Published in *Artificial Life V: Proc. Fifth Inter. Conf. on the Synthesis and Simulation of Living Systems*, Nara, Japan, May, 1996.

Perception and Learning in Artificial Animals

Demetri Terzopoulos, Tamer Rabie and **Radek Grzeszczuk**
Department of Computer Science, University of Toronto
6 King's College Road, Toronto, ON M5S 3H5, Canada
e-mail: {dt|tamer|radek}@cs.toronto.edu

Abstract

We employ a virtual marine world inhabited by realistic artificial animals as an ALife laboratory for developing and evaluating zoomimetic perception and learning algorithms. In particular, we propose active perception strategies that enable artificial marine animals to navigate purposefully through their world by using computer vision algorithms to analyze the foveated retinal image streams acquired by their eyes. We also demonstrate learning algorithms that enable artificial marine animals to acquire complex motor skills similar to those displayed by trained marine mammals at aquatic theme parks.

Figure 1: *Artificial fishes swimming among aquatic plants in a physics-based virtual marine environment.*

1 Introduction

A recent result of artificial life research is a virtual world inhabited by artificial animals and plants that emulate some of the fauna and flora of natural marine environments [1]. In this paper, we employ this highly realistic virtual world as an artificial zoological laboratory. The laboratory facilitates the investigation of open problems related to biological information processing in animals, and it has enabled us to develop and evaluate zoomimetic perception and learning algorithms.

The psychologist J.J. Gibson studied (in pre-computational terms) the perceptual problems faced by an active observer situated in the dynamic environment [2].[1] We present a prototype active perception system that enables artificial marine animals to navigate purposefully through their world by analyzing the retinal image streams acquired by their eyes. Retinal image analysis is carried out using computer vision algorithms. We equip our artificial animals with directable, virtual eyes capable of foveal vision. This aspect of our work is related to that of Cliff and Bullock [5], but our realistic animal models have enabled us to progress a great deal further.[2] Our goal is to engineer general-purpose vision systems for artificial animals possessing zoomimetic eyes that image continuous 3D photorealistic worlds. We assemble a suite of vision algorithms that support foveation, retinal image stabilization, color object recognition, and perceptually-guided navigation. These perceptual capabilities allow our artificial fishes to pursue moving targets, such as fellow fishes. They do so by saccading their eyes to maintain foveation on targets as they control their muscle-actuated bodies to locomote in the direction of their gaze.

We also demonstrate motor learning algorithms that enable artificial marine animals to acquire some nontrivial motor skills through practice. In particular, these algorithms enable an artificial dolphin to learn to execute stunts not unlike those performed by trained marine mammals to the delight of spectators at aquatic theme parks. This research builds upon the low-level motor learning algorithms described in our prior work [1]. It reinforces our earlier claim that biomechanical models of animals situated in physics-based worlds are fertile ground for learning novel sensorimotor control strategies.

2 Review of Artificial Fishes

Artificial fishes are autonomous agents inhabiting a realistic, physics-based virtual marine world (Fig. 1). Each agent has a deformable body actuated by internal muscles. The body also

[1] Computational versions of Gibson's paradigm were developed in computer vision by Bajcsy [3] and Ballard [4] under the names of "active perception" and "animate vision", respectively.

[2] Cliff and Bullock [5] were concerned with the evolution of simple visually guided behaviors using Wilson's animat in a discrete 2D grid world.

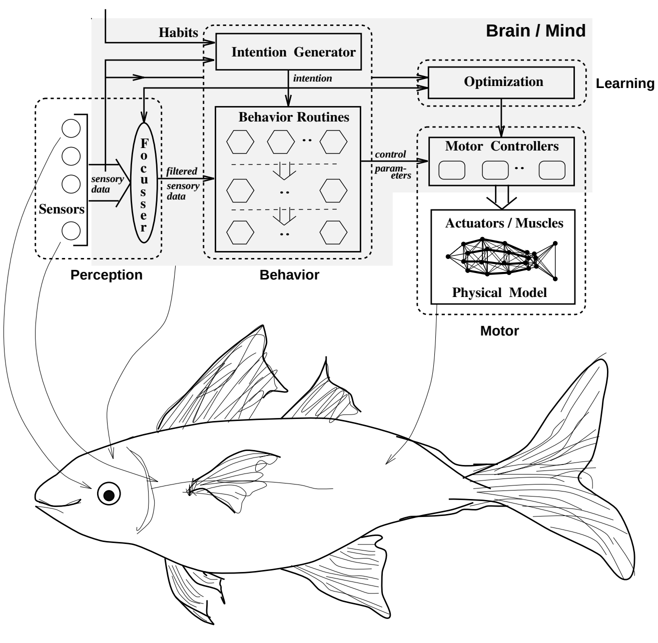

Figure 2: *Artificial fish model (from [1]).*

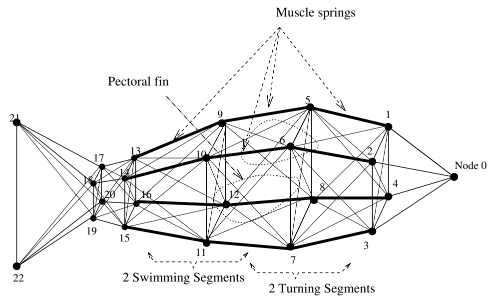

Figure 3: *Biomechanical fish model. Nodes denote lumped masses. Lines indicate uniaxial elastic elements (shown at natural length). Bold lines indicate muscle elements.*

