}, \sigma_T), \tag{5.6}$$

where $\mathbf{u} = (u,v)$ is the current optical flow estimate at time t, $\mathbf{u}_{BW}$ is the previous estimate at $t-1$ obtained by setting it to the most recent estimate, and $\mathbf{u}_{FW}$ is a

the future.

prediction of what the optical flow will be at $t+1$ and is computed by forward warp of the current estimate by itself.[4] The λ parameters in (5.3) control the relative importance of the terms, and the $\rho(x,\sigma)$ functions in the above equations are taken to be the Lorentzian robust estimator:

$$\rho(x,\sigma) = log\left(1+\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right), \tag{5.7}$$

where σ is a parameter in the robust estimator formulation that controls the outlier rejection point. Reducing σ will cause the estimator to reject more measurements as outliers and visa versa. The influence function, $\psi(x,\sigma)$, is the first derivative of $\rho(x,\sigma)$ with respect to x:

$$\psi(x,\sigma) = \frac{2x}{2\sigma^2+x^2}. \tag{5.8}$$

This function characterizes the bias that a particular measurement has on the solution [Hampel, 1974, Black and Anandan, 1993]. The choice of a particular robust estimator depends on the the optimization scheme used to minimize the cost function. The scheme we use requires that the estimator be twice differentiable. The second partial derivative of $\rho(x,\sigma)$ is

$$\frac{\partial\psi(x,\sigma)}{\partial x} = \frac{2(2\sigma^2-x^2)}{(2\sigma^2+x^2)^2}. \tag{5.9}$$

An upper bound on this second partial derivative is obtained when $x = 0$ in equation (5.9).

This robust formulation of our cost function E causes it to be non-convex. A local minimum can, however, be obtained using a gradient-based optimization technique. We choose the successive over-relaxation minimization technique because of its rapid convergence and well defined theory.

The iterative equations for minimizing E for a single pixel location are

$$\begin{aligned} u^{i+1} &= u^i - \frac{\mu}{T_u}\frac{\partial E}{\partial u}, \\ v^{i+1} &= v^i - \frac{\mu}{T_v}\frac{\partial E}{\partial v}, \end{aligned} \tag{5.10}$$

[4]Note that $\mathbf{u}_{BW}$ can also be estimated by backward warping of $\mathbf{u}$ by itself.

where $1 < \mu < 2$ is an overrelaxation parameter that controls convergence, and is used to overcorrect the estimate at the next iteration step $i + 1$, thus anticipating future corrections.[5]

The terms T_u, T_v are upper bounds on the second partial derivatives of E, and can be given as

$$\begin{aligned} T_u &= \frac{\lambda_D I_x^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2} + \frac{2\lambda_T}{\sigma_T^2}, \\ T_v &= \frac{\lambda_D I_y^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2} + \frac{2\lambda_T}{\sigma_T^2}, \end{aligned} \tag{5.11}$$

The partial derivatives in (5.10) are

$$\begin{aligned} \frac{\partial E}{\partial u} &= \lambda_D I_x \psi(uI_x + vI_y + I_t, \sigma_D) + \\ &\quad \lambda_S \sum_{m,n \in N} \psi(u - u(m,n), \sigma_S) + \\ &\quad \lambda_T[\psi(u - u_{BW}, \sigma_T) + \psi(u - u_{FW}, \sigma_T)], \\ \\ \frac{\partial E}{\partial v} &= \lambda_D I_y \psi(uI_x + vI_y + I_t, \sigma_D) + \\ &\quad \lambda_S \sum_{m,n \in N} \psi(v - v(m,n), \sigma_S) + \\ &\quad \lambda_T[\psi(v - v_{BW}, \sigma_T) + \psi(v - v_{FW}, \sigma_T)]. \end{aligned} \tag{5.12}$$

The above minimization will generally converge to a local minimum. A global minimum may be found by constructing an initially convex approximation to the cost function by choosing initial values of the σ parameters to be sufficiently large (equal to the maximum expected outlier in the argument of $\rho(x)$), effectively blurring the cost function E. The minimum is then tracked using the graduated non-convexity (GNC) continuation method as described by Blake and Zisserman [Blake and Zisserman, 1987] by decreasing the values of the σ parameters from one iteration to the next, which serves to gradually return the cost function to its non-convex shape, thereby introducing discontinuities in the data, spatial, and temporal terms. These discontinuities are, however, dealt with by the robust formulation and are rejected as outliers, thus producing more accurate optical

[5]Successive over-relaxation is Newton-Raphson's minimization technique when $\mu = 1$ and $T_x = \frac{\partial^2 E}{\partial x^2}$.

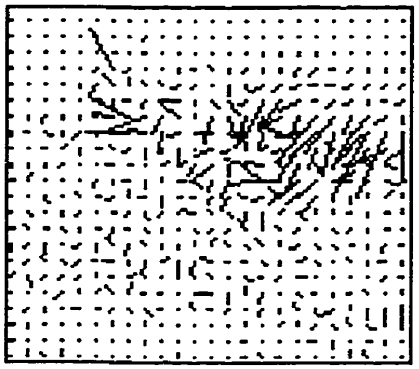


Figure 5.3: The robust optical flow vectors estimated for the four image probe sequence. Large vectors indicate large motion of the fish object.

flow estimates. The values of the λ parameters are determined empirically (typically $\lambda_D = 10, \lambda_S = \lambda_T = 1$).

To deal with large motions in the image sequence, we perform the minimization using a coarse-to-fine flow-through strategy. A Gaussian pyramid [Burt and Adelson, 1983] is constructed for each image in the sequence, and minimization starts at the coarsest level and flows through to the finest resolution level. Our flow-through technique is based on the assumption that displacements which are less than 1 pixel are estimated accurately at each individual level and thus need not be updated from a coarser level's estimate, while estimates that are greater than 1 pixel are most probably more accurately computed at the coarser level, and are updated by projecting the estimate from the coarser level.

This incremental minimization approach foregoes a large number of relaxation iterations over a 2 frame sequence in favor of a small number of relaxation iterations over a longer sequence. This satisfies our need for speed and any-time access and causes the flow estimate to be improved and refined gradually as more retinal images are acquired. Fig. 5.3 shows the optical flow estimated for the sequence of four image probes of Fig. 5.1. The figure clearly shows the complex motion of the target fish. It is a non-trivial task to segment such motions.

It must be stressed that for a reliable optical flow estimate to be obtained the retinal image stream must have been rotationally stabilized a priori against the animat's undulation. To facilitate this, a number of objects in the virtual world of the animat (which can

include other fish) are labeled by the animat vision system as reference objects and their color models are stored in the animat's mind. Once it recognizes a reference fish, the animat fixates it at all times as explained in Section 4.4, thus stabilizing its perception of the world. This allows it to perform many visual tasks accurately; for example, it can now explore its surroundings by saccading to locations of detected objects of interest in its periphery, examining them, and then return its gaze to the previous reference point. This is equivalent to having an object-centered frame of reference [Ballard and Brown, 1992] which is appropriate for dynamic vision systems such as the animat vision system.

5.1.3 Motion Segmentation and Color Recognition

For the animat to recognize objects moving in its periphery it must first detect their presence by means of a saliency map as described earlier in Section 5.1.1. Once the animat detects something that might be worth looking at, the animat must then segment its region of support out from the whole peripheral image and match this segmentation with mental models of important objects. Fig. 5.4 shows the steps involved in an incremental segmentation of the detected object over the duration of the four probe images as explained above. The accuracy of our segmentation is affected by the accuracy of the initial optical flow estimates. The robust formulation of our optical flow method ensures that the initial optical flow estimates are suitable for incremental motion segmentation.

The second-order projective parametric model that describes the motion of a planar surface with respect to the camera can be given as [Adiv, 1985, Horn, 1986]

$$\begin{aligned} u_p(x,y) &= a + bx + cy + gx^2 + hxy, \\ v_p(x,y) &= d + ex + fy + gxy + hy^2. \end{aligned} \tag{5.13}$$

In a relatively small image region, the variations in the optical flow vectors are also relatively small due to the intensity and spatial coherence constraints imposed, and the contributions from the second-order terms in equation (5.13) are usually negligible. Therefore, one may safely ignore these terms and use an affine six-parameter (a, b, c, d, e, f) motion model which is sufficient to completely specify the flow vector at every point within this

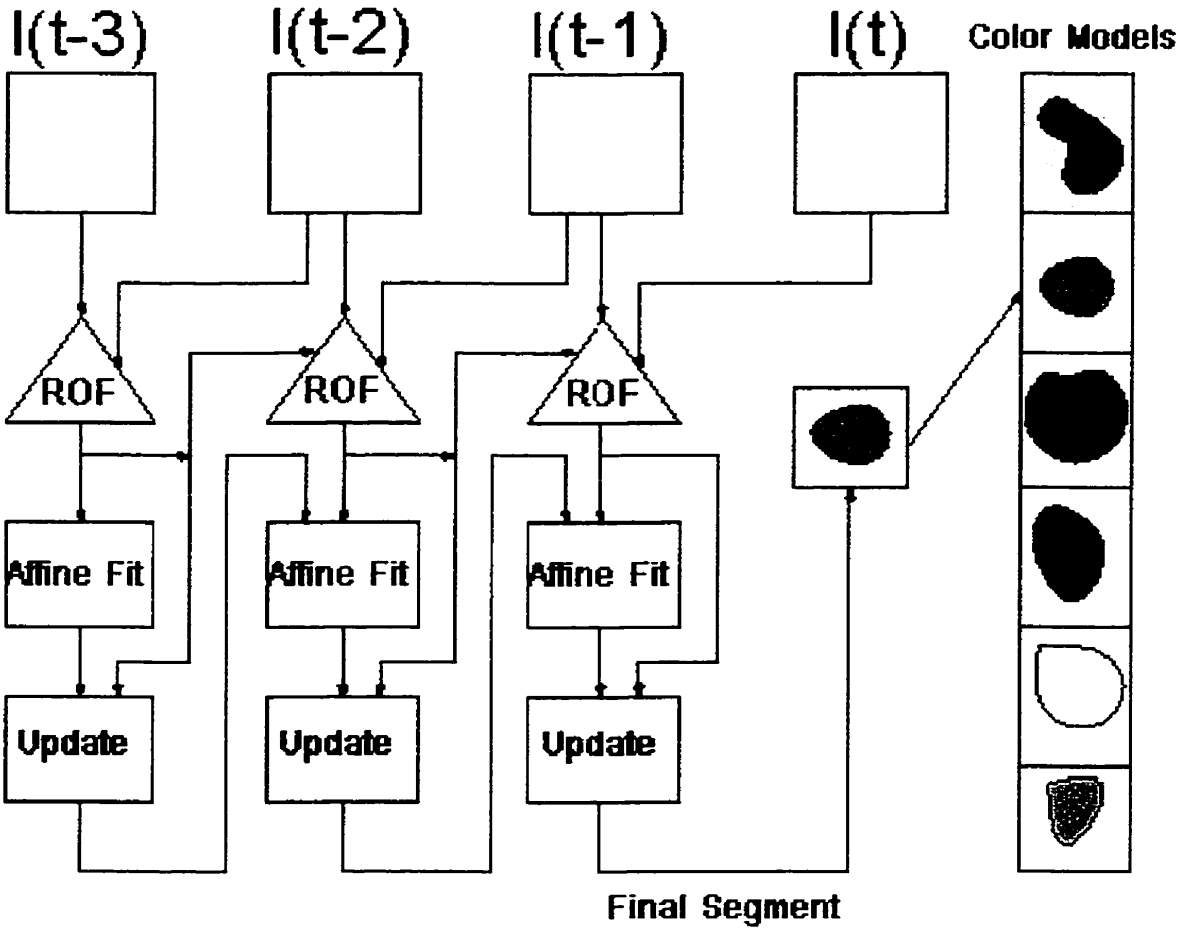


Figure 5.4: Incremental motion segmentation and object recognition using multi-resolution robust optical flow (ROF) estimation, affine parametric motion segmentation and color object recognition.

region. This approximation is justified since the second-order coefficients (g, h) become sensitive to image noise and cannot be estimated accurately in small regions typical of our multi-level 64×64 peripheral images.

Segmentation of the optical flow at each time instant is performed by fitting an affine parametric motion model to the robust optical flow (ROF) estimated so far at the current time instant. This is done by incrementally minimizing the cost function given as

$$E(a, b, c, d, e, f) = E_x(a, b, c) + E_y(d, e, f), \tag{5.14}$$

where (a, b, c, d, e, f) are the affine motion parameters. E_x and E_y are formulated using

robust estimation to account for outliers

$$\begin{aligned} E_x &= \sum_{x,y \in R} \rho(u_a(x,y) - u(x,y), \sigma), \\ E_y &= \sum_{x,y \in R} \rho(v_a(x,y) - v(x,y), \sigma), \end{aligned} \tag{5.15}$$

where R is the current region of support of the segmented object (initially equal to the full frame image size). The quantities u_a and v_a are the horizontal and vertical components of the affine flow vectors according to equation (5.2). The ROF estimated at the current instant is (u, v), and $\rho(x, \sigma)$ is taken to be the Lorentzian robust estimator. We use successive over relaxation and GNC to minimize this cost function, as described in Section 5.1.2, by using a small number of iterations over a sequence of four image probes and updating the segmentation at every time instant. The iterative equations for minimizing E_x are

$$\begin{aligned} a^{i+1} &= a^i - \frac{\mu}{T_a} \frac{\partial E_x}{\partial a}, \\ b^{i+1} &= b^i - \frac{\mu}{T_b} \frac{\partial E_x}{\partial b}, \\ c^{i+1} &= c^i - \frac{\mu}{T_c} \frac{\partial E_x}{\partial c}, \end{aligned} \tag{5.16}$$

and similarly for E_y. The T_i terms are given as

$$\begin{aligned} T_a &= \sum_{x,y \in R} \frac{1}{\sigma^2} \\ T_b &= \sum_{x,y \in R} \frac{x^2}{\sigma^2} \\ T_c &= \sum_{x,y \in R} \frac{y^2}{\sigma^2}. \end{aligned} \tag{5.17}$$

The partial derivatives in (5.16) are

$$\begin{aligned} \frac{\partial E_x}{\partial a} &= \sum_{x,y \in R} \psi(u_a - u, \sigma), \\ \frac{\partial E_x}{\partial b} &= \sum_{x,y \in R} x\psi(u_a - u, \sigma), \\ \frac{\partial E_x}{\partial c} &= \sum_{x,y \in R} y\psi(u_a - u, \sigma), \end{aligned} \tag{5.18}$$

and similarly for partial derivatives of E_y.

The estimated affine motion parameters at the current time instant are then used to update the segmentation by calculating an error norm S between the affine flow estimate (u_a, v_a) and the ROF estimate as in (5.1). This norm is then thresholded by an appropriate threshold $\tau_{\min}$ taken to be the minimum outlier in the affine fit. Values of $S < \tau_{\min}$ are considered to belong to the object being segmented, while values of $S > \tau_{\min}$ are discarded as outliers. The updated segmentation serves as the region of support R for the next frame's affine minimization step.

