Presented to Eurographics UK 17th Annual Conference (Eurographics UK '99)
Fitzwilliam College, Cambridge, UK, 13-15 April, 1999 ISBN: 0-9521097-8-6

Ball Catching: An Example of Psychologically-based Behavioural Animation

M F P Gillies and N A Dodgson
Cambridge University Computer Laboratory

February 3, 1999

Abstract

Behavioural animation aims to simulate the behaviour of a character so as to automatically produce sections of animation. An important aspect is simulation of perceptuo-motor behaviour: the ways in which a virtual actor senses and reacts to its virtual environment. The fact that all of the information about the virtual environment is available to the computer makes the problem easier than the equivalent problem in robotics but, to balance this, there is a greater requirement to produce realistic human-like behaviour. These two features combine to make this a novel and interesting area of study.

This paper presents a new approach: in which results from experimental psychology are used as a basis for these algorithms. This provides us with guiding principles for the design of perceptuo-motor routines at a higher level of abstraction than the computer vision methods while maintaining a strong basis in real vision.

As a demonstration of this approach: we describe a simulation of how a cricket or baseball fielder runs to catch a ball. This is a real world problem that requires very specific perceptuo-motor skills and a tight coupling between vision and action. There is extensive work on this area in the experimental psychology literature. Using psychological theories produces a simple, efficient computer algorithm which produces a realistic running path. This example supports the thesis that results from experimental psychology provide a sound basis for behavioural animation.

1 Introduction

Behavioural animation and virtual actors have attracted a lot of interest as an aspect of computer animation. A virtual actor is an entity representing an animated character. These actors can contain representations of various non-visual aspects of a character. In particular, behavioural animation gives autonomous, computer controlled behaviour patterns to an actor. This means that, given sufficiently sophisticated behaviours an animation can be produced purely by interaction between actors and the environment and also between individual actors. This has achieved excellent results with simple animal behaviour, for example models of flocking birds [10] and full fish behaviour [15]. Most behaviour required of actors involves reacting to their environment, or at least taking the environment into account. For this to happen there needs to be some means by which the actor learns about the environment. This means that simulating perception is an integral part of simulating behaviour. All work on virtual actors has addressed this problem to some degree but it has been of particular interest to Renault, Noser, Magnenat-Thalmann and Thalmann [9, 8], Blumberg and Galyean [1] and Terzopoulos and Rabie [11].

The next section summarises previous work in this area and presents a methodology based on using results from empirical, visual psychology to produce algorithms for behavioural animation. Sections 3 and 4 describe an example of this methodology, simulating how an outfielder runs to catch a ball. If a batsman in cricket hits a ball upwards it will move in a roughly parabolic path and the fielder has to run so as to end up at the point at which the ball can be caught. This is a problem which has not been significantly addressed in the computer animation literature but has been studied extensively studied by experimental psychologists. Section 3 presents the current range of theories expressed in the psychology literature. Section 4 discusses how these can be used to produce an algorithm for behavioural animation.

2 Simulated Vision

For a virtual actor to perform realistic autonomous behaviour there needs to be some way in which it gains information about the environment. Real people and animals do this through sensory perception and robots do this through computer vision or other sensors. In the case of virtual actors, however, the problem seems much easier. The actor is contained in a virtual environment which is represented in its entirety within a computer. This means that all the information about the environment is already available to the computer. This unusual situation simplifies the problem of producing perception driven behaviour, but it does produce one complication. For virtual actors it is important to use algorithms which produce realistic behaviour rather than just solving a problem. Giving too much information to an actor can produce unrealistic behaviour, a very simple example is an actor that reacts to events occurring behind its back. This section summarises the ways this problem has been addressed in the past. Work in the behavioural animation literature can be divided into two approaches. One uses ad hoc solutions for individual behaviours. The other uses computer vision techniques to simulate the low level vision processes. In section 2.3 presents a methodology to unify the two approaches while maintaining the advantages of both.