harbors eyes and a brain with motor, perception, behavior, and learning centers (Fig. 2). Through controlled muscle actions, artificial fishes are able to swim through simulated water in accordance with hydrodynamic principles. Their functional fins enable them to locomote, maintain balance, and maneuver in the water. Thus the model captures not just the form and appearance of the animal, but also the basic physics of the animal in its environment. Although rudimentary compared to those of real animals, the brains of artificial fishes are nonetheless able to learn some basic motor functions and carry out perceptually guided motor tasks. The behavior center of the artificial fish's brain mediates between its perception system and its motor system, harnessing the dynamics of the perception-action cycle. The innate character of the fish is determined by fixed habits. Its dynamic mental state is represented by a set of mental variables—hunger, libido, and fear. An intention generator serves as the fish's cognitive faculty, arbitrating the artificial fish's behavioral repertoire in accordance with its perceptual awareness of the virtual world. The behavioral repertoire includes primitive, reflexive behavior routines, such as collision avoidance, as well as more sophisticated motivational behavior routines such as foraging, preying, schooling, and mating.

The details of the artificial fish model are presented in the paper [1] (or see an earlier version in the ALIFE IV Proceedings). The reminder of this section covers details about the motor system which are necessary to understand the learning and vision algorithms to follow.

The motor system comprises the dynamic model of the fish including its muscle actuators and a set of motor controllers (MCs). Fig. 3 illustrates the biomechanical body model which produces realistic piscine locomotion using only 23 lumped masses and 91 elastic elements. These mechanical components are interconnected so as to maintain the structural integrity of the body as it flexes due to the action of its 12 contractile muscles.

Artificial fishes locomote like real fishes, by autonomously contracting their muscles. 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 are continually integrated through time by a numerical simulator.[3]

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 that enable it to navigate freely in its 3D aquatic world by pitching, rolling, and yawing its body. Additional motor controllers coordinate the fin actions.

3 Perception

This section describes a vision system for artificial fish which is based solely on retinal image analysis via computer vision algorithms [6].[4] We have developed a prototype active

[3] The artificial fish model achieves a good compromise between realism and computational efficiency. For example, 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, and full color texture mapped GL rendering at video resolution can take 5 seconds or more per frame.

[4] By contrast, in our prior work [1] the artificial fishes rely on *simulated perception*—a "perceptual oracle" which satisfies the fish's sensory needs by directly interrogating the 3D world model; i.e., the autonomous agents were permitted direct access to the geometric and

vision system for our artificial animals. The system is designed for extensibility, so that it can eventually support the broad repertoire of individual and group behaviors of artificial fishes. However, our approach to perception applies to any animal, not just fishes. In fact, we view artificial fishes as virtual piscine robots, and we do not restrict ourselves to modeling the perceptual mechanisms of real fishes [7]. Indeed, it will soon become evident that our piscine robots sense their world through virtual eyes that are patterned after those of primates!

3.1 Active Vision System

The basic functionality of the active vision system starts with binocular perspective projection of the color 3D world onto the 2D retinas of the artificial fish. Retinal imaging is accomplished by photorealistic graphics rendering of the world from the animal's point of view. This projection respects occlusion relationships among objects. It forms spatially nonuniform visual fields with high resolution foveas and low resolution peripheries. Based on an analysis of the incoming color retinal image stream, the perception center of the artificial fish's brain supplies saccade control signals to its eyes and stabilize the visual fields during locomotion, to attend to interesting targets based on color, and to keep targets fixated. The artificial fish is thus able to approach and track other artificial fishes using sensorimotor control.

Fig. 4 is a block diagram of the active vision system showing two main modules that control foveation of the eyes and retinal image stabilization.