If more than one moving object is present in the probe sequence, the current segmentation is subtracted from the image, and another affine motion model is fitted to the remaining pixels thus segmenting other moving objects. To clean up the segmentation (in case some pixels where misclassified as outliers) a 9×9 median filter is passed over the segmentation mask to fill in missing pixels and remove misclassified outliers. Fig. 5.5 shows the segmented background (showing two objects as outliers) and the segmentation of the outlier pixels into the object of interest (a blue fish).

At the end of the motion segmentation stage, the segmented objects are matched to color models using the color histogram intersection method of Section 4.3. If a match occurs, the current estimate of the ROF is set to zero, thus accounting for the dynamic changes in the system, otherwise the ROF is used to initialize the optical flow at the next time step as shown in Fig. 5.2.

If the model object matches the peripheral segmented region, the animat localizes the recognized object using color histogram backprojection and foveates it to obtain a high-resolution view. It then engages in appropriate behavioral responses.

5.1.4 Behavioral Response to a Recognized Target

The behavioral center of the brain of the artificial animal assumes control after an object is recognized and fixated. If the object is classified as food, the behavioral response would be to pursue the target in the fovea with maximum speed until the animat is close

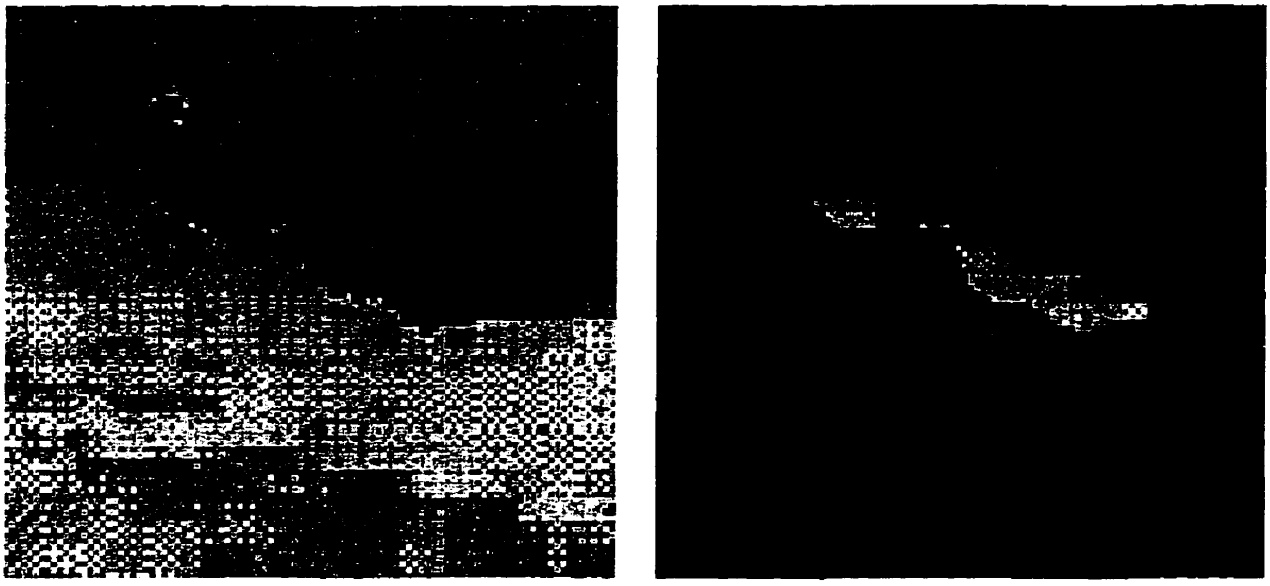
enough to open its mouth and eat the food. If the object is classified as a predator and the animat is a prey fish, then the behavioral response would be to turn in a direction opposite to that of the predator and swim at maximum speed. Alternatively, an object in the scene may serve as a visual frame of reference. When the animat recognizes a reference object (which may be another fish) in its visual periphery, it will fixate on it and track it in smooth pursuit at an intermediate speed. Thus, the fixation point acts as the origin of an object-centered reference frame allowing the animat to stabilize its visual world and explore its surroundings.

Fig. 5.6 shows a sequence of retinal images taken from the animat's left eye. The eyes are initially fixated on a red reference fish and thus the images are stabilized. In frame 283 to 286 a blue fish swims close by the animat's right side. The animat recognizes this as a reference fish and thus saccades the eyes to foveate the fish. It tracks the fish around, thereby exploring its environment. By foveating different reference objects, the animat can explore different parts of its world. The lines emanating from the animat's eyes show the line of sight and give an example of how the ground truth data available from the graphics pipeline can be useful in evaluating the accuracy of the vision algorithms such as the estimated depth to the target. The white portion of the lines represent our algorithm's estimation of the depth to the fixated target. The yellow portion to the target gives the error in this estimation. This would have been very difficult with physical active vision implementations due to the lack of ground truth data that is available on the fly.

Fig. 5.7 shows a plot of the (θ_L, θ_R) gaze angles and the turn angle between frames 200 and 400. It is clear from the figure that the animat was first fixated on the red fish which was to the left of the animat (negative gaze angles). At frame 286 and subsequent frames, the animat is foveated on the blue fish which is to its right (positive gaze angles).

5.2 Summary

This chapter presented gaze control algorithms for active perception in mobile autonomous agents with directable, foveated vision sensors. The active perception systems of the



Segmented Background Segmented Object

Figure 5.5: Results of incremental motion segmentation module.

animats continuously analyze photorealistic retinal image streams to glean information useful for controlling their eyes and body. The vision system computes optical flow and segments moving targets in the low-resolution visual periphery. It then matches segmented targets against mental models of colored objects of interest. The eyes saccade to increase acuity by foveating objects. The resulting sensorimotor control loop can support complex behaviors, such as predation.

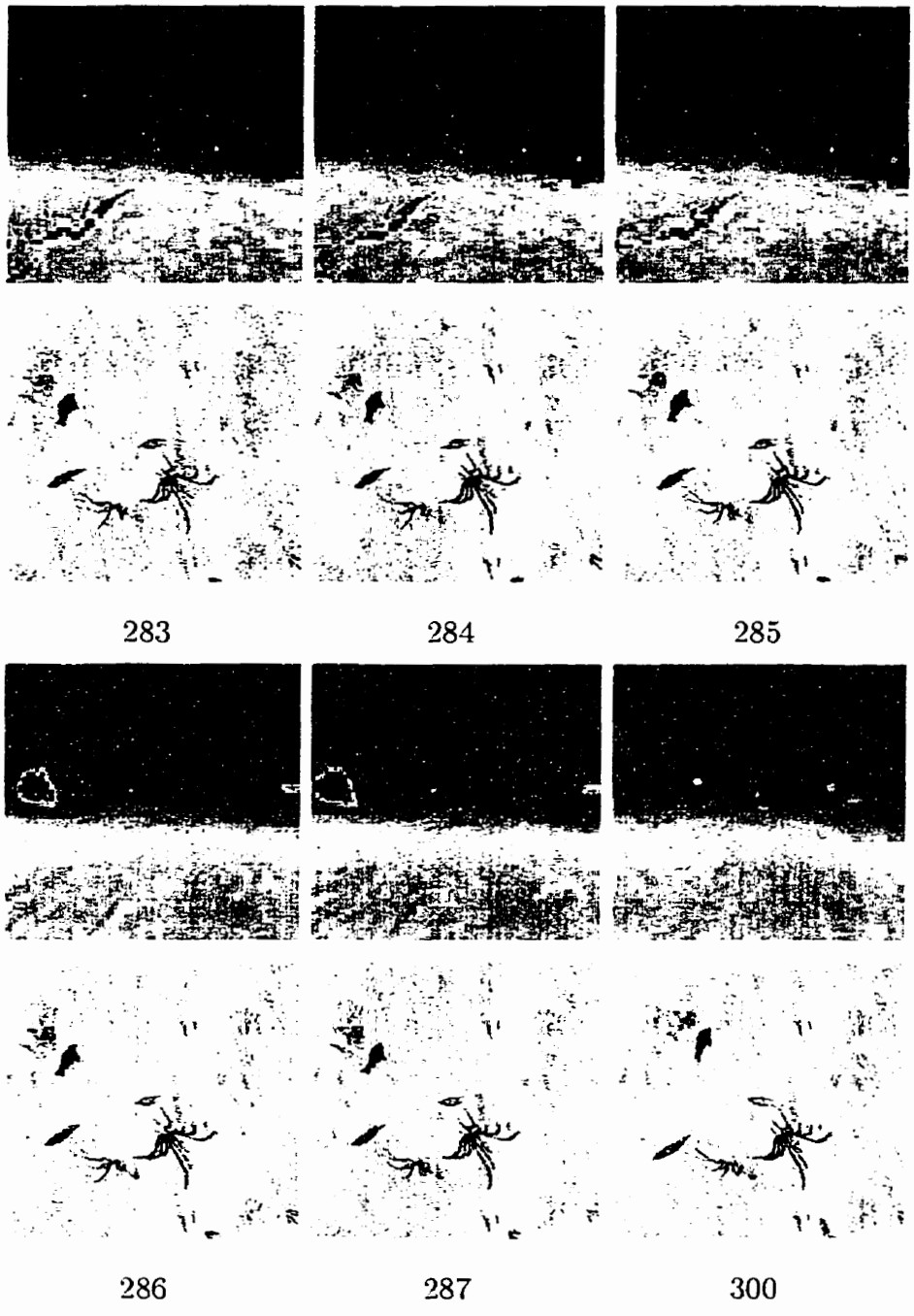


Figure 5.6: Retinal image sequence from the predator's left eye (1st and 3rd rows) and overhead view (2nd and 4th rows) of the predator as it pursues a red reference fish (frames 283–285). A blue reference fish appears in the predator's right periphery and is recognized, fixated and tracked (frames 286–300). The white lines indicate the gaze direction.

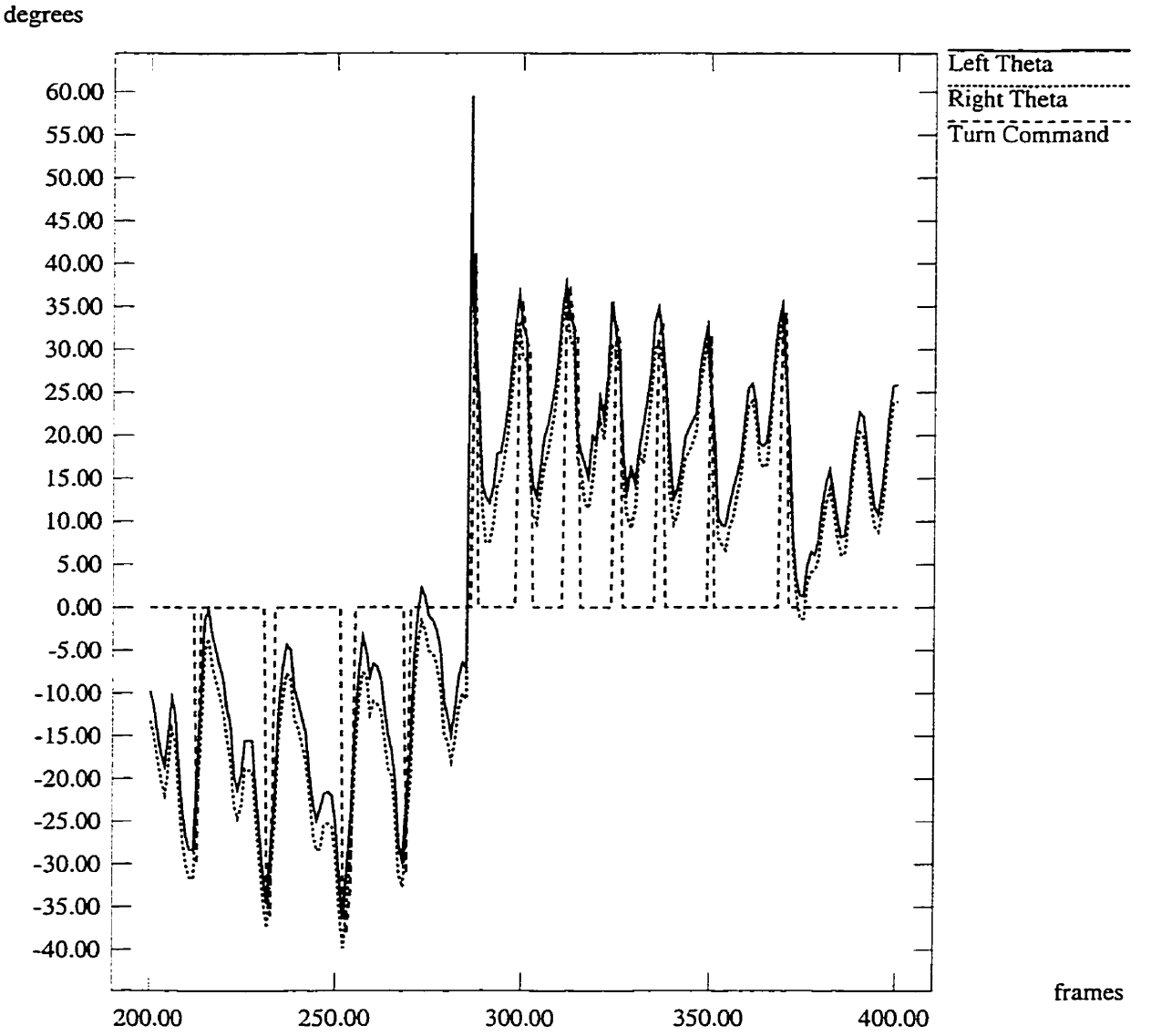


Figure 5.7: Gaze angles as the animat changes reference points at frame 286 from left (negative angles) to right (positive angles).

Chapter 6

Stereo and Color Analysis for Dynamic Obstacle Avoidance

Biological creatures move through the world with little apparent effort. For example, a person can safely navigate a hallway while reading a book that occupies her foveal vision while avoiding potential threats identified through peripheral vision. In fact, a great deal of mobility can be supported by low resolution peripheral vision, freeing the small, high resolution visual area to attend to important matters during navigation [Coombs and Roberts, 1992]. As mentioned earlier, spatially nonuniform retinal imaging provides opportunities for increased computational efficiency through economization of photoreceptors and focus of attention, but it forces the visual system to solve problems that do not generally arise with a uniform field of view. A key problem is determining how to avoid danger when obstacles are detected in the low resolution periphery while focusing attention on an object of interest fixated in the high resolution fovea. In this chapter we present a solution to this problem through the combined exploitation of color information and depth information from stereo disparity.

Our new navigation algorithms further enhance the prototype animat vision system implemented in Chapter 4. They support in the artificial fishes more robust vision-guided navigation, including obstacle recognition and avoidance. In the next section we present our work on integrating stereo disparity and color analysis for animat navigation and perception.

6.1 Disparity and Color for Obstacle Avoidance

Color and stereo algorithms have been discussed extensively in the literature in a variety of passive vision systems, but rarely have they been integrated for use in dynamic obstacle avoidance systems. Color and stereo cues have recently been integrated together with motion cues to implement a real-time passive stereo system that can detect and identify moving objects for application to surveillance and human-computer interaction [Arakawa and Etoh, 1995]. Disparity and color cues have also been combined to improve the focus of attention and recognition capabilities of an active vision system [Grimson *et al.*, 1994].