2.1 Ad hoc methods

The first attempts at perception driven behaviours used ad hoc methods which seemed convincing but were mostly geared solely to producing realistic perception in one particular behaviour pattern. In his work on flocking behaviour Reynolds [10] attempted to provide the same information to his virtual birds (called boids) that a real bird would gain from sensory perception without directly simulating that sensory perception. He divides a boid's perception of the environment into two aspects which use different methods. Perception of other boids is used by the flocking behaviour and consists of merely returning the position and velocity of all the boids within a certain radius of the boid. Perception of environmental obstacles for collision avoidance is provided by an extra database of objects in the environment which uses simplified shapes. This is directly interrogated by the obstacle avoidance behaviour. McKenna *et al* [6] produced an animated cockroach which detected a "grabbing hand" and other objects in the environment by interrogating input devices and the graphical database. Tu and Terzopoulos [15] produced animated models of fish. They detect any object that is not fully occluded by another object and that is within a certain visibility radius and visibility angle.

These methods can produce good results efficiently with simple techniques. They can also be well adapted to individual behaviours, producing a specialised, optimised algorithm in each case. However, their originators tend to express worries about their realism. Reynolds thinks his boids would have more realistic behaviour if they could actually see their environment rather than merely being given information about the position of objects. Tu and Terzopoulos suggest that more accurate behaviour could be produced by using computer vision techniques to simulate real vision and, in fact, their work was later extended in this way [11]. The major problem is that though the database can be interrogated directly there still needs to be some sort of algorithm for using the data and for filtering it. Without a guiding principle these algorithms can become rather arbitrary.

2.2 Computer Vision Based Methods

These methods are characterised by rendering the scene from the point of view of the actor and then using computer vision techniques on the resulting images. This approach is often motivated by considerations beyond the needs of merely animating actors in virtual environments. Blumberg and Gaylean [1] use this method in an augmented reality system (the ALIVE system) where an actor needs to be able to detect real objects as well as virtual ones. Terzopoulos and Rabie [11] augmented the original artificial fishes with a sophisticated active, binocular vision system. This was aimed at testing computer vision techniques in a virtual environment rather than producing animations. Renault, Noser *et al* [9, 8] use computer vision style techniques purely for virtual environments. Their system is a sophisticated hybrid technique which augments the information from a rendered image with further information from the graphical database. Instead of simply rendering an image they render two buffers, one is a z-buffer, which makes available the distance of a pixel from the actor and the other contains an object id for each pixel, i.e. it gives the object which is visible at each pixel position. The position of an object can easily be extracted from the z-buffer. This technique can produce good results. The overhead of having to render a scene can normally be reduced by taking advantage of hardware rendering in graphics workstations. It can also take advantage of vision algorithms and behaviour patterns developed in robotics, and are also theoretically closer to how human and animal vision works. It does have a major disadvantage, however. Computer vision is at the moment still very primitive compared to human vision. This means that these techniques are unlikely to be able to produce higher level, more sophisticated behaviours. Renault, Noser *et al* do try to solve this problem by giving the visual system more information than is contained in an image.

2.3 A Psychologically-based Methodology

All the methods described above attempt to simulate the visual processes which humans and animals use to interact with their environments. The computer vision approaches do this at the lowest level but the others do so at different levels of abstraction. For example, Reynolds' boids assume that object detection has already been done and that the positions of objects have been estimated. Renualt's combination of a z-buffer and a buffer of object identifiers assumes that object identification and distance estimation have been done. Visual psychologists have studied and understood human vision at various levels of abstraction, though not necessarily how one level of abstraction is obtained from another. It is also known that different behaviours are performed at different levels of abstraction. We propose that it is possible to use knowledge from visual psychology to produce a behaviour at an appropriate level of abstraction. For example, a low level behaviour like obstacle avoidance might use a low level visual model while classification of different types of tree might use a very high level model. This allows a lot of flexibility in choosing the right method for a given behaviour. The reference to psychology will help achieve realism and the use of higher level models of human vision will help reduce the problems of simulating very low level vision and using imperfect computer vision algorithms.