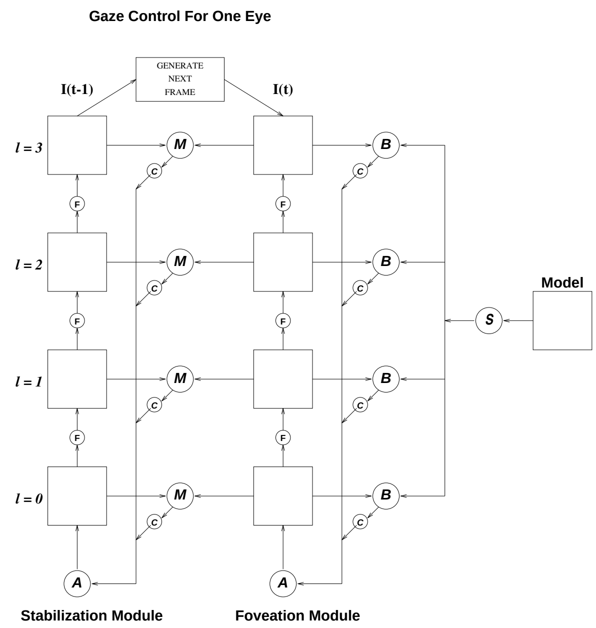

Figure 4: *The active vision system. The flow of the algorithm is from right to left. A: Update gaze angles (θ, ϕ) and saccade using these angles, B: Search current level for model target and if found localize it, else search lower level, C: Select level to be processed (see text), F: Reduce field of view for next level and render, M: Compute a general translational displacement vector (u, v) between images $I(t-1)$ and $I(t)$, S: Scale the color histogram of the model for use by the current level.*

Eyes and Foveated Retinal Imaging The artificial fish is capable of binocular vision and possesses an ocular motor system that controls eye movements [8]. The movements of each eye are controlled through two gaze angles (θ, ϕ) which specify the horizontal and vertical rotation of the eyeball, respectively, with respect to the head coordinate frame (when $\theta = \phi = 0°$, the eye looks straight ahead).

Each eye is implemented as four coaxial virtual cameras to approximate the spatially nonuniform, foveal/peripheral imaging capabilities typical of biological eyes. Fig. 5(a) shows an example of the 64×64 images that are rendered (using the GL library and SGI graphics pipeline) by the four coaxial cameras of the left and right eye. The level $l = 0$ camera has the widest field of view (about $120°$). The field of view decreases with increasing l. The highest resolution image at level $l = 3$ is the fovea and the other images form the visual periphery. Fig. 5(b) shows the 512×512 binocular retinal images composited from the coaxial images at the top of the figure (the component images are expanded by factors 2^{l-3}). To reveal the retinal image structure in the figure, we have placed a white border around each magnified component image. Significant computational efficiency accrues from processing four 64×64 component images rather than a uniform 512×512 retinal image.

Foveation by Color Object Detection The brain 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 mental models of prey fish. The models are stored as a list of 64×64 RGB color images in the fish's visual memory.

To detect and localize any target that may be imaged in the low resolution periphery of its retinas, the active vision system of the fish employs an improved version of a color indexing algorithm proposed by Swain [9]. Since each model object has a unique color histogram signature, it can be detected in the retinal image by histogram intersection and localized by histogram backprojection. Our algorithms are explained more fully in [10].

Saccadic Eye Movements When a target is detected in the visual periphery, the eyes will saccade to the angular offset of the object 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.

Module *A* in Fig. 4 performs the saccades by incrementing the gaze angles (θ, ϕ) in order to rotate the eyes to achieve the required gaze direction.

photometric information available to the graphics rendering engine, as well as object identity and dynamic state information about the physics-based world model.