Recent work involving autonomous mobile robot systems have used single image cues for obstacle detection and avoidance such as stereo disparity [Cho and Cho, 1994], optical flow [Campani *et al.*, 1995], visual looming [Joarder and Raviv, 1994], peripheral optical flow [Coombs and Roberts, 1992], divergence of image flow and time-to-contact [Coombs *et al.*, 1995], and appearance based models of color and shape [Salgian and Ballard, 1998]. Shigang *et. al.* [Shigang *et al.*, 1995] have recently proposed a method for autonomous robot navigation along routes described by landmarks. These landmarks are selected from a set of objects with distinct features segmented out from a continuous scan of the environment based on range and color information. Color cues have also been used to increase the precision of stereo matching [Nguyen and Cohen, 1992].

The following sections describe our dynamic obstacle recognition and avoidance algorithms. Exploiting stereo and color cues, the algorithms enable the animat to navigate through its virtual environment fixating and tracking a reference target in the fovea while avoiding obstacles that appear in its low resolution visual periphery.

6.1.1 Stereo Analysis

Stereo analysis is a process of extracting depth information from a pair of left and right camera images viewing objects from slightly different view points by identifying corresponding points in the left and right images. The horizontal difference between the two matched points is known as the disparity. Depth can be directly recovered from a pri-

ori knowledge of the binocular camera geometry [Chen and Bovik, 1995]. The task of determining the correspondence between points in the two views is known as the correspondence problem and is considered difficult. In general it is a two dimensional search through the entire image space [Jenkin and Tsotsos, 1994]. Knowledge of the camera geometry can be used to limit the search to be one dimensional along the epipolar line, which is the intersection of the left and right image planes with the epipolar plane (the plane through a point in the scene and the nodal points of the two cameras) [Horn, 1986].

Classical approaches to stereo analysis try to deal with the correspondence problem with two basic algorithms; area-based [Lucas and Kanade, 1981, Horn, 1986] and feature-based approaches [Marr and Poggio, 1979, Horn, 1986]. Both types of stereo algorithms have computational problems. For example, in feature-based stereo algorithms the intensity data is first converted to a set of features assumed to be a more stable image property than raw intensities. The matching stage operates only on these extracted image features, consequently, producing sparse disparity maps. In order to obtain dense disparity maps, one is forced to interpolate these missing values. Furthermore, false matches are basic to all feature-based stereo algorithms. These problems can be reduced by introducing additional constraints derived from reasonable assumptions about the physical properties of object surfaces and by increasing the number of features considered in the matching process. In area-based stereo algorithms intensity values within small image patches of the left and right views are compared and the correlation between these patches is attempted to be maximized. To assure stable performance, area-based stereo algorithms need suitably chosen correlation measures and a sufficiently large patch size which is a computationally expensive process. Other methods extract local Fourier phases of left and right images and the phase difference at each location is used to estimate disparity [Sanger, 1988, Langley *et al.*, 1990, Fleet *et al.*, 1991].

Several approaches take into consideration available biological and neurophysiological data about the human visual system [Marr and Poggio, 1979, Sanger, 1988, Jones and

Malik, 1992]. There is biological evidence that the pattern of light projected on the human retina is sampled and spatially filtered. Very early in the cortical visual processing, receptive fields become oriented and are well approximated by linear spatial filters, with impulse response functions that are similar to partial derivatives of a Gaussian function [Young, 1986].

6.1.2 Disparity Estimation for Animat Vision

Our animat vision approach for estimating stereo disparity draws ideas from the early visual processing in the primate cortex. We implement the receptive fields as steerable spatial filters that process the input images. The steerable filter responses at an image location form a *feature vector* that is used for solving the correspondence problem. The outputs of a steerable filter convolved with an image at multiple orientations provides very rich information about a local neighborhood around each pixel. Thus matching image patches from the left and right images of a stereo pair becomes simpler and the probability of a correct match increases as the length of the feature vector increases.

Oriented filters are important for many computer vision and image processing tasks, such as texture analysis, image enhancement, and motion analysis. One approach to finding the response of a filter at many orientations is to apply many versions of the same filter each differing from one another by a small rotation in angle. A more efficient approach is to apply a few filters corresponding to a few angles and interpolate between the responses. With the correct filter set and the correct interpolation rule, it is possible to determine the response of a filter of arbitrary orientation without explicitly applying that filter.

"Steerable filter" is a term used to describe a class of spatial filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters. Steerable filters, first developed by Freeman and Adelson [Freeman and Adelson, 1991], have been recently used for estimation of scene motion [Huang and Chen, 1995] and for object recognition [Ballard and Wixson, 1993] and stereopsis [Jones and Malik, 1992].

As an example, consider the two-dimensional circularly symmetric Gaussian function

$$G(x,y) = e^{-(x^2+y^2)}. \tag{6.1}$$

The first x derivative of this Gaussian is

$$G_1^{0^\circ} = \frac{\partial}{\partial x} G(x,y) = -2xe^{-(x^2+y^2)}, \tag{6.2}$$

and the same function rotated 90° is

$$G_1^{90^\circ} = \frac{\partial}{\partial y} G(x,y) = -2ye^{-(x^2+y^2)}. \tag{6.3}$$

Thus, the derivative in an arbitrary direction θ can be synthesized by taking a linear combination of the basis filters $G_1^{0^\circ}$ and $G_1^{90^\circ}$ as follows:

$$G_1^{\theta} = \cos(\theta) G_1^{0^\circ} + \sin(\theta) G_1^{90^\circ}. \tag{6.4}$$

The $\cos(\theta)$ and $\sin(\theta)$ terms are the corresponding interpolation functions for those basis filters. Since convolution is a linear operation, it is possible to synthesize an image filtered at an arbitrary orientation by taking linear combinations of the images filtered with $G_1^{0^\circ}$ and $G_1^{90^\circ}$:

$$G_1^{\theta} * I(x,y) = \cos(\theta) G_1^{0^\circ} * I(x,y) + \sin(\theta) G_1^{90^\circ} * I(x,y). \tag{6.5}$$

This gives an illustration of steerability, which is a very useful property, since the response of a steerable filter at an arbitrary orientation can be obtained from a small number of precomputed basis responses using the corresponding interpolation functions. Simoncelli and Freeman have recently introduced a multi-scale, multi-orientation steerable filter image decomposition framework called the Steerable Pyramid [Simoncelli and Freeman, 1995] which we use as a front-end for our stereo algorithm. It has the advantage of producing feature descriptions that are both translation- and rotation-invariant.

Our disparity estimation algorithm starts by decomposing the left and right images into steerable pyramid representations. The input images are initially low-pass filtered using a low-pass filter (L_0) with a radially symmetric frequency response. Each successive

level of the pyramid is constructed from the previous level's low-pass band by subsampling it then convolving it with a bank of oriented basis filters (B_i) and a low-pass filter (L_1) (refer to figure 6.1). Other orientations at each level are synthesized by taking linear combinations of the basis filtered images. The number of basis filters that are needed for steering the filter is $n + 1$ for an n_{th}−order filter. We use third-order filters, thus requiring four basis filters oriented at $0°, 45°, 90°$, and $135°$ [Freeman and Adelson, 1991]. Fig. 6.2(a) shows these four spatial basis filters (B_i) which form a steerable basis set; any orientation of this filter can be written as a linear combination of the basis filters. Fig. 6.2(b) shows the two low-pass filters used to construct the pyramid. Typically, $L_0(w)$ is chosen to be $L_1(w/2)$ in the frequency domain so that the initial low-pass shape is the same as that used within the recursion due to subsampling [Simoncelli and Freeman, 1995]. Fig. 6.3 shows an example of a three-level steerable pyramid of an image acquired by the animat's right eye for two orientations.

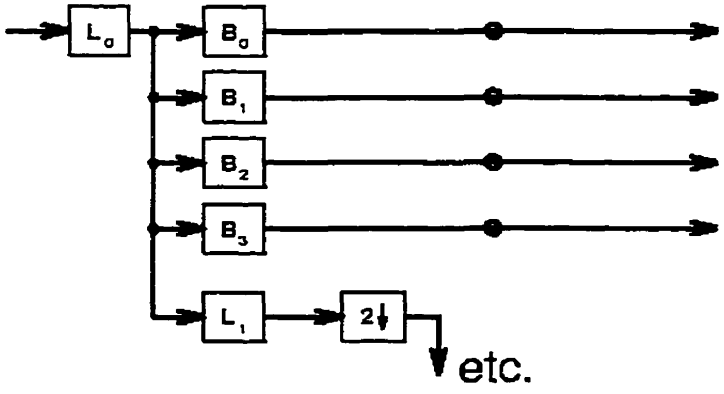
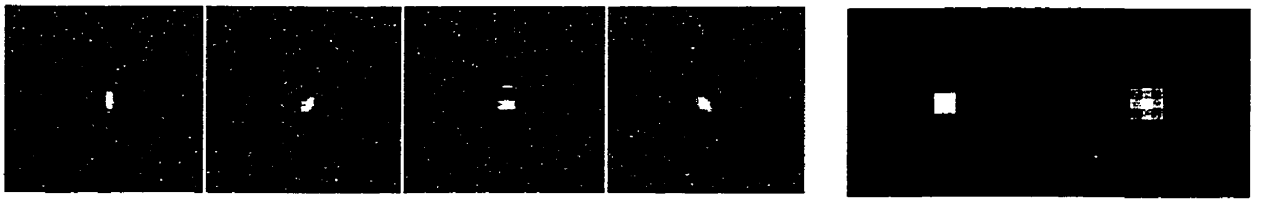


Figure 6.1: System diagram for the first level of the steerable pyramid. Boxes represent filtering and subsampling operations: L_0 and L_1 are low-pass filters, and B_i are oriented basis filters. Circles in the middle represent the basis filter responses. Successive levels of the pyramid are computed by applying the B_i and L_1 filtering and subsampling operations recursively (represented by "etc." at the bottom).

Feature vectors $\mathbf{f}_R(x, y, l)$ and $\mathbf{f}_L(x, y, l)$ are then constructed from the right and left pyramid responses for each pixel at each level of the pyramid by combining the responses



(a) (b)

Figure 6.2: (a) Four basis filters oriented at $0°, 45°, 90°$, and $135°$. (b) Left: L_1 low-pass filter, Right: L_0 low-pass filter.

of the multi-orientation steerable filters at each pixel into a vector that provides a very rich description of the intensities at that pixel in the image. To further enrich the description of each pixel, we make use of the (R, G, B) color signals from our color images by including them in the feature vector. This simple addition improves our matching process considerably by restricting the matching process to areas of similar color composition, which can be considered as a sort of color-feature constraint.

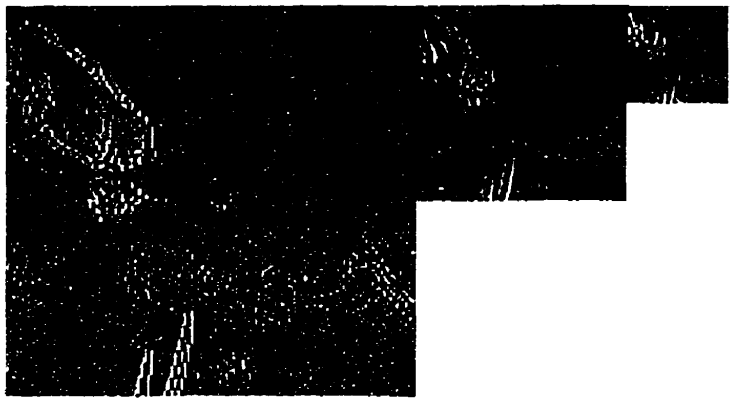
An initial disparity map is estimated at each individual level by matching left and right feature vectors by minimizing the mean square error (MSE) between left and right feature vectors. The MSE measure is computed over all the elements in the vector as follows:

$$E_m = \frac{1}{S} \sum_{i \in S} [\mathbf{f}_R^i(x, y, l) - \mathbf{f}_L^i(x + d_x, y + d_y, l)]^2, \tag{6.6}$$

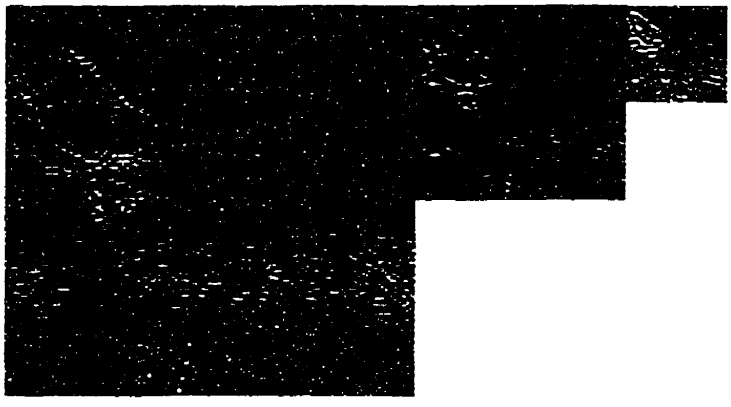
where S is the feature vector size. The MSE measure E_m is computed for a limited range of horizontal and vertical disparities $d_x(l) \in D_x(l)$ and $d_y(l) \in D_y(l)$ within a window of size $D_x(l) \times D_y(l)$ (typically, $D_x(0) = 20$, and $D_y(0) = 10$). The $(d_x(l), d_y(l))$ value that minimizes the MSE within this window is taken as the best initial disparity estimate at pixel (x, y, l) at pyramid level l. A boundary condition of zero disparity at image borders is applied. Also a zero disparity condition is applied to locations where no match is possible such as across constant intensity areas. The disparity range used lies within $[-\frac{D(l)}{2}, \frac{D(l)}{2}]$. The disparity range differs from level to level and is given as,

$$\begin{aligned} D_x(l) &= \frac{D_x(0)}{2^l}, \\ D_y(l) &= \frac{D_y(0)}{2^l}. \end{aligned} \tag{6.7}$$

(a)



(b)



(c)

Figure 6.3: An image acquired by the animat's right eye (a), and a three-level Steerable Pyramid of the image in (a) shown for two orientations 0° (b) and 90° (c).

Note that there is no need to equally weigh the dimension of the (R, G, B) color signals to correspond to the rest of the feature vector element dimension (i.e. the dimensions of the steerable filter responses) since in equation (6.6), the MSE compares corresponding elements of the same dimension together. Thus, for example, the element that corresponds to red in the right feature vector $\mathbf{f}_R^{red}$ is compared to the red element in the left feature vector $\mathbf{f}_L^{red}$. Similarly, the steerable filter elements from the right vector are compared to corresponding elements from the left. Therefore, the dimensions of the corresponding elements of the feature vector must be the same for proper matching.

A coarse-to-fine-flow-through strategy is then taken based on the assumption that for level l disparity estimates $|d(l)| > |\frac{D(l)}{4}|$ are more accurately estimated at the coarser level $l+1$. Thus at coarse levels, large disparities are estimated presumably more accurately, and these flow through to the finer levels, while small disparities that are estimated from the finer levels are assumed accurate since they cannot be estimated at coarser levels due to the loss of high frequency structure from the original coarse-level images.