This approach has a number of advantages. First, it provides a sound basis for designing visual routines. Ad-hoc methods can lead to rather arbitrary algorithms whereas the computer vision style solutions can be over complicated. This basis can allow us to build simple but empirically justified behaviours. Another advantage is that in a virtual environment the psychological ideas can often have very simple implementations as the sources of information can be calculated quite simply from known information in the environment.

3 Psychological Background

There are two main theories of how fielders run to catch balls. Either they predict the path of the ball and run directly to the right place or they run based on instantaneous visual cues and constantly update their velocity so as to arrive at the right place at the right time without actually knowing where it will land. The second theory claims that there is a direct connection between the fielder's image of the ball and the fielder's motion which results in the correct path. This paper describes two algorithms based on schemes which fall into this latter category. Chapman [2] produced an early theory based on keeping the rate of increase of the angle of elevation of the ball from the fielder constant (ψ in Figure 1). There have been several variations on this idea [14, 4, 7]. Todd [12] gives an overview of the visual information available to a fielder about the path of the ball. He also suggests and tests a number of strategies, one of which he found to be compatible with the empirical data. This strategy only works when the fielder is in the plane of motion of the ball, however. It is also unclear whether all the required information needed is available to the fielder. Finally, McBeath, Shaffer and Kaiser [5] suggest a new strategy called the Linear Optical Trajectory (LOT). The two algorithms described below are based on a variation of Chapman's strategy and the Linear Optical Trajectory strategy.

3.1 Chapman's Strategy and Variants

In his 1968 paper Chapman [2] suggests a strategy for running to catch a ball based on the angle of elevation of the ball from the fielder (ψ in Figure 1).

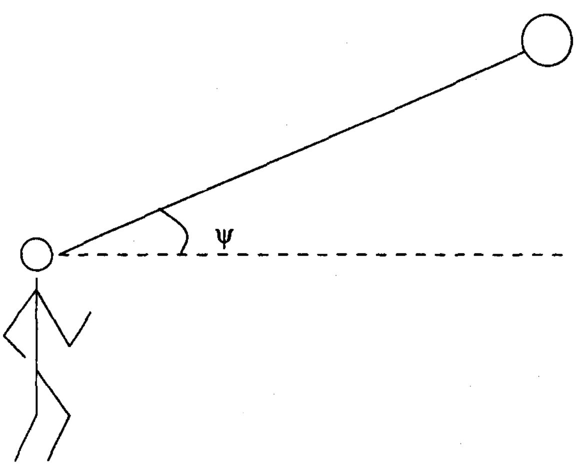

Figure 1: The angle of elevation of the ball

ψ is proportional to the optical projection of the height of the ball, which is the projection onto the position of the eye assuming that the eye is horizontal rather than taking the rotation of the eye into account. This means that the entire analysis can equivalently be applied to the height of the image of the ball at the fielder's eye . Chapman assumes that the fielder is standing in the plane of motion of the ball and shows algebraically that if the fielder is standing at the point at which the ball will land $tan(\psi)$ will be:

$$tan(\psi) = \frac{gt}{2V cos(\theta)}$$

Where V is the speed of projection of the ball, θ is the angle of projection of the ball to the horizontal and g is the acceleration due to gravity. This means that if the fielder is not moving and the first derivative of $tan(\psi)$ is:

$$\frac{d(tan(\psi))}{dt} = \frac{g}{2Vcos(\theta)}$$

then he or she is in the correct position to catch the ball. However, this is only the case in the special situation that the fielder starts off in exactly the right position to catch the ball. Chapman goes on to show that if the fielder runs in the right direction to catch the ball with a speed v such that he or she arrives at the interception point at exactly the right time to catch the ball then $tan(\psi)$ will be equal to:

$$tan(\psi) = \frac{gt}{2Vcos(\theta) - v}$$

Thus the fielder will catch the ball if he or she runs at the correct speed such that $tan(\psi)$ increases at a constant rate given by:

$$\frac{d(tan(\psi))}{dt} = \frac{g}{2Vcos(\theta) - v}$$

This gives us a feasible strategy which *could* be used by cricketers (or baseballers) to catch balls. However, it does not actually prove that it is the strategy used by real fielders. Dienes and McLeod [4, 7] point out that fielders do not run at a constant velocity as required by Chapman's analysis. They generalise Chapman's findings by showing that the fielder will catch the ball if he or she runs so that:

$$\frac{d(tan(\psi))}{dt} = c$$

for any constant c. This no longer requires that the fielder runs at constant velocity, nor does it require the fielder to judge exactly the correct rate of increase of $tan(\psi)$.