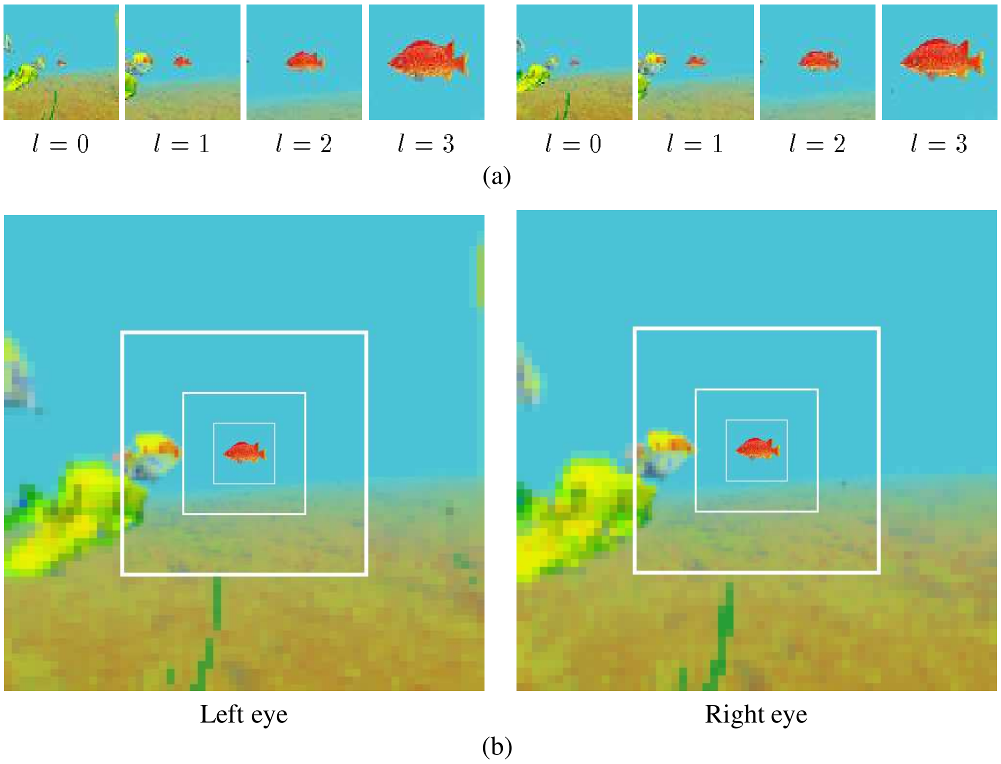

Figure 5: *Binocular retinal imaging (monochrome versions of original color images). (a) 4 component images;* $l = 0, 1, 2,$ *are peripheral images;* $l = 3$ *is foveal image. (b) Composited retinal images (borders of composited component images are shown in white).*

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. Once a target is verged in both foveas, the stabilization process (Fig. 4) assumes the task of keeping the target foveated as the fish locomotes. Thus, it emulates the optokinetic reflex in animals.

Stabilization is achieved by computing the overall translational displacement (u, v) of light patterns between the current foveal image and that from the previous time instant, and updating the gaze angles to compensate. The displacement is computed as a translational offset in the retinotopic coordinate system by a least squares minimization of the optical flow between image frames at times t and $t-1$ [6].

The optical flow stabilization method is robust 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 foveation module is invoked 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 retinal images must be correlated to keep them verged accurately on the target. Referring to Fig. 6, the vergence angle is $\theta_V = (\theta_R - \theta_L)$ and its magnitude increases as the fish comes closer to the target. Therefore, once the eyes are verged on a target, it is straightforward for the fish vision system to estimate the range to the target by triangulation using the gaze angles.

3.2 Vision-Guided Navigation

The fish can use the gaze direction for the purposes of navigation 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. 6, and the ϕ angles are similarly used to compute an up/down turn angle ϕ_P. The fish's turn motor controllers (see Section 2) are invoked to execute a left/right turn—left-turn-MC for an above-threshold positive θ_P and right-turn-MC for negative θ_P—with $|\theta_P|$ as parameter. Up/down turn motor commands are issued to the fish's pectoral fins, with an above-threshold positive ϕ_P interpreted as "up" and negative as "down".

The problem of pursuing a moving target that has been fixated in the foveas of the fish's eyes is simplified by the gaze control mechanism described above. The fish can robustly track a target in its fovea and locomote to follow it around the environment 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 prey fish whose color histogram model is stored in the memory of a predator fish equipped with the active vision system. Fig. 7 shows plots of the

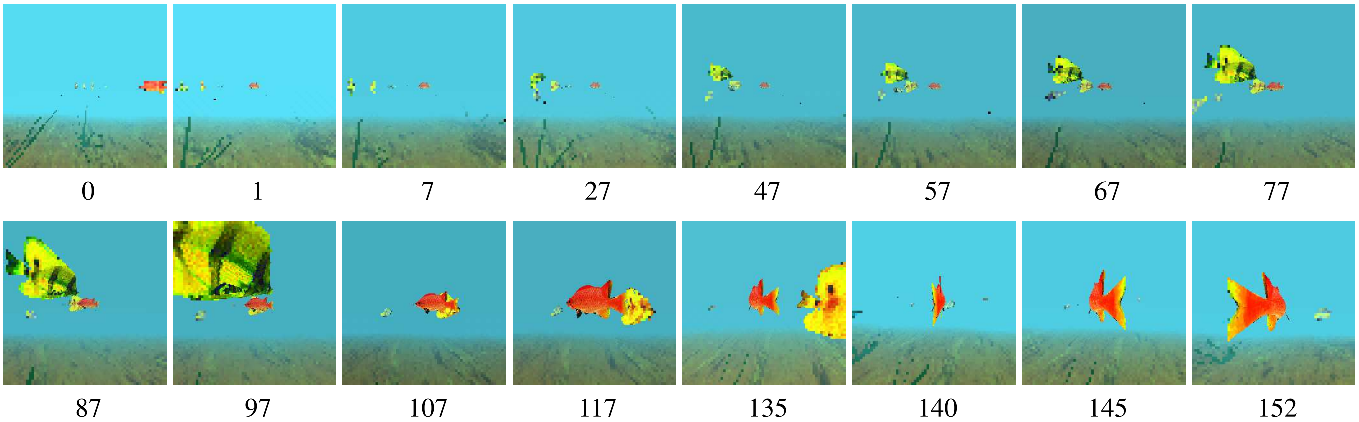

Figure 8: *Retinal image sequence from the left eye of the active vision fish as it detects and foveates on a reddish fish target and swims in pursuit of the target (monochrome versions of original color images). The target appears in the periphery (middle right) in frame 0 and is foveated in frame 1. 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).*