Each disparity estimate $(d_x(l), d_y(l))$ at each level is median filtered at an appropriate scale (window size used increases from coarse levels to fine levels – mainly 3x3 and 5x5 for 128x128 images) before flow-through is performed. The full frame level is then, median filtered to give the final disparity estimate. The median filtering step is required to correct for outlier disparity estimates that deviate from the correct expected estimate (a form of smoothness constraint on the estimates).

The stereo matching algorithm can be made more efficient by exploiting the epipolar geometry of the eyes of the artificial animal. The eye virtual cameras described in Section 4.1 have identical focal lengths f_c. The eyes mounted in the animat's head may be aligned horizontally to within a scan line. To simplify the matching process we try to reduce the vertical disparity search range D_y as much as possible by restricting epipolar lines to one row. This is done by tying the vertical gaze angles together when acquiring stereo images; i.e., setting $\phi_R = \phi_L$. The vertical disparity search range D_y is nevertheless larger than one pixel due to non-uniform perspective distortions associated with the

large field of view virtual cameras.

Fig. 6.4 shows disparity maps estimated by the algorithm for real images, a random-dot stereogram, and retinal images acquired by our animat.

6.1.3 Color Obstacle Recognition and Localization

Next we develop a strategy to distinguish between dangerous obstacles and benign objects in the environment by combining the disparity cues estimated using the above algorithm with color cues available naturally from the acquired photorealistic images.

The animat continuously computes a disparity map from its stereo retinal input as it navigates through the virtual world. The estimated disparity map is used as a bottom-up cue to alert the animat of potential danger from objects that come too close. The disparity map is first segmented into potential obstacles. Then each segmented object is examined and matched against color mental models of designated dangerous obstacle objects. A match indicates that a candidate object is really an obstacle and is to be avoided, otherwise the object is considered harmless. Harmless objects include food particles and sea weeds.

The disparity map $d(x, y)$ is segmented via thresholding. The appropriate disparity threshold is taken to be proportional to the disparity at the fixation point, which is the reference target that the animat is tracking foveally. This is given as, $dt = \mu d(0, 0)$; i.e., if the disparity of the localized object is at least μ times the estimated disparity of the reference target then this object is considered too close (typically $\mu = 1.0$). Values of $d(x, y) > dt$ are considered to belong to potential obstacles, while values of $d(x, y) < dt$ are set to the minimum disparity estimate. This segmentation step focuses the attention of the animat on potential obstacles while disregarding the rest of the peripheral view, thus simplifying the system as well as improving its robustness to false alarms.

The corresponding segmented pixels in the right eye image give the actual segmentation of the color objects. The color histogram of this segmentation is intersected with the color histogram of the mental models of stored obstacles, using the color methods

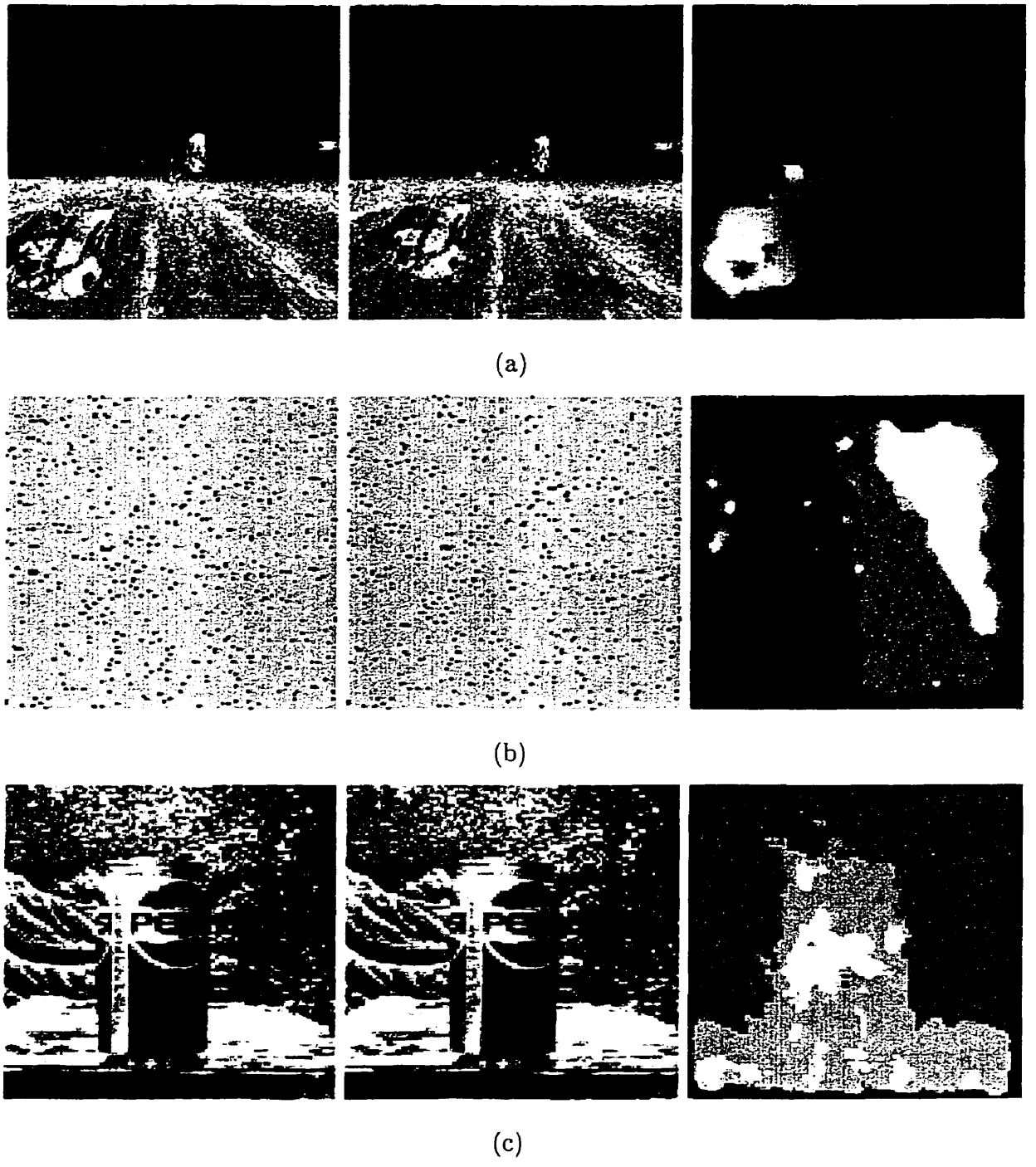


Figure 6.4: (a) Right and Left images acquired by the animat's eye and the estimated disparity map, (b) Stereo sparse random-dot-stereogram with 3% black dots and estimated disparity, (c) Pepsi sequence, left: frame 3, center: frame 0, right: estimated disparity.

described in section 4.3. A match indicates that this segment contains an obstacle; no match indicates a false alarm and the animat continues in its current path.

To accurately localize a detected obstacle the exact region of support of this obstacle must be properly segmented out from the original segmentation obtained above. To tackle this non-trivial problem, we make use of Swain's color histogram backprojection methods [Swain and Ballard, 1991]. Briefly, histogram backprojection gives large weights to pixel locations in the image whose color histogram closely resembles the color histogram of the model. This suggests, that we can use the backprojection itself to get an accurate segmentation of the detected obstacle; pixel weights in the backprojection that are greater than an appropriate threshold are considered to belong to the obstacle. The threshold is determined empirically and for our case we used a value of 0.5 to separate the obstacle from outliers. Once the color region of support of the obstacle has been determined, the corresponding region in the disparity map gives the estimated disparities of the obstacle over the region. The updated disparity map is convolved with a circular disc of area equal to the area of the segmented obstacle's region of support. This will blur out any misclassified pixels in the segmentation while emphasizing the obstacle and facilitating its localization. The pixel location (x_c, y_c) of the peak in the blurred disparity map localizes the obstacle. Fig. 6.5 shows images of the various segmentation steps.

6.1.4 Obstacle Avoidance Strategy

The point of localization (x_c, y_c) obtained from the peak in the blurred disparity map is used to compute the steering angles the animat must use to steer clear of the obstacle. The angular location with respect to the right eye is given as

$$\theta = \tan^{-1}\left(\frac{x_c}{f_c}\right),$$
$$\phi = \tan^{-1}\left(\frac{y_c}{f_c}\right). \tag{6.8}$$

The turn angles given to the animat's motor controller are, thus, proportional to $(-\theta, -\phi)$; i.e., in the opposite direction, to avoid collision while still fixating on a reference target

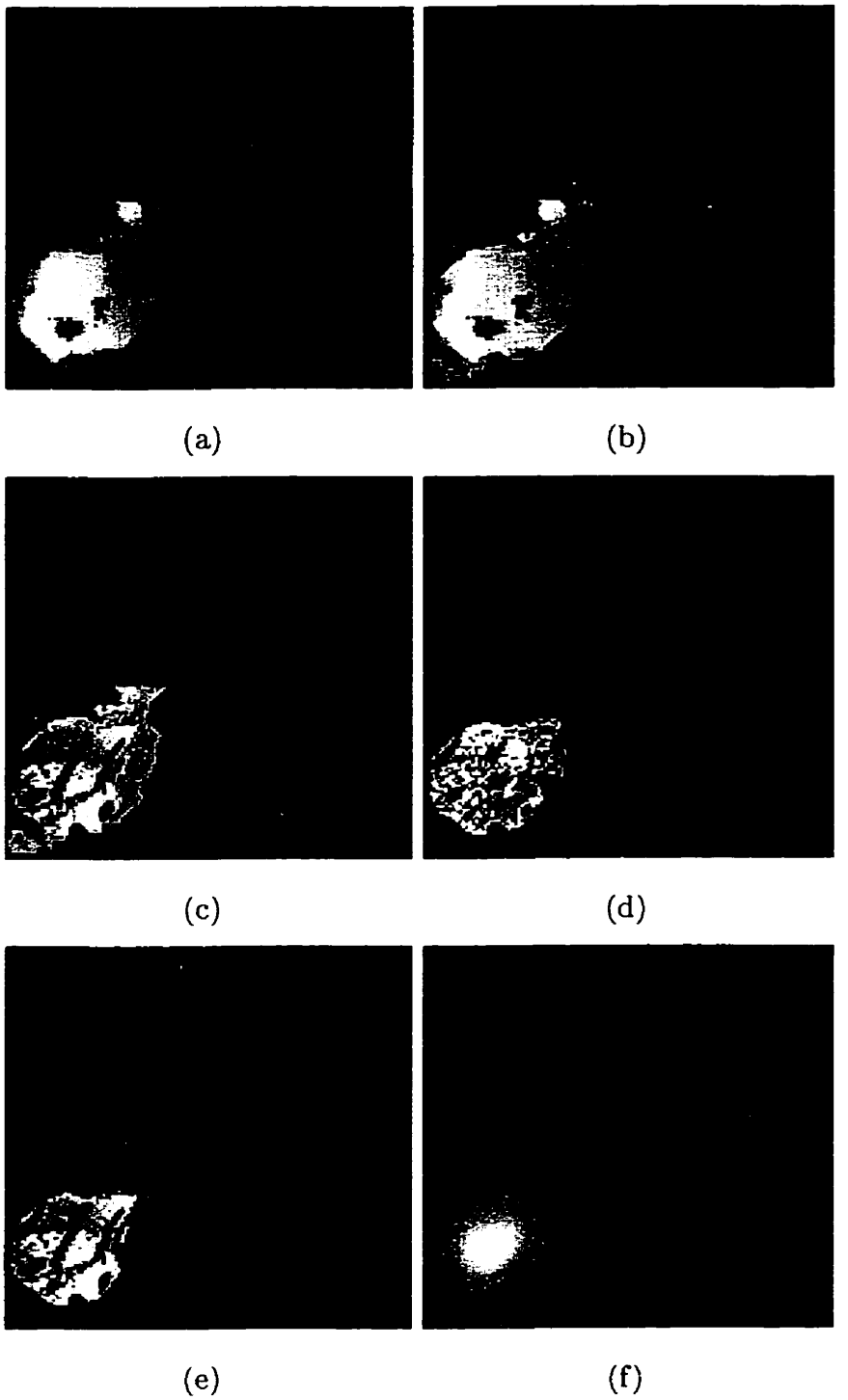


Figure 6.5: (a) Disparity map of fig. 6.4-(a), (b) Thresholded disparity map, (c) Corresponding color segmentation of potential obstacles, (d) Backprojection map, (e) The exact region of support of the segmented obstacle, (f) The localization of the obstacle by blurring the corresponding segmentation of the disparity map.

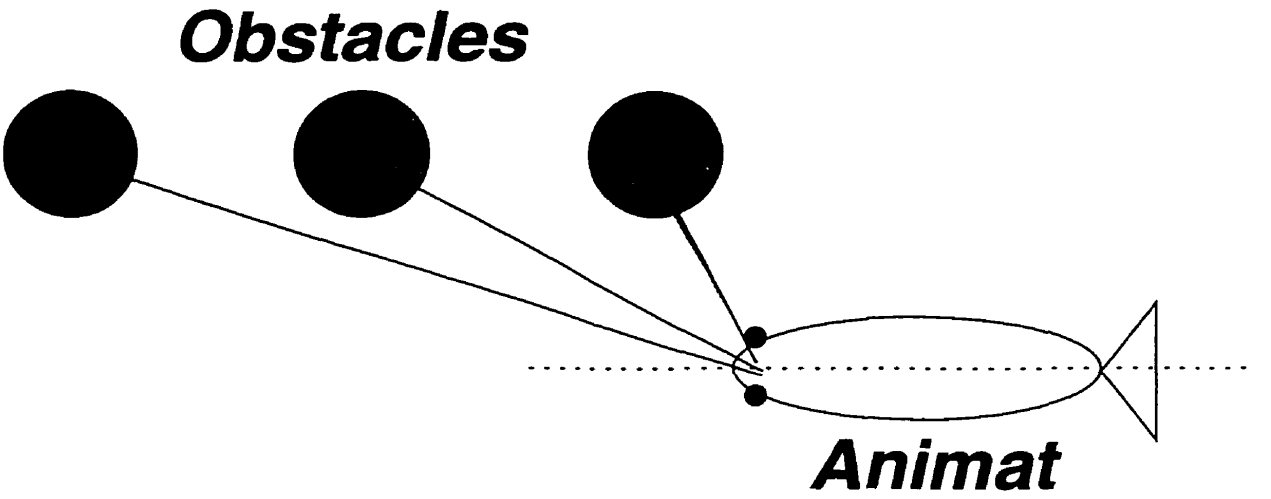


Figure 6.6: Relationship between close objects and large steering angles.

to stabilize the visual world.

The merit of using $(-\theta, -\phi)$ for steering the animat is twofold: 1) simplicity of computing a steering vector and, 2) the fact that (θ, ϕ) is large for close objects, as is depicted in Fig. 6.6. Therefore, the turn maneuver will be large to avoid the obstacle quickly. The farther away the obstacle, the smaller the turn angles, hence steering will not be excessive.