So far the analyses I have described assume that the fielder is in the plane of motion of the ball. This makes the problem a one dimensional one and thus it is mathematically simpler. Chapman [2] and McBeath, Shaffer and Kaiser [5] point out that this is in fact not only an unusual special case but that it is in fact more difficult for the player to catch the ball, as there is less visual information available. It is possible that fielders use the same method in 2D but Tresilian points out that this often requires the fielder to have an unrealistically high acceleration. Chapman suggests a generalisation to 2D by which the fielder runs so as to both keep a constant rate of increase of $tan(\psi)$ and a constant horizontal bearing for the ball. Tresilian [14], however, found that this strategy results in the fielder moving in the wrong direction. He proposes another augmentation to Chapman's method, in addition to the acceleration produced by the 1D case there is a second acceleration proportional to the rate of change of direction of the ball relative to the fielder. Thus if α is the visual angle between the ball and an arbitrary reference direction then the magnitude of the acceleration is:

$$a_2 = c\frac{d\alpha}{dt}$$

This acceleration is at an angle β to the direction of the ball such that

$$\beta = arctan(\frac{a_2}{a_1})$$

where a_1 is the magnitude of the 1D acceleration from Champan's method and a_2 is the new acceleration.

3.2 The Linear Optical Trajectory Strategy

In their 1995 paper McBeath, Shaffer and Kaiser [5] describe a different method for catching a ball in two dimensions. They point out that catching in two dimensions is easier than in one dimension so it would seem natural that the best way of describing how people catch is to use an innately two dimensional method rather than trying to generalise a one dimensional method.

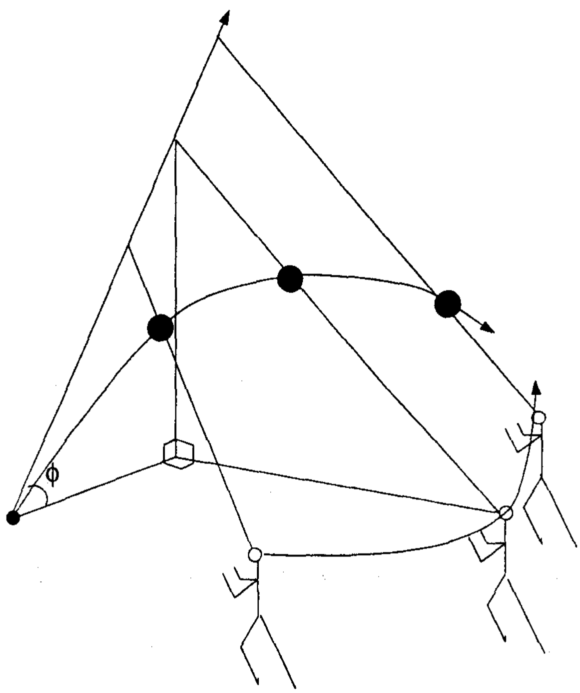

Figure 2: The Linear Optical Trajectory Strategy

They describe the Linear Optical Trajectory (LOT) strategy which states that fielders run so that their image of the ball travels upwards in a straight line. Figure 2 illustrates this, the theory is that if the angle ϕ is kept constant the ball would appear to be moving higher and higher. What is actually happening is that in the second part of the path the ball is moving downwards but as the fielder is moving towards it the ball is coming overhead and so is higher in the visual field. The ball can only keep rising like this if the fielder is moving so that he or she is underneath the ball as it completes its fall, i.e. in the correct place to catch a ball. This theory is supported empirically by an experiment in which amateur baseball fielders wore head mounted cameras and attempted various catches. In most cases the optical path of the ball was roughly linear and so it seems feasible that people use this strategy. They also observed that the paths taken by the fielders were roughly convex as predicted by the LOT theory.