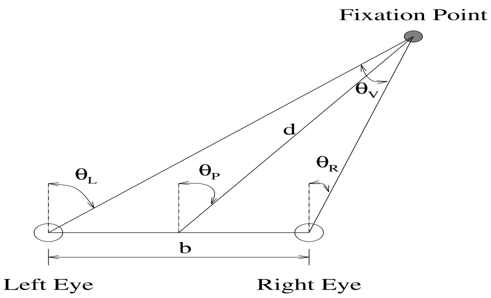

Figure 6: *Gaze angles and range to target geometry.*

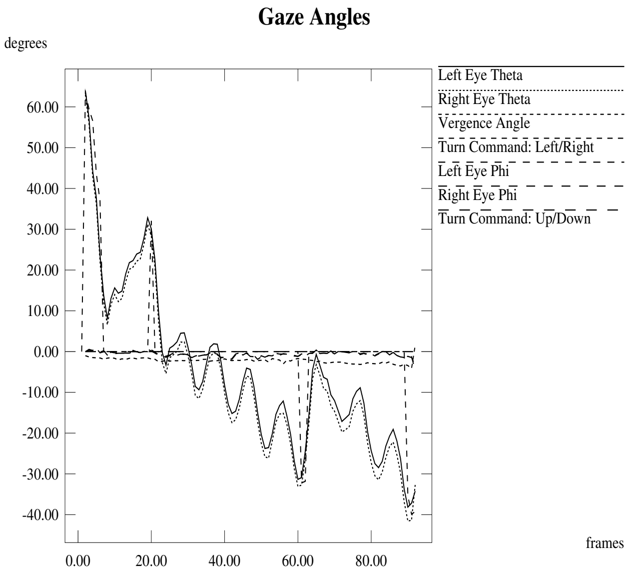

Figure 7: *Gaze angles resulting from the pursuit of a target by the AV fish.*

gaze angles and the turn angles obtained over the course of 100 frames in a typical experiment as the predator is fixated upon and actively pursuing a prey target. Fig. 8 shows a sequence of image frames acquired by the fish during its navigation (monochrome versions of 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 performed a saccade to foveate the target (the scale difference of the target after foveation is due to perspective distortion). 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.

The saccade signals that keep the predator's eyes fixated on its prey as both are swimming are reflected by the undulatory responses of the gaze angles in Fig. 7. The figure also shows that the vergence angle increases as the predator approaches its target (near 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 well correlated.

Note that in frames 87–117 of Fig. 8, a yellow fish whose size is similar to the target fish passes behind the target. In this experiment the predator was programmed to be totally disinterested in and not bother to foveate any non-reddish objects. Because of the color difference, the yellowish object does not distract the fish's gaze from its reddish target. This demonstrates the robustness of the color-based fixation algorithm.

4 Learning

The learning center of its brain (see Fig. 2) enables the artificial fish to acquire effective locomotion skills through practice and sensory reinforcement. Our second challenge has been to enhance the algorithms comprising the artificial fish's learn-

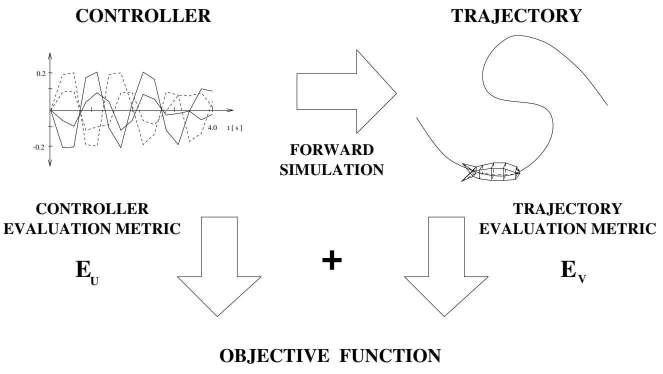

Figure 9: *The objective function that guides the learning process is a weighted sum of terms that evaluate the controller and the trajectory.*

ing center so that it can learn more complex motor skills than those we demonstrated in reference [1].

4.1 Low-Level Motor Learning

Recall that some of the deformable elements in the biomechanical model (Fig. 3) play the role of contractile *muscles* whose natural length decreases under the autonomous control of the motor center of the artificial animal's brain. To dynamically contract a muscle, the brain must supply an *activation function* $a(t)$ to the muscle. This continuous time function has range $[0, 1]$, with 0 corresponding to a fully relaxed muscle and 1 to a fully contracted muscle. Typically, individual muscles form muscle groups, called *actuators*, that are activated in unison. Referring to Fig. 3, the artificial fish has 12 muscles which are grouped pairwise in each segment to form 3 left actuators and 3 right actuators. Each actuator i is activated by a scalar *actuation function* $u_i(t)$, whose range is again normalized to $[0, 1]$, thus translating straightforwardly into activation functions for each muscle in the actuator. Thus, to control the fish's body we must specify the actuation functions $\mathbf{u}(t) = [u_1(t), \ldots, u_i(t), \ldots, u_N(t)]'$, where $N = 6$. The continuous vector-valued function of time $\mathbf{u}(t)$ is called the *controller* and its job is to produce locomotion. Learned controllers may be stored within the artificial animal's motor control center.