Figure 6.7 shows frames from a top view of a sequence showing the animat navigating in its environment. The animat is fixating and tracking a target red fish while avoiding obstacles taking the form of other fish obstructing its path. The figure shows three instances where the animat encounters an obstacle (frames 50, 156 and 180). These are followed by frames showing how the animat has successfully avoided the obstacle by steering its body in the opposite direction as explained above.

6.2 Summary

This chapter presented a vision system for highly mobile autonomous agents that is capable of dynamic obstacle avoidance. Through active perception, each agent controls its eyes and body by continuously analyzing photorealistic binocular retinal image streams. The vision system computes stereo disparity and segments looming targets in the low-resolution visual periphery while controlling eye movements to track an object fixated

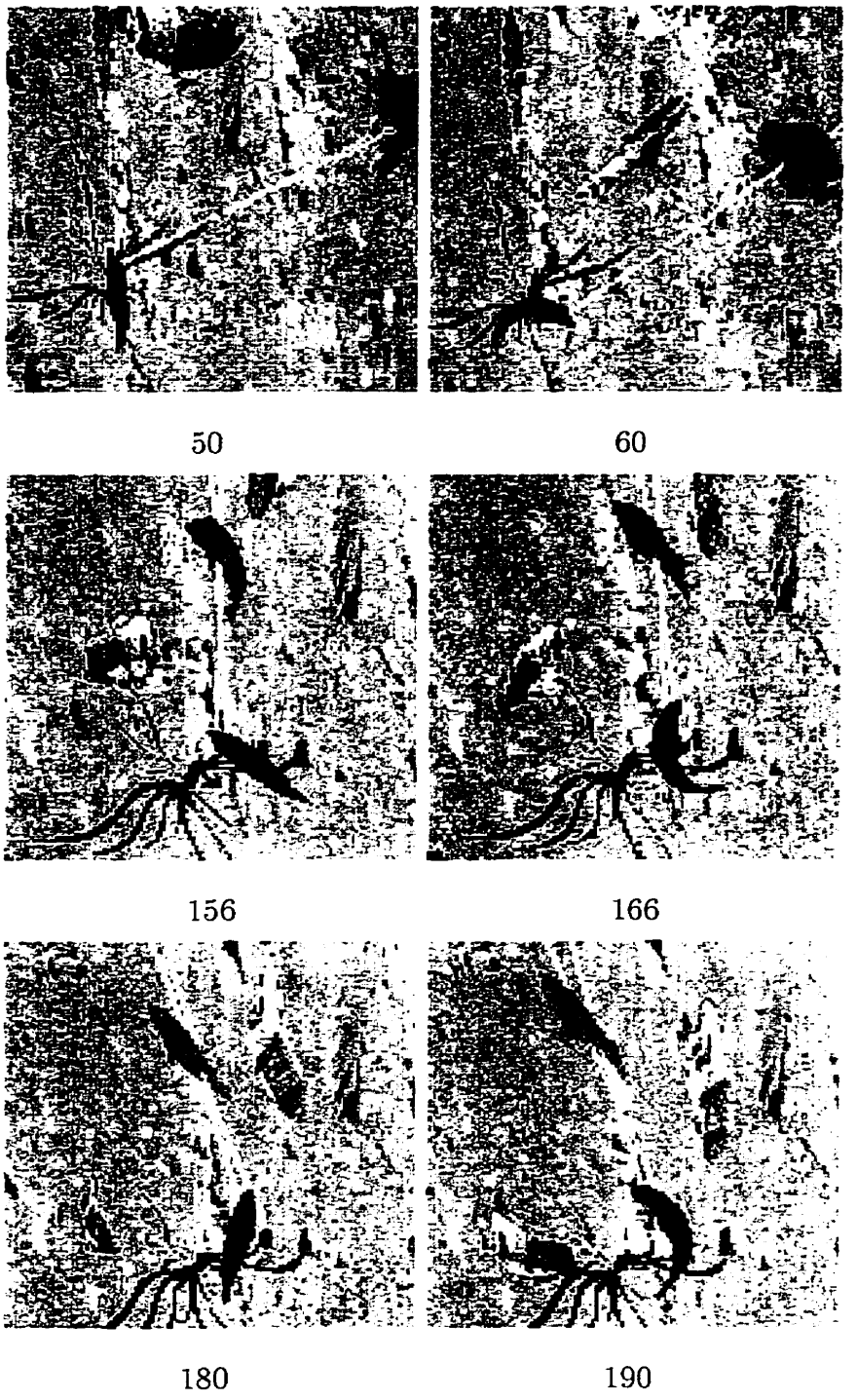


Figure 6.7: An overhead view of the animat as it pursues a red reference fish while detecting and avoiding other fish obstacles. The white lines emanating from the eyes of the observer indicate the gaze direction.

in the high-resolution fovea. It matches segmented targets against mental models of colored objects of interest in order to decide whether the segmented objects are harmless or represent dangerous obstacles. The latter are localized, enabling the artificial animal to exercise the sensorimotor control necessary to avoid collision.

Chapter 7

Animat Vision in Virtual Humans

The preceeding chapters have demonstrated that animat vision offers an alternative research strategy for developing biologically inspired active vision systems in realistic artificial animals implemented entirely in software on readily available 3D graphics workstations.

In this chapter, we demonstrate that the animat vision paradigm is flexible enough to be implanted into animats other than artificial fish. We suitably modify our prototype animat vision system and transplant it into two human animats; a human soldier model called *DI-Guy* developed by Boston Dynamics, Inc., (BDI) and the "first-person warrior" in the well-known interactive computer game *DOOM* developed by id Software, Inc. The following sections describe the implementation of animat vision systems in the DI-Guy soldier and DOOM environments. We present experimental results with these animat vision systems and demonstrate the appropriateness of such virtual environments as a framework for doing active vision research.

7.1 Human Animats

Recent advancements in physics-based human simulation have prompted us to incorporate the animat vision system into a human model. We have chosen the commercially available DI-Guy API developed by BDI, because it can depict the appearance and mimic the actions of humans with reasonable fidelity and computational cost. The ability of the DI-Guy animat to synthesize human actions, such as walking and running, forces the an-

imat vision system to contend with dynamics similar to those of real human bodies. Such dynamics are absent when wheel-driven hardware lab robots are used as platforms for active vision research. Hopefully our animat vision approach will foster the development of active vision systems that better approximate those responsible for human vision.

7.1.1 The DI-Guy Animat

DI-Guy is a software library for integrating life-like human characters into real-time simulated environments [Koechling *et al.*, 1998]. Each character moves realistically, and responds to simple motor commands, locomoting about the environment as directed. DI-Guy animates each character automatically so an animator is not needed. Even when switching from one activity to another, a DI-Guy makes seamless transitions and moves naturally like a real person. DI-Guy has a well documented API that allows users to specify characters, select uniforms and equipment, and control actions. The software comes with fully textured models at multiple levels of detail for efficient rendering (see figure 7.1-b), a motion library, and a high-performance real-time motion engine based on motion capture technology. The original DI-Guy character is a soldier portraying dismounted infantry for military simulations (Fig. 7.1-a). It synthesizes authentic military behavior based on the motions of trained soldiers. The system has fully textured multiresolution models, several uniforms (Battle Dress, Desert Camouflage, Land Warrior II, etc.), weapons (M16, AK47, M203) and a variety of auxiliary equipment (gas mask, backpack, canteen, bayonet, etc.).

The DI-Guy software includes a variety of other characters in addition to soldiers: Flight deck crew (FDC-Guy), landing signal officers, and airplane captains (Fig. 7.1-c). Civilian male and female pedestrians (PED-Guy) who stand, stroll, stride and strut, and sit around having a conversation (e.g., Fig. 7.1-d). Chem/Bio characters (CB-Guy) who wear gas masks, and display the effects of fatigue and toxic exposure (Fig. 7.1-e,f), and several athletes such as gymnasts, joggers, baseball and football players (Fig. 7.1-g,h).

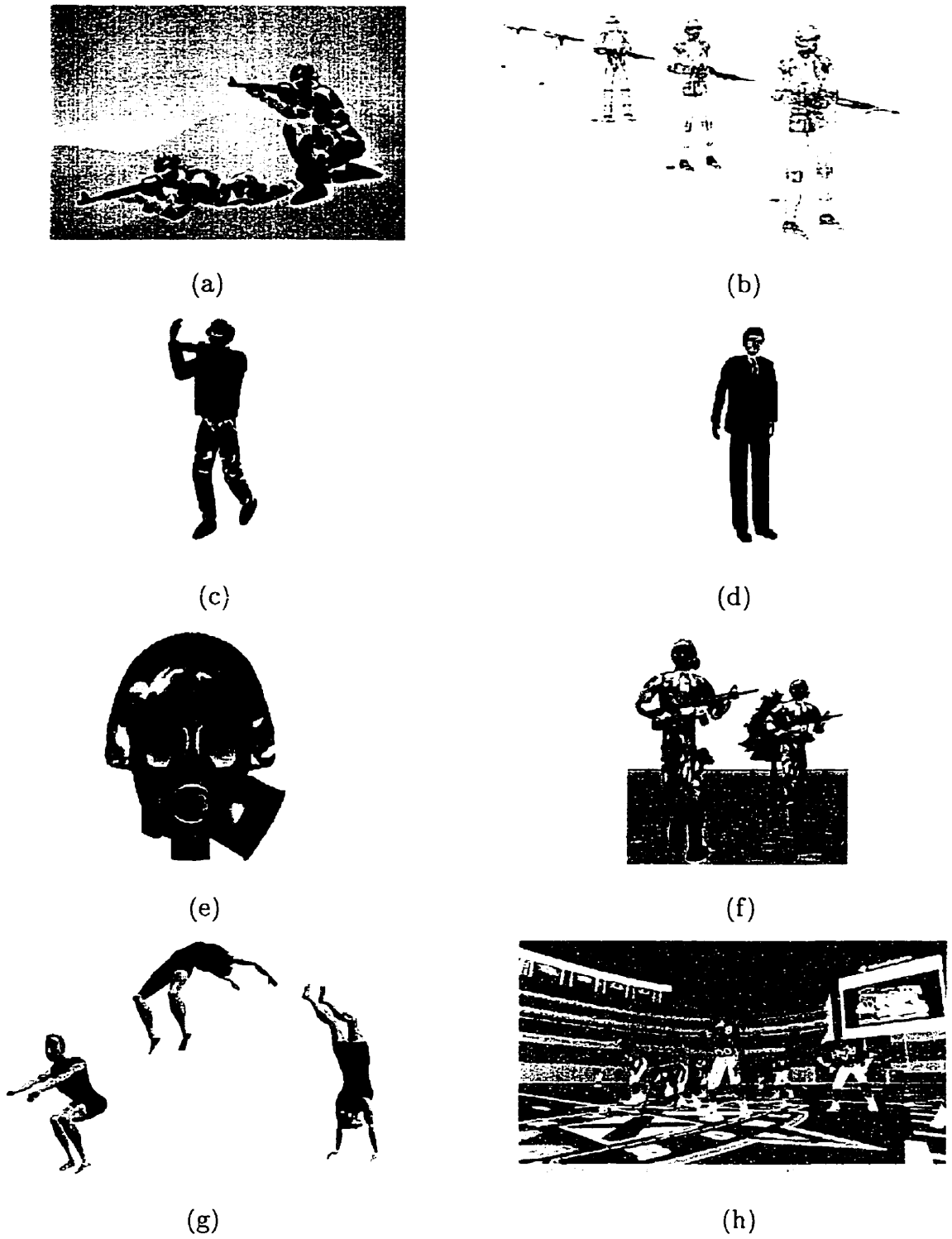


Figure 7.1: (a) Two DI-Guy soldiers in BDU military uniform. (b) DI-Guy models come in multiple levels of detail, ranging from 2500 polygons down to 38. (c) Landing signal officer. (d) A pedestrian DI-Guy character. (e) CB-Guy with gas mask equipment. (f) Two soldiers wearing gas masks. (g) DI-Guy athlete doing a back flip. (h) The Superbowl using real-time DI-Guy athletes. (Images courtesy of BDI.)

7.1.2 Programming DI-Guy

DI-Guy offers a simple programming interface [BDI, 1998]. Basic calls in the DI-Guy API are:

- `diguy_initialize()`: Initializes the environment and preloads geometry and motion data.
- `diguy_create()`: Creates a DI-Guy character with default type, location and activity.
- `diguy_set_action()`: Set desired action.
- `diguy_set_desired_speed()`: Specify speed and heading.
- `diguy_set_path()`: Specify path for character to follow.
- `diguy_destroy()`: Remove character from scene.
- `diguy_set_gaze()`: Set the $(\theta_{\text{head}}, \phi_{\text{head}})$ gaze angles for turning character's head.
- `diguy_set_orientation()`: Set steering angle θ_{steer} with respect to forward direction to steer character left and right.

DI-Guy actions include: `stand`, `walk`, `jog`, `go prone`, `walk backwards`, `kneel`, `walk crouched`, `crawl`, `aim`, `fire weapon`, and `die`. The DI-Guy coordinate system is right-handed, with the positive Y-axis pointing forward and the positive Z-axis pointing upward. A character standing at the origin with zero orientation faces in the positive Y direction, with the positive X-axis to its right, and the positive Z-axis starting on the ground between the feet and extending up through the head.

7.2 Animat Vision in DI-Guy

To incorporate the animat vision system into the DI-Guy soldier, the position of the eyes must be located on the graphics model of the character's head. An API call,

`diguy_get_full_base_position()`, that returns the exact (x, y, z) location in meters from the origin of a point on the character's pelvis was provided by the library. This point is the root of the character's graphics hierarchy. The API also returns the exact orientation angle, θ_{steer}, in degrees counter-clockwise from the positive Y direction.

Given that a character at a scale of 1.0 is about 1.83 meters in height [BDI, 1998], and knowing the offsets from the root point to different joint positions, we are able to work our way up the kinematic chain of the body to calculate the location of a point in the head centered between the two eyes. Choosing an appropriate baseline to separate the two virtual eyes, we are able to localize the left and right eyes of the character in arbitrary pose. This can be visualized from figure 7.2.

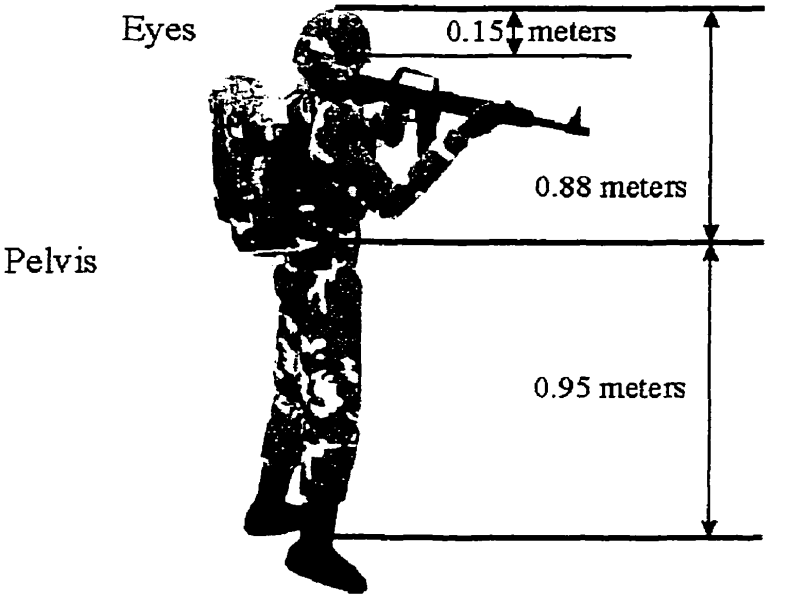


Figure 7.2: Measurements used to compute the location of the eyes for a DI-Guy character at a scale of 1.0.

7.2.1 Eyes and Retinal Imaging

Like the artificial fish, the DI-Guy virtual character has binocular vision. The movements of each eye are controlled through two gaze angles $(\theta_{eye}, \phi_{eye})$ which specify the horizontal and vertical rotation of the eyeball, respectively, independent of the movement of the

head. The angles are measured with respect to the head coordinate frame, which is itself relative to the body coordinate frame. Therefore, the eye is looking straight ahead when $\theta_{\text{eye}} = \phi_{\text{eye}} = 0°$ with respect to the head forward direction. Also the head is gazing forward when $\theta_{\text{head}} = \phi_{\text{head}} = 0°$ with respect to the forward direction of the body.

The retinal field of each eye has three levels of decreasing resolution. This approximates the spatially nonuniform, foveal/peripheral imaging capabilities typical of human eyes. The level $l = 0$ camera has the widest field of view (about 80°) and the horizontal and vertical fields of view for the level l camera are related by

$$f_x^l = 2\tan^{-1}\left(\frac{d_x/2}{2^l f_c^0}\right); \qquad f_y^l = 2\tan^{-1}\left(\frac{d_y/2}{2^l f_c^0}\right), \tag{7.1}$$

where d_x and d_y are the horizontal and vertical image dimensions and f_c^0 is the focal length of the wide field of view camera ($l = 0$). Initially f_c^0 is unknown, but the $l = 0$ field of view is known, then f_c^0 is first computed using

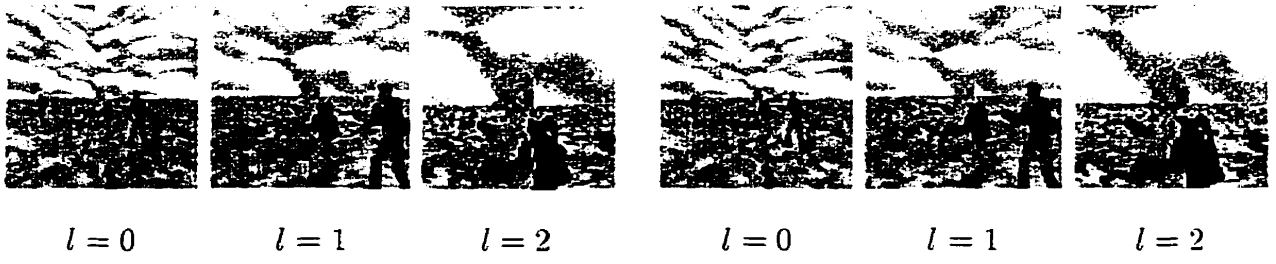
$$f_c^0 = \frac{d_x}{2\tan(\frac{f_x^0}{2})}, \tag{7.2}$$

and this value is used to determine the field of view at the other levels.

Fig. 7.3(a) shows an example of the multiscale retinal pyramid with highest resolution and smallest field of view at the fovea $l = 2$, and lowest resolution with largest field of view at the peripheral image $l = 0$. Fig. 7.3(b) shows the binocular retinal images with a black border around each magnified component image to reveal the retinal image structure in the figure.

7.2.2 Foveation and Vergence

The DI-Guy animat employs the same color histogram methods of Chapter 4. A model image used to recognize targets is shown in Fig. 7.4. When a target is detected in the visual periphery using color histogram intersection, it is localized using the color histogram backprojection method. The eyes will then saccade to the angular offset of the target's location to bring it within the fovea. The left and right eyes are then converged by computing the stereo disparities (u, v) between the left and right foveal images at the



(a)

Right eye Left eye

(b)

Figure 7.3: Binocular retinal imaging. (a) 3 component images; $l = 0, 1$ are peripheral images; $l = 2$ is foveal image. (b) Binocular retinal images (borders of component images are shown in black).

current level, using the optical flow method described in Section 4.4, and correcting the gaze angles of the left eye to bring it into registration with the right eye.

Detection and localization continues from frame to frame at the current foveal level as long as the area of the target in this level is below a specific threshold area. When the DI-Guy animat comes too close to the target it is tracking and the target area increases accordingly, the gaze control algorithm will work at the next lower level were the field of view is larger and thus the target area is smaller and contained inside the level's frame. Also, the speed of the animat is reduced when it approaches too close to the

target in order to avoid collision. When the target moves farther away from the animat as indicated by a smaller target area in the current level's image frame, the animat will increase its speed and the foveation and vergence will take place at the next higher level where the calculations are more accurate.

It is straightforward to estimate the area of the target accurately once it has been detected. This is done using our implementation of the histogram intersection method, which sizes down an initially larger model histogram to the approximate size of the target histogram as explained in Section 4.3. The area of the target is thus obtained by summing up the number of pixels in the sized down model histogram bins. This is another advantage of our robust implementation of the color histogram intersection method.

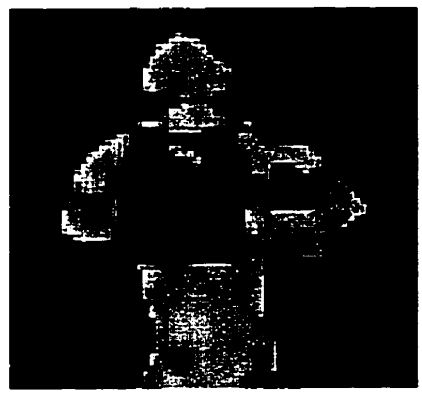


Figure 7.4: The model image of the target detected by the DI-Guy animat.

7.2.3 Vision-Guided Navigation

The DI-Guy animat has different degrees of freedom than the artificial fish animat. The soldier's body can be steered relative to the positive Y' direction using the orientation angle θ_{steer}. The soldier can also move its head relative to its body using the head gaze angles $(\theta_{\text{head}}, \phi_{\text{head}})$. It can also move its eyes relative to the head using the eye gaze angles $(\theta_{\text{eye}}, \phi_{\text{eye}})$. This added freedom necessitates a modification to the original gaze control algorithm in order to coordinate the eye-head-body motion. Initially, all angles are set to zero indicating a forward orientation with the head and eyes gazing forward. As the

animat fixates and tracks a target, the eye gaze angles are used to rotate the head such that if $(\theta_{\text{eye}}, \phi_{\text{eye}}) > \tau_{\text{head}}$ then $(\theta_{\text{head}}, \phi_{\text{head}}) = (\theta_{\text{eye}}, \phi_{\text{eye}})$. Thus, the head is turned to align with the gaze direction of the eyes. This continues until $\theta_{\text{head}} > \tau_{\text{steer}}$, at which point θ_{steer} is set to equal θ_{head}, thus steering the animat in the gaze direction. This simple control method allows the animat effectively to navigate the virtual environment in a natural way while visually tracking targets.

Figure 7.5 shows a sequence of image frames of a DI-Guy soldier animat pursuing a differently dressed soldier. The sequence is shown from frame 57 to frame 97, sampling every 10 frames. The inter-frame time step was approximately 0.13 seconds. The left column in the figure shows a top and side view of the observer in the grey uniform fixating on the soldier in the desert camouflage uniform, and successfully tracking it from frame to frame. The green lines emanating from the soldier animat's eyes indicate the lines of sight from the left and right eyes intersecting at the fixation point on the target soldier, thus clearly showing the vergence of the eyes on the target. There are seven other soldiers in land warrior camouflage uniforms training in the background. They are ignored by the observer animat even though they appear in its peripheral vision. For this sequence, we have set $\tau_{\text{head}} = 15°$, and $\tau_{\text{steer}} = 30°$.

The right column in the figure shows the corresponding stereo images acquired by the observer during navigation. It shows the target nicely fixated in the center of the left and right foveas as the animat tracks the target throughout the sequence. Fixation is achieved by foveating the eyes with compensating saccade signals.

7.3 Doom Vision

Another virtual environment with which we experimented in our animat vision research was the DOOM environment. DOOM is a first-person game from id Software, Inc., which puts the player in the perspective of a battle-hardened marine fighting for survival against unyielding demons. DOOM is a visually simpler three dimensional graphics environment compared to DI-Guy. Enemy agents are rendered as two dimensional sprites.

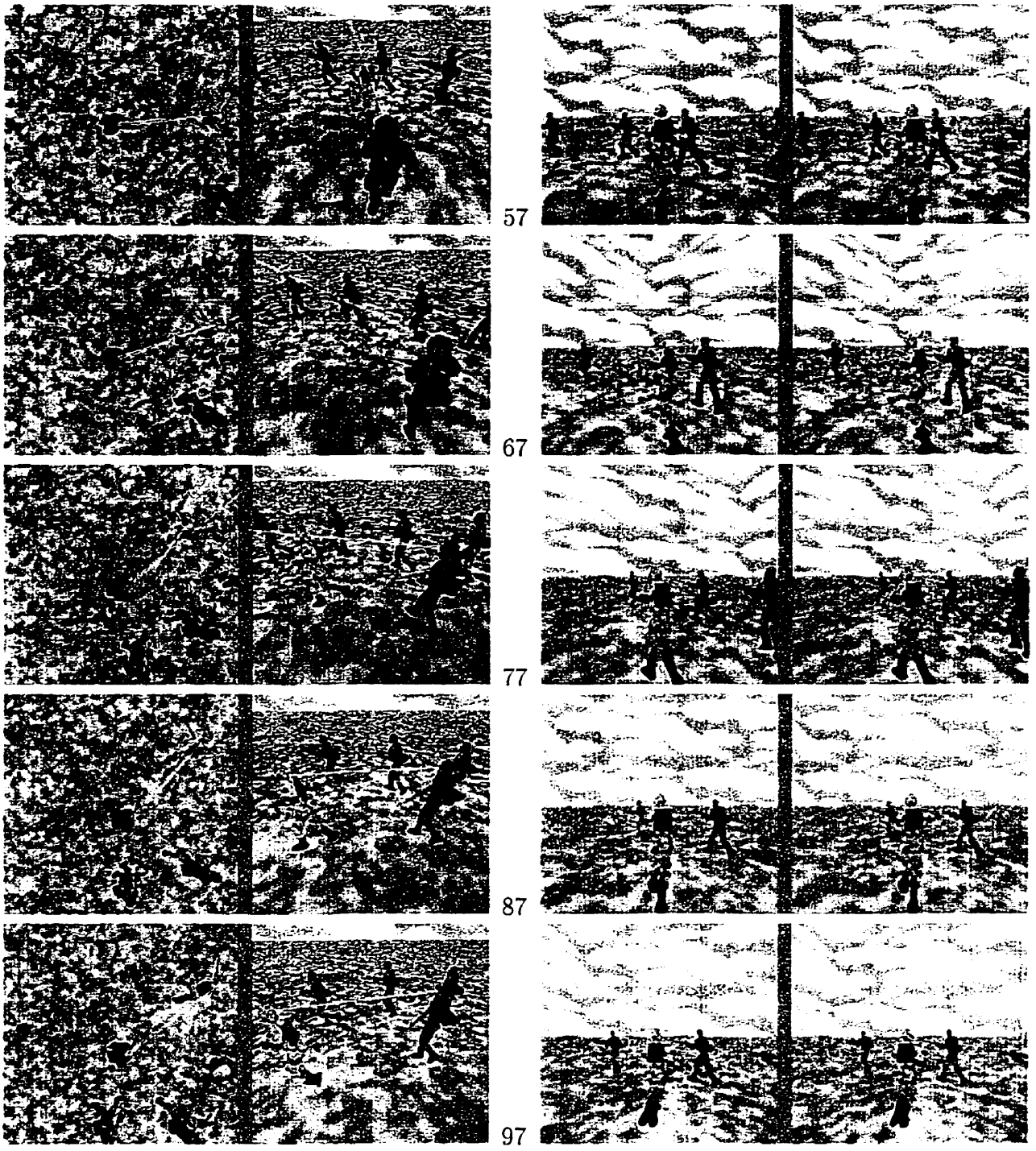


Figure 7.5: Left column: Top and side view of the soldier animat tracking another soldier. Right column: Right and left multi-scale retinal images from the animat's stereo vision eyes.

Furthermore, there is no graphics model of the observer animat, other than its arms and weapons, since the perspective is always from the first person. The DOOM animat also has fewer degrees of freedom.

7.3.1 The DOOM Graphics Engine

The DOOM graphics engine renders a 3D interior scene. This virtual world is made up of connected sectors that have a floor and ceiling height. The sectors are formed by wall lines, animated and transparent textures, 2D sprites, and one lightlevel per sector with distance cueing. There are 4 locomotional degrees of freedom:

- MOVE FORWARD,
- MOVE BACKWARD,
- TURN RIGHT,
- TURN LEFT.

Obviously the DOOM animat has fewer degrees of freedom than DI-Guy, thus forcing certain constraints on the animat vision system. Since the user is afforded a first-person perspective rendition of the environment, the task of implementing virtual eyes is trivial. Although a single eye is already implemented, there is no independent control over the motion of the observer's head relative to the body. Hence, the only way to gaze at a target is to steer the observer animat in the direction of the target.

7.3.2 Animat Vision using DOOM

The animat vision that we have implemented into DOOM is a simplified version of the full prototype implemented in the fish and the DI-Guy animats. The animat vision system starts by detecting simple movements in the environment using a simple motion detection technique based on the difference between two consecutive frames. This suffices to segment out objects in motion, thus, focusing the attention of the DOOM animat on interesting targets. The segmented target can be a potential enemy. Hence, the doom

animat compares it with color mental models of known targets using the modified color histogram intersection methods. If there is a match, color histogram backprojection is employed to estimate the location of the target. The doom animat then turns in the direction of the target and fires its weapon. Fig. 7.6 shows a sequence of image frames of the DOOM animat moving in its environment, employing vision algorithms to detect and recognize an enemy, then localize it and turn to fire at it.

If, however, the segmented target does not match with any of the mental models, it is considered harmless and the doom animat disregards it. Fig. 7.7-(a) shows an example of a segmented target at frame 166 from one of the animat's stereo eyes. Note that most of the background is removed except for some of the walls. Figure 7.7-(b) is a color model image of a potential target known by the doom animat. Fig. 7.7-(c) shows the result of blurring the histogram backprojected image of the segmented target in (a). The location of the maximum intensity value in this image localizes the target. The rest of the sequence in Fig. 7.6 from frame 168 to the end, which shows the doom animat turning in pursuit of the localized target and eventually shooting it down, is a direct result of visual processing.