A continuous *objective functional* E provides a quantitative measure of the progress of the locomotion learning process. The functional is the weighted sum of a term E_u that evaluates the controller $\mathbf{u}(t)$ and a term E_v that evaluates the motion $\mathbf{v}(t)$ that the controller produces in a time interval $t_0 \leq t \leq t_1$, with smaller values of E indicating better controllers $\mathbf{u}$. Mathematically,

$$E(\mathbf{u}(t)) = \int_{t_0}^{t_1} (\mu_1 E_u(\mathbf{u}(t)) + \mu_2 E_v(\mathbf{v}(t)))\ dt, \qquad (1)$$

where μ_1 and μ_2 are scalar weights. Fig. 9 illustrates this schematically.

It is important to note that the complexity of our models precludes the closed-form evaluation of E. As Fig. 9 indicates, to compute E, the artificial animal must first invoke a controller $\mathbf{u}(t)$ to produce a motion $\mathbf{v}(t)$ with its body (in order to evaluate term E_v). This is done through forward simulation of the biomechanical model over the time interval $t_0 \leq t \leq t_1$ with controller $\mathbf{u}(t)$.

We may want to promote a preference for controllers with certain qualities via the controller evaluation term E_u. For example, we can guide the optimization of E by discouraging large, rapid fluctuations of $\mathbf{u}$, since chaotic actuations are usually energetically inefficient. We encourage lower amplitude, smoother controllers through the function $E_u = (\nu_1 |d\mathbf{u}/dt|^2 + \nu_2 |d^2\mathbf{u}/dt^2|^2)/2$, where the weighting factors ν_1 and ν_2 penalize actuation amplitudes and actuation variation, respectively. The distinction between good and bad controllers also depends on the goals that the animal must accomplish. In our learning experiments we used trajectory criteria E_v such as the final distance to the goal, the deviation from a desired speed, etc. These and other criteria will be discussed shortly in conjunction with specific experiments.

The low level motor learning problem optimizes the objective functional (1). This cannot be done analytically. We convert the continuous optimization problem to an algebraic parameter optimization problem [11] by parameterizing the controller through discretization using basis functions. Mathematically, we express $u_i(t) = \sum_{j=1}^{M} u_i^j B^j(t)$, where the u_i^j are scalar parameters and the $B^j(t)$, $1 \leq j \leq M$ are (vector-valued) temporal basis functions. The simplest case is when the u_i^j are evenly distributed in the time interval and the $B^j(t)$ are tent functions centered on the nodes with support extending to nearest neighbor nodes, so that $\mathbf{u}(t)$ is the linear interpolation of the nodal variables.

Since $\mathbf{u}(t)$ has N basis functions, the discretized controller is represented using NM parameters. Substituting the above equation into the continuous objective functional (1), we approximate it by the discrete *objective function* $E([u_1^1, \ldots, u_N^M]')$. Learning low level motor control amounts to using an optimization algorithm to iteratively update the parameters so as to optimize the discrete objective function and produce increasingly better locomotion.

We use the simulated annealing method to optimize the objective function [12]. Simulated annealing has three features that make it particularly suitable for our application. First, it is applicable to problems with a large number of variables yielding search spaces large enough to make exhaustive search prohibitive. Second, it does not require gradient information about the objective function. Analytic gradients are not directly attainable in our situation since evaluating E requires a forward dynamic simulation. Third, it avoids getting trapped in local suboptima of E. In fact, given a sufficiently slow annealing schedule, it will find a global optimum of the objective functional. Robustness against local suboptima can be important in obtaining muscle control functions that produce realistic motion.

In summary, the motor learning algorithms discover muscle controllers that produce efficient locomotion through optimization. Muscle contractions that produce forward movements are "remembered". These partial successes then form the basis for the fish's subsequent improvement in its swimming technique. Their brain's learning center also enable these artificial animals to train themselves to accomplish higher level sensorimotor tasks, such as maneuvering to reach a visible target (see [1] for the details).

4.2 Learning Complex Skills

Abstracting Controllers It is time consuming to learn a good solution for a low level controller because of the high dimensionality of the problem (large NM), the lack of gradient information to accelerate the optimization of the objective functional, and the presence of suboptimal traps that must be avoided. For tractability, the learning procedure must be able to abstract compact higher level controllers from the low level controllers that have been learned, retain the abstracted controllers, and apply them to future locomotion tasks.