7.4 Summary

In this chapter, the animat vision system which was implemented for the artificial fish world in Chapters 4, 5, and 6, was adapted and integrated into two different virtual environments, the DI-Guy virtual human and the DOOM first-person interactive combat game. In the former, the full animat vision prototype system was implemented in the DI-Guy soldier which served as a virtual robotic agent with binocular mutiresolution retinas, visual field stabilization, color object recognition and localization, target foveation, vergence of left and right eyes, and saccadic eye movements to fixate and track interesting targets. The DOOM environment had fewer degrees of freedom, thus forcing some constraints on the animat vision system. Nevertheless, this implementation was useful enough to allow the doom warrior to move around in its environment detecting poten-

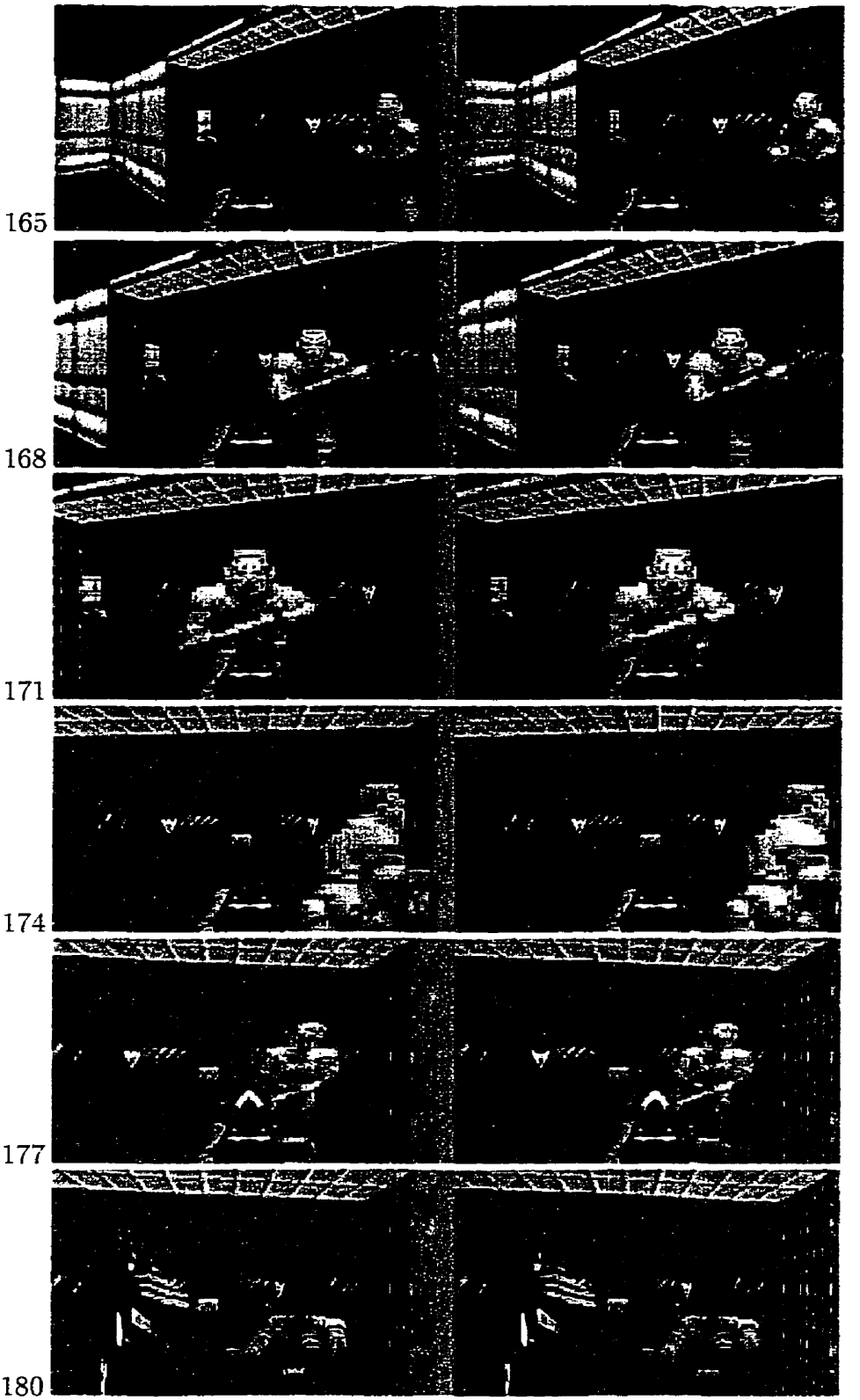


Figure 7.6: Stereo retinal image sequence from the doom animat's stereo vision eyes.

(a) (b) (c)

Figure 7.7: (a) Segmented enemy target at frame 166. (b) The model image of the target detected by the doom animat. (c) The target localized from the blurred histogram backprojection of the image in (a).

tial targets using motion segmentation, recognizing the detected targets and localizing them using the same color histogram intersection and backprojection methods developed for the fish world, and finally shooting down its enemies without the intervention of an external player.

Chapter 8

Conclusion and Future Directions

This thesis has presented work that spans the fields of computer vision, artificial life, and computer graphics. Our research was motivated in part by the realization that many active vision researchers would rather not have their progress impeded by the limitations and complications of currently available hardware. Animat vision offers a viable, purely software alternative to the prevailing hardware vision mindset. Our approach is uniquely defined by the convergence of advanced physics-based artificial life modeling of natural animals, efficient photorealistic rendering of 3D virtual worlds on standard computer graphics workstations, and active computer vision algorithms.

To demonstrate the animat vision approach, we have employed a physics-based virtual world inhabited by lifelike artifical animals. Artificial animals are virtual zoological robots ("zoobots") that offer active vision researchers greater mobility and maneuverability, lower cost, and higher reliability/repeatability than can be expected from present-day physical robots. The visual fidelity of the virtual world is sufficient for the automated visual analysis of the retinal image streams acquired by the artificial animal to be a significant challenge. Moreover, physics-based virtual robots are governed in their virtual world by the same principles that physical robots are subject to in the physical world, hence they share the attraction of situated physical robots for the purposes of active vision research.

On the one hand, skeptics may regard it as a large leap of faith that the animat vision paradigm could be useful to real world computer vision and robotics. They may

argue, for example, that a great deal of control theory would likely need to be applied to real world mechanical systems and that, for example, segmentation of real-world images is a notorious stumbling block, if not a brick wall. Optimists, on the other hand, may counter that we are already employing "real-world computer vision algorithms" in our animat vision system, in the sense that our algorithms are modified (typically improved) versions of vision algorithms reported in the vision literature with demonstrated utility in analyzing real-world scenes (as shown in [Swain and Ballard, 1991, Black and Anandan, 1993] and in our own results with natural images reported in chapter 6). Of course, we conjecture that active vision algorithms which work robustly within animat vision systems like ours should also work in a real-world context. Proving this conjecture, however, will require further work that is beyond the scope of this thesis.

The main contributions of this thesis are:

1. The animat vision approach: A new paradigm which prescribes the use of artificial animals for active vision research. It is implemented in software with realistic artificial animals which have the ability to locomote, perceive, and understand the virtual worlds in which they are situated. We show that a software implementation has significant advantages over hardware implementations.

2. A prototype active vision system was successfully implemented within the framework of a virtual marine world inhabited by artificial fish animats that emulate the appearance, motion, and behavior of natural fishes in their physical habitats. The animat vision system consists of a set of active vision algorithms for foveation and vergence of interesting targets, for retinal image stabilization and color object recognition, and for pursuit of moving targets through visually-guided navigation.

3. We further demonstrated the potential applications of the animat vision approach in the DI-Guy and DOOM environments. The same animat vision prototype was implemented for the DI-Guy animat, and, a simplified version of the algorithm was implemented for the warrior animat in DOOM.

4. We adapted and integrated a suite of active vision algorithms into the working prototype animat vision system. We integrated motion and color-based gaze control algorithms to enhance the prototype and to support more robust vision-guided navigation and selective attention abilities in the animat. We further enhanced the animat's navigation and perception abilities by combining stereo and color-based motor control algorithms which extended the animat's functionality by supporting obstacle recognition and avoidance.

5. We made improvements to the color histogram intersection methods originally introduced by Swain [Swain and Ballard, 1991]. We developed a more robust intersection measure that is invariant to scale changes. We adapted it to foveated systems with high resolution foveas and lower resolution peripheries to make use of the information present in the peripheral image and absent from the foveal image.

6. We developed an incremental motion segmentation technique that makes use of the robust optical flow estimate at each time instant to refine an initial segmentation over time as the animat navigates and acquires more retinal image frames. Also, motion and color were integrated to increase the robustness of the animat's recognition senses.

7. We developed stereo disparity algorithms based on steerable filters that make use of the color signals available naturally from the photorealistic images acquired by the animat to improve the matching process and to obtain more accurate disparity estimates. We demonstrated that this method was very effective and when combined with color cues, gave the animat the abilities required to avoid obstacles.

The fact that the animat vision system was successfully integrated into three different virtual environments—the fish world, the DI-Guy environment, and the DOOM world—demonstrates the versatility of the paradigm. We believe that it is a general vision research framework applicable to virtual robotic systems of varying degrees of complexity. The work presented in this thesis should also be relevant in whole or in part to physical

robotics (e.g., autonomous underwater vehicles). In conclusion, it appears that artificial animals in their virtual worlds can serve as a proving ground for theories that claim sensorimotor abilities in animal or robotic situated agents.

8.1 Future Work

An obvious direction for animat vision research would include augmenting the complexity of the virtual worlds (the current DI-Guy environment is especially simplistic) and the development of a more extensive arsenal of active vision algorithms to support the complete behavioral repertoire of the animats. The animat vision approach allows us to do this in stages without compromising the complete functionality of the artificial animal as demonstrated by the vision algorithm enhancements described in Chapters 5 and 6.

Another challenge is to implement effective animat vision systems within animats situated in physics-based virtual environments where more realistic environmental conditions are simulated. For example, the active vision system would have to be enhanced with robust vision algorithms that can deal with varying illumination conditions, such as variable cloudiness for outdoor environments, varying distance of the target from the light source in indoor environments, controlling the animat's vision parameters to adapt to day and night vision, and dealing with objects that have similar color signatures, to name a few. Vision algorithms of greater sophistication would have to be implemented to deal with these problems. For example, a more robust target recognition algorithm would be needed. Our modified color histogram methods can be used in conjuction with a color constancy algorithm such as that of Barnard *et al.* [Barnard *et al.*, 1996] to minimize the effects of varying lighting conditions. Also, our color object recognition algorithms may be improved by combining the color cues with other robust cues such as steerable filter features which have been used successfully in object recognition [Ballard and Wixson, 1993].

To deal with the problem of distinguishing objects with similar color signatures that are normally confused by histogram intersection, we can modify our algorithm to ignore

colors that are common to the objects of interest and only concentrate on those salient colors that are unique to the objects [Swain, 1990]. This method will, however, fail for the degenerate case when the objects of interest are all exact copies of each other with no salient colors. In this case our object recognition algorithms will initially pick one of the objects at random. Once foveated, the algorithm will have no difficulty in keeping a fix on this specific object as long as it is the only object with salient colors in the fovea. The confusion will return, however, when another copy of the foveated object appears in the fovea. This problem is not unique to computer vision since even humans will have difficulty distinguishing between identical twins wearing the exact same clothes.

Exploiting additional visual cues, such as shape, is an interesting topic for future investigation. Targets of interest in our experimental virtual environments include other moving virtual fish and humans. These objects are clearly nonrigid deformable bodies which makes the task of recognizing their continuously changing shapes a fairly complex one. For example, the virtual human soldier target is an articulated object which is composed of nonrigid parts constrained together at joints. This type of body can produce a complex variety of shapes, because of the articulate skeleton and tissue deformations due to muscle actions and gravitational effects. One possible approach to tackle this problem is to use dynamic primitives that are derived from physics-based object models similar to those used to construct the target objects the animat vision system deals with. A paradigm for tracking nonrigid 3D objects using deformable models has recently been introduced by Terzopoulos and Metaxas [Terzopoulos and Metaxas, 1992]. They use dynamic modeling primitives (such as 3D deformable cylinderical objects) that can deform as they move freely in space. Simulated physical constraints based on Lagrangian dynamics are applied between these primitives which make them responsive to applied forces derived from the visual data, thus, giving them the ability to construct dynamic models of articulated objects with deformable parts. This makes them an obvious candidate for future enhancements to our animat vision system's target recognition capabilities.

A final topic for future work would be to adapt the animat vision system to physical

robotics (e.g., autonomous vehicles). Regardless of their limitations, mobile robots have been used as an experimental platform for artificial intelligence research for more than four decades, and will remain the link between research and applications in the physical world.

Bibliography

[Adiv, 1985] G. Adiv. Determining three-dimentional motion and structure from optical flow generated by several moving objects. *IEEE Trans. PAMI*, 7(4):384–401, 1985.

[Adler, 1975] H. E. Adler. *Fish Behavior: Why Fishes do What They Do.* T.F.H Publications, Neptune City, NJ, 1975.

[Aloimonos *et al.*, 1987] Y. Aloimonos, A. Bandyopadhyay, and I. Weiss. Active vision. *Int. J. Computer Vision*, pages 333 – 356, 1987.

[Anandan, 1989] P. Anandan. A computational framework and algorithm for the measurement of visual motion. *Inter. J. Computer Vision*, 2:283–310, 1989.

[Arakawa and Etoh, 1995] H. Arakawa and M. Etoh. An integration algorithm for stereo, motion and color in real-time applications. *IEICE Transactions on Information and Systems*, E78-D(12):1615 – 1620, December 1995.

[Bajcsy, 1988] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.

[Ballard and Brown, 1982] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[Ballard and Brown, 1992] D.H. Ballard and C.M. Brown. Principles of animate vision. *CVGIP: Image Understanding*, 56(1):3 – 21, July 1992.

[Ballard and Wixson, 1993] D.H. Ballard and L.E. Wixson. Object recognition using steerable filters at multiple scales. In *Proc. IEEE Workshop on Qualitative Vision*, pages 2 – 10, Los Alamitos, CA, June 1993.

[Ballard, 1991] D. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[Barnard *et al.*, 1996] K. Barnard, G. Finlayson, and B. Funt. Colour constancy for scenes with varying illumination. *ECCV*, B:3–15, 1996.

[BDI, 1998] BDI, editor. *DI-Guy User Manual, Version 3.* Boston Dynamics Inc., Cambridge, MA, 1998.

[Beer, 1990] R. Beer. *Intelligence as Adaptive Behavior.* Academic press, NY, 1990.

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. Euro. Conf. Computer Vision (ECCV'92)*, pages 237 - 252, Portofino, Italy, 1992.

[Black and Anandan, 1990] M.J. Black and P. Anandan. A model for the detection of motion over time. In *Proc. Inter. Conf. Computer Vision (ICCV'90)*, pages 33–37, Osaka, Japan, 1990.

[Black and Anandan, 1993] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Inter. Conf. Computer Vision (ICCV'93)*, pages 231–236, Berlin, 1993.

[Black, 1992] M.J. Black. Robust incremental optical flow. Technical Report YALEU/DCS/RR-923, Yale University, Dept. of Computer Science, 1992.

[Blake and Yuille, 1992] A. Blake and A. Yuille, editors. *Active Vision.* MIT Press, Cambridge, MA, 1992.

[Blake and Zisserman, 1987] A. Blake and A. Zisserman, editors. *Visual Reconstruction.* The MIT Press, Cambridge, Massachusetts, 1987.

[Blumberg and Galyean, 1995] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Computer Graphics* Proceedings, Annual Conference Series, Proc. SIGGRAPH '95 (Los Angeles, CA), pages 47–55. ACM SIGGRAPH, August 1995.

[Braitenberg, 1984] V. Braitenberg. *Vehicles, Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

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

[Brown and Terzopoulos, 1994] C. Brown and D. Terzopoulos. *Real-Time Computer Vision.* Cambridge University Press, Cambridge, UK, 1994.

[Brown *et al.*, 1992] C. Brown, D. Coombs, and J. Soong. Real-time smooth pursuit tracking. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 8, pages 123–136. MIT Press, Cambridge, MA, 1992.

[Burt and Adelson, 1983] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, 31(4):532–540, 1983.

[Burt *et al.*, 1989] P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczynski, W.A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera: An application of dynamic motion analysis. *Proc. IEEE Workshop Visual Motion*, pages 2 – 12, March 1989.

[Campani *et al.*, 1995] M. Campani, A. Giachetti, and V. Torre. Optic flow and autonomous navigation. *Perception*, 24:253–267, 1995.

[Cedras and Shah, 1995] C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[Chen and Bovik, 1995] T.Y. Chen and A.C. Bovik. Stereo disparity from multiscale processing of local image phase. In *Proc. Fifth Inter. Conf. Computer Vision (ICCV'95)*, pages 188 – 193, MIT, Cambridge, MA, June 20 - 23 1995.

[Cho and Cho, 1994] Y.C. Cho and H.S. Cho. A stereo vision-based obstacle detecting method for mobile robot navigation. *Robotica*, 12(3):203 – 216, May - June 1994.

[Coombs and Brown, 1991] D.J. Coombs and C.M. Brown. Cooperative gaze holding in binocular vision. *IEEE Control Systems Magazine*, pages 24 – 33, June 1991.

[Coombs and Brown, 1993] D.J. Coombs and C.M. Brown. Real-time binocular smooth pursuit. *Inter. J. Computer Vision*, 11(2):147 – 164, 1993.

[Coombs and Roberts, 1992] D.J. Coombs and K. Roberts. Bee-bot: Using peripheral optical flow to avoid obstacles. *Proc. SPIE Intelligent Robots and Computer Vision XI*, 1825:714 – 721, 1992.

[Coombs *et al.*, 1995] D.J. Coombs, M. Herman, T. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. In *Proc. Fifth Inter. Conf. Computer Vision (ICCV'95)*, pages 276 – 283, MIT, Cambridge, MA, June 20 - 23 1995.

[Dubuisson and Jain, 1993] M. Dubuisson and A.K. Jain. Object contour extraction using color and motion. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'93)*, pages 471–476, 1993.

[Dubuisson and Jain, 1994a] M. Dubuisson and A.K. Jain. 2d matching of 3D moving objects in color outdoor scenes. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'94)*, pages 887–891, 1994.

[Dubuisson and Jain, 1994b] M. Dubuisson and A.K. Jain. Fusing color and edge information for object matching. In *Proc. Inter. Conf. Computer Vision (ICCV'94)*, pages 982–986, 1994.

[Ennesser and Medioni, 1993] F. Ennesser and G. Medioni. Finding waldo, or focus of attention using local color information. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'93)*, pages 711 – 712, 1993.

[Fleet and Jepson, 1990] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *Inter. J. Computer Vision*, 5:77–104, 1990.

[Fleet *et al.*, 1991] D. Fleet, A. Jepson, and M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53:198 – 210, 1991.

[Freeman and Adelson, 1991] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. PAMI*, 13(9):891 – 906, September 1991.

[Gibson, 1979] J. J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, MA, 1979.

[Gong and Sakauchi, 1992] Y. Gong and M. Sakauchi. An object-oriented method for color video image classification using the color and motion features of video images. In *2nd. Inter. Conf. on Automation, Robotics and Computer Vision*, Singapore, 1992.

[Grimson *et al.*, 1994] W.E.L. Grimson, A.L. Ratan, G. Klanderman, and A. O'Donnell. An active visual attention system to play 'where's waldo'. In *Proc. of the Image Understanding Workshop*, volume 2, pages 1059 –1065, Monterey, CA, 13 - 16 Nov. 1994.

[Grosso *et al.*, 1995] E. Grosso, R. Manzotti, R. Tiso, and G. Sandini. A space-variant approach to oculomotor control. In *Proc. Int. Symp. on Computer Vision*, pages 509–514, Florida, 1995.

[Hagen and Heyerdahl, 1992] E. Hagen and E. Heyerdahl. Navigation by optical flow. In *Proc. 11th IAPR, Int. Conf. on Pattern Recognition*, pages 700–703, 1992.

[Hampel, 1974] F.R. Hampel. The influence curve and its role in robust estimation. *J. Amer. Statistical Association*, 69(346):383–393, 1974.

[Horn and Jr., 1988] B.K.P. Horn and E.J. Weldon Jr. Direct methods for recovering motion. *Int. J. Computer Vision*, 2:51–76, 1988.

[Horn and Schunck, 1981] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[Horn, 1986] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[Horridge, 1993] G.A. Horridge. What can engineers learn from insect vision? In *Proc. Inter. Conf. on Systems, Man and Cybernetics. Systems Engineering in the Service of Humans.*, pages 138–143, Le Touquet, France, 1993.

[Huang and Chen, 1995] C.L. Huang and Y.T. Chen. Motion estimation method using a 3D steerable filter. *Image and Vision Computing*, 13(1):21–32, 1995.

[Husbands, 1994] P. Husbands, editor. *From Animals to Animats: The 3rd International Conf. on Simulation of Adaptive Behavior*, Cambridge, MA, 1994. MIT Press.

[Irani and Peleg, 1992] M. Irani and S. Peleg. Image sequence enhancemenmt using multiple motion analysis. In *Proc. 11th IAPR, Int. Conf. on Pattern Recognition*, pages 216–221, 1992.

[Irani *et al.*, 1994] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. *IEEE Workshop on Visual Motion*, pages 454 – 460, 1994.

[Jagersand, 1995] M. Jagersand. Saliency maps and attention selection in scale and spatial coordinates: An information theoretic approach. In *Proc. Inter. Conf. Computer Vision (ICCV'95)*, pages 195–202, MIT, Cambridge, Massachusetts, June 20 - 23 1995.

[Jenkin and Tsotsos, 1994] M.R.M. Jenkin and J.K. Tsotsos. Active stereo vision and cyclotorsion. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'94)*, pages 806 – 811, 1994.

[Jepson and Black, 1993] A.D. Jepson and M.J. Black. Mixture models for optical flow computation. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'93)*, pages 760–761, New York, 1993.

[Joarder and Raviv, 1994] K. Joarder and D. Raviv. A new method to calculate looming for autonomous obstacle avoidance. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'94)*, pages 777 – 780, 1994.

[Jones and Malik, 1992] D.G. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Proc. Euro. Conf. Computer Vision (ECCV'92)*, pages 395 – 410, Portofino, Italy, 1992.

[Koechling *et al.*, 1998] J. Koechling, A. Crane, and M. Raibert. Applications of realistic human entities using DI-Guy. In *Proc. of Spring Simulation Interoperability Workshop*, Orlando, Florida, 1998.

[Langley *et al.*, 1990] K. Langley, T.J. Atherton, R.G. Wilson, and M.H.E. Larcombe. Vertical and horizontal disparities from phase. In *Proc. Euro. Conf. Computer Vision (ECCV'90)*, pages 315 – 325, 1990.

[Levine, 1985] M.D. Levine, editor. *Vision in man and machine.* McGraw-Hill, Inc., New York, 1985.

[Lucas and Kanade, 1981] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. Image Understanding Workshop*, pages 121–130, 1981.

[Maes *et al.*, 1994] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. Interacting with animated autonomous agents. Believable Agents: Working Notes of the AAAI Spring Symposium Series, March 1994.

[Maes, 1991] P. Maes, editor. *Designing Autonomous Agents.* MIT Press, Cambridge, MA, 1991.

[Marr and Poggio, 1979] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. R. Soc. Lond. B.*, 204:301 – 328, 1979.

[Marr, 1982] D.C. Marr, editor. *Vision.* Freeman, Oxford, 1982.

[Maver and Bajcsy, 1990] J. Maver and R. Bajcsy. How to decide from the first view where to look next. In *Proc. DARPA Image Understanding Workshop*, 1990.

[Meyer and Bouthemy, 1992] F. Meyer and P. Bouthemy. Estimation of time-to-collision maps from first order motion models and normal flows. In *Proc. 11th IAPR, Int. Conf. on Pattern Recognition*, pages 78–82, 1992.

[Murray *et al.*, 1995] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, and I.D. Reid. Driving saccade to pursuit using image motion. *Inter. J. Computer Vision*, 16:205–228, 1995.

[Nguyen and Cohen, 1992] H.H. Nguyen and P. Cohen. Correspondence from color shading. In *Proc. 11th IAPR Inter. Conf. on Pattern Recognition*, volume 1, pages 113 – 116, 1992.

[Olson and Lockwood, 1992] T.J. Olson and R.J. Lockwood. Fixation-based filtering. *Proc. SPIE Intelligent Robots and Computer Vision XI*, 1825:685 – 695, 1992.

[Peleg and Rom, 1990] S. Peleg and H. Rom. Motion based segmentation. In *Proc. Int. Conf. on Pattern Recognition*, pages 109–113, 1990.

[Prokopowicz and Cooper, 1995] P.N. Prokopowicz and P.R. Cooper. The dynamic retina: Contrast and motion detection for active vision. *Inter. J. Computer Vision*, 16:191–204, 1995.

[Rabie and Terzopoulos, 1996] T.F. Rabie and D. Terzopoulos. Motion and color analysis for animat perception. In *Proc. Thirteenth National Conf. on Artificial Intelligence (AAAI'96)*, pages 1090–1097, Portland, Oregon, August 4-8 1996.

[Rabie and Terzopoulos, 1998] T.F. Rabie and D. Terzopoulos. Stereo and color analysis for dynamic obstacle avoidance. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, California, June 23-25 1998.

[Renault *et al.*, 1990] O. Renault, N. Magnenat-Thalmann, and D. Thalmann. A vision-based approach to behavioural animation. *Visualization and Computer Animation*, 1:18–21, 1990.

[Reynolds, 1987] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[Rimey and Brown, 1994] R.D. Rimey and C.M. Brown. Control of selective perception using bayes nets and decision theory. *Int. J. Computer Vision*, 12(2,3):173–208, April 1994.

[Roberts, 1965] L.G. Roberts. Machine perception of three-dimensional solids. In Tippett et al., editor, *Optical and Electro-Optical Information Processing*, chapter 9, pages 159 – 197. MIT Press, Cambridge, MA, 1965.

[Salgian and Ballard, 1998] G. Salgian and D.H. Ballard. Visual routines for autonomous driving. In *Proc. Sixth Inter. Conf. Computer Vision (ICCV'98)*, Bombay, India, January 1998.

[Sanger, 1988] T. Sanger. Stereo disparity computation using gabor filters. *Biological Cybernetics*, 59:405 – 418, 1988.

[Seelen *et al.*, 1995] W.V. Seelen, S. Bohrer, J. Kopecz, and W.M. Theimer. A neural architecture for visual information processing. *Inter. J. Computer Vision*, 16:229–260, 1995.

[Shigang *et al.*, 1995] L. Shigang, A. Ochi, and S. Tsuji. Route description by landmarks. In *Proc. of the Intelligent Vehicles '95 Symposium*, pages 454 – 459, Detroit, MI, 25 - 26 September 1995.

[Simoncelli and Freeman, 1995] E.P. Simoncelli and W.T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE Inter. Conf. on Image Processing*, pages 444 – 447, Washington, DC, October 23-26 1995.

[Spetsakis, 1994] M.E. Spetsakis. Optical flow estimation using discontinuity conforming filters. In *Proc. British Machine Vision Conf. Vol. II*, pages 569–578, 1994.

[Swain and Ballard, 1991] M. Swain and D. Ballard. Color indexing. *Inter. J. Computer Vision*, 7:11 – 32, 1991.

[Swain and Stricker, 1993] M.J. Swain and M.A. Stricker. Promising directions in active vision. *Inter. J. Computer Vision*, 11(2):109 – 126, 1993.

[Swain *et al.*, 1992] M.J. Swain, R.E. Kahn, and D.H. Ballard. Low resolution cues for guiding saccadic eye movements. In *Proc. Inter. Conf. Computer Vision (ICCV'92)*, pages 737–740, 1992.

[Swain, 1990] M.J. Swain. Color indexing. Technical Report TR-360, University of Rochester, Computer Science, 1990.

[Terzopoulos and Metaxas, 1992] D. Terzopoulos and D. Metaxas. Tracking nonrigid 3D objects. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 5, pages 75–89. MIT Press, Cambridge, MA, 1992.

[Terzopoulos and Rabie, 1995] D. Terzopoulos and T.F. Rabie. Animat vision: Active vision in artificial animals. In *Proc. Fifth Inter. Conf. Computer Vision (ICCV'95)*, pages 801 – 808, MIT, Cambridge, MA, June 20 - 23 1995.

[Terzopoulos and Rabie, 1997] D. Terzopoulos and T.F. Rabie. Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research*, 1(1):2–19, September 1997.

[Terzopoulos *et al.*, 1988] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[Terzopoulos *et al.*, 1994] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.

[Tsotsos *et al.*, 1995] J.K. Tsotsos, S.M. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78(1-2):507–547, 1995.

[Tu and Terzopoulos, 1994a] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Computer Graphics* Proceedings, Annual Conference Series, Proc. SIGGRAPH '94 (Orlando, FL), pages 43–50. ACM SIGGRAPH, July 1994.

[Tu and Terzopoulos, 1994b] X. Tu and D. Terzopoulos. Perceptual modeling for the behavioral animation of fishes. In *Proc. 2nd Pacific Conf. on Computer Graphics*, Beijing, China, 1994.

[Walter, 1953] W.G. Walter, editor. *The living brain.* Penguin, Harmondsworth, 1953.

[Wang and Adelson, 1993] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *Proc. Computer Vision and Pattern Recognition Conf. (CVP'93R)*, pages 361–366, 1993.

[Weber and Malik, 1993] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. In *Proc. Inter. Conf. Computer Vision (ICCV '93)*, pages 12–20, 1993.

[Wilson, 1991] S. W. Wilson. The animat path to AI. In J.-A. Meyer and S. Wilson, editors, *From Animals to Animats*, pages 15–21. MIT Press, Cambridge, MA, 1991.

[Wixson and Ballard, 1989] L.E. Wixson and D.H. Ballard. Real-time detection of multicolored objects. In *SPIE Symp. on Advances in Intelligent Robotics Systems*, pages 435–446, 1989.

[Ye and Tsotsos, 1995] Y. Ye and J.K. Tsotsos. Where to look next in 3D object search. In *Proc. Int. Symp. on Computer Vision*, pages 539–544, Florida, 1995.

[Young, 1986] R.A. Young. Simulation of human retinal function with the Gaussian derivative model. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'86)*, pages 564 – 569, 1986.