The process of abstraction takes the form of a dimensionality reducing change of representation. More specifically, it seeks to compress the many parameters of the discrete controllers to a compact form in terms of a handful of basis functions. Natural, steady-state locomotion patterns tend to be quasi-periodic and they can be abstracted very effectively without substantial loss. A natural approach to abstracting low-level motor controllers is to apply the fast Fourier transform (FFT) [12] to the parameters of the controller and then suppress the below-threshold amplitudes.

Typically, our artificial animals are put through a "basic training" regimen of primitive motor tasks that it must learn, such as locomoting at different speeds and executing turns of different radii. They learn effective low level controllers for each task and retain compact representations of these controllers through controller abstraction. The animals subsequently put the abstractions that they have learned into practice to accomplish higher level tasks, such as target tracking or leaping through the air. To this end, abstracted controllers are concatenated in sequence, with each controller slightly overlapping the next. To eliminate discontinuities, temporally adjacent controllers are smoothly blended together by linearly fading and summing them over a small, fixed region of overlap, approximately 5% of each controller (Fig. 10).

Composing Macro Controllers Next the learning process discovers composite abstracted controllers that can accomplish complex locomotion tasks. Consider the spectacular stunts performed by marine mammals that elicit applause at theme parks like "SeaWorld". We can treat a leap through the air as a complex task that can be achieved using simpler tasks; e.g., diving deep beneath a suitable leap point, surfacing vigorously to gain momentum, maintaining balance during the ballistic flight through the air, and splashing down dramatically with a belly flop.

We have developed an automatic learning technique that constructs a macro jump controller of this sort as an optimized sequence of basic abstracted controllers. The optimization process is, in principle, similar to the one in low level learning. It uses simulated annealing for optimization, but rather than optimizing over nodal parameters or frequency parameters, it optimizes over the selection, ordering, and duration of abstracted controllers. Thus the artificial animal applying this method learns effective macro controllers of the type shown at the bottom of Fig. 10 by optimizing over a learned repertoire of basic abstracted controllers illustrated at the top of the figure.

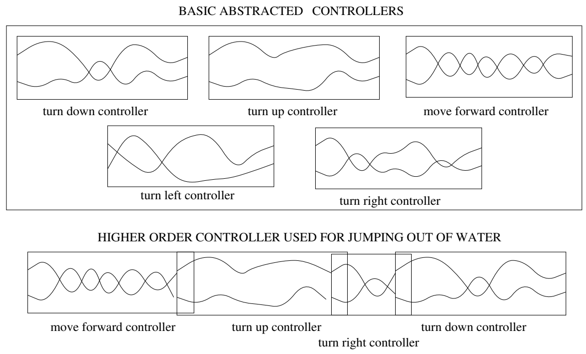

Figure 10: *Higher level controller for jumping out of water is constructed from a set of abstracted basic controllers.*

We have trained an artificial dolphin to learn effective controllers for 5 basic motor tasks: turn-down, turn-up, turn-left, turn-right, and move-forward. We then give it the task of performing a stunt like the one described above and the dolphin discovers a combination of controllers that accomplishes the stunt. In particular, it discovers that it must build up momentum by thrusting from deep in the virtual pool of water up towards the surface and it must exploit this momentum to leap out of the water. Fig. 11(a) shows a frame as the dolphin exits the water. The dolphin can also learn to perform tricks while in the air. Fig. 11(b) shows it using its nose to bounce a large beach-ball off a support. The dolphin can learn to control the angular momentum of its body while exiting the water and while in ballistic flight so that it can perform aerial spins and somersaults. Fig. 11(c) shows it in the midst of a somersault in which it has just bounced the ball with its tail instead of its nose. Fig. 11(d) shows the dolphin right after splashdown. In this instance it has made a dramatic bellyflop splash.

5 Conclusion

We have demonstrated that the artificial fishes model that we developed in our prior work may be effectively employed to devise sophisticated algorithms for perception and learning. We have successfully implemented within the framework of the artificial fish 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 that these vision algorithms confront synthetic retinal images that are by no means easy to analyze (compared to the sorts of images encountered in physical robotics). We have also demonstrated enhanced learning

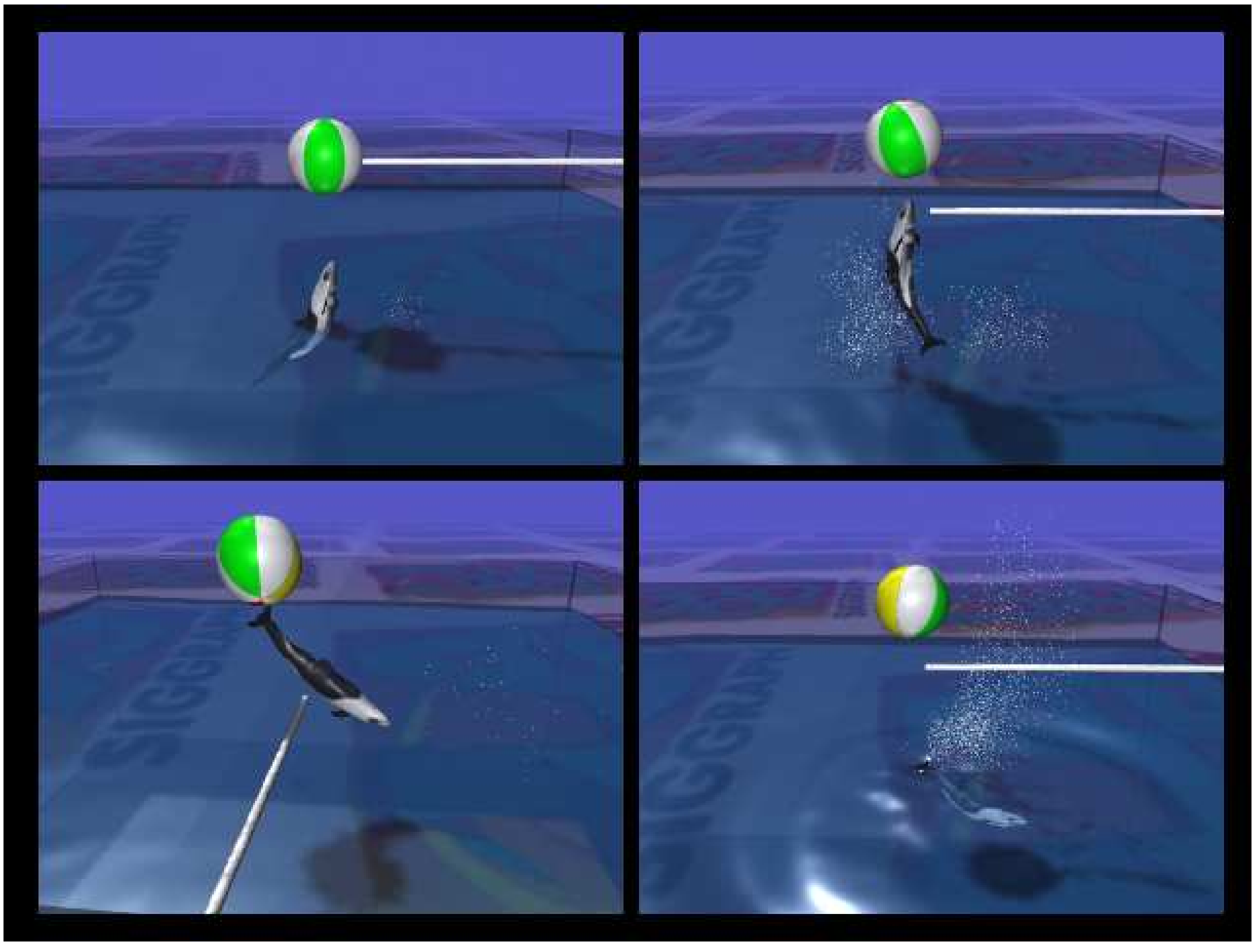

Figure 11: *"SeaWorld" skills learned by an artificial dolphin.*

algorithms that can enable an artificial marine mammal to acquire complex motor skills. These skills necessitate locomotion through water, ballistic flight through air, and a graceful style of execution. The use of highly realistic, physics-based virtual worlds inhabited by biomimetic autonomous agents appears to be a fruitful strategy for exploring difficult open problems in biological information processing and control. Our approach has value beyond artificial life to related disciplines such as vision, robotics, and virtual reality.

Acknowledgements

We would like to thank Xiaoyuan Tu for making the research described herein possible by developing and implementing the artificial fish model. This work was supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

[1] 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.

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

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

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

[5] D. Cliff and S. Bullock. Adding "foveal vision" to Wilson's animat. *Adaptive Behavior*, 2(1):49–72, 1993.

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

[7] R. D. Fernald. Vision. In D. H. Evans, editor, *The Physiology of Fishes*, chapter 6, pages 161–189. CRC Press, Boca Raton, FL, 1993.

[8] M. E. Goldberg, H. M. Eggers, and P. Gouras. The ocular motor system. In E. R. Kandel, J. H. Schwartz, and T. M. Jessel, editors, *Principles of Neural Science*, pages 660–678. Elsevier, New York, NY, 1991.

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

[10] D. Terzopoulos and T.F. Rabie. Animat vision. In *Proc. Fifth International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press.

[11] C. J. Goh and K. L. Teo. Control parameterization: A unified approach to optimal control problems with general constraints. *Automatica*, 24:3–18, 1988.

[12] W. H. Press, B. Flannery, et al. *Numerical Recipes: The Art of Scientific Computing, Second Edition*. Cambridge University Press, 1992.