4 Behavioural Algorithms

4.1 Strategies based on Chapman's ideas

Dienes and Mcleod's variation on Chapman's method suggests a very simple algorithm for an actor. The position of the actor and the ball can easily be extracted from the model of the environment and then tan ψ can easily be calculated as:

$$tan(\psi) = \frac{y_b - y_f}{d}$$

where y_b and y_f are the heights of the ball and the fielder and d is the horizontal distance between the ball and the fielder. The actor can then try maintain the value of $d(tan\psi)/dt$ constant at its initial value by having an acceleration opposing the rate of change of this value in the direction of the ball:

$$a_1 = -\frac{d(tan\psi)}{dt}$$

This simple, fast algorithm allows an actor to run and catch a ball. However, as Tresilian predicted it produces unrealistically large accelerations. To avoid this it is necessary to add Tresilian's second acceleration:

$$a_2 = \frac{1}{c}\frac{d\alpha}{dt}$$

Where α is the angle between the fielder's local x-axis (an arbitrary direction) and the direction of the ball, this can also be easily calculated from the positions of the ball and actor. c is a scaling constant (a value of 100 seemed to work best). The acceleration was at an angle β to the direction of the ball where:

$$\beta = arctan(\frac{a_2}{a_1})$$

This algorithm has been designed fairly directly from the theories described in section 3.1. It is computationally simple and effective. The fielder catches the ball fairly reliably though there is a tendency to oscillate when directly underneath the ball. This is not necessarily as great a problem as it seems as there is evidence that fielders use a different strategy in the last stages of a catch, in particular McLeod and Dienes [7] show that their strategy breaks down at the last moment. The main features of the path taken are that the fielder tends to run towards the ball, taking a concave path which moves first towards the early positions of the ball and then towards the final one. The fielder also tends to catch up with the ball and then follow it to its final position.

4.2 The Linear Optical Trajectory Strategy

McBeath, Shaffer and Kaiser's paper [5] did not suggest a strategy for maintaining a linear optical trajectory and so it was more of a challenge to implement. Simply having a linear optical path for the ball is not enough. A linear optical path for the ball throughout its flight is not possible unless the fielder is in the plane of motion of the ball, otherwise the path will bend as the ball passes below eye level. If the fielder is in the plane of motion of the ball then any movement by the fielder in the plane of the ball will result in a linear path. What is needed is an optical image of the ball which moves upwards in a straight line and a fielder motion which results in the fielder being in the right place at the right time to catch the ball. Taking x_f and z_f to be the coordinates of the fielder (throughout this section I will take y to mean up and assume that the fielder only moves in the x-z plane) and x_b, y_b and z_b to be the coordinates of the ball, i.e.:

$$\begin{aligned} x_b &= Vtcos(\theta)cos(\alpha) \\ y_b &= Vtsin(\theta) - \tfrac{1}{2}gt^2 \\ z_b &= Vtcos(\theta)sin(\alpha) \end{aligned}$$

The path of $(x_b - x_f, y_b, z_b - z_f)$ should be a straight line. Decomposing equation into two equations for x and z we get:

$$\begin{aligned} y_b &= m_x(x_b - x_f) + c_x \\ y_b &= m_z(z_b - z_f) + c_z \end{aligned} \qquad (1)$$

for some m_x, m_z, c_x and c_z. We take the initial position for the ball and fielder to be (x_{b0}, y_{b0}, z_{b0}) and $(x_{f0}, 0, z_{f0})$. We also have the requirement that the fielder is in the same place as the ball when the catch is made. We can express this as:

$$\begin{aligned} x_b &= x_f \\ y_b &= h \\ z_b &= z_f \end{aligned}$$

where h is the height at which the ball is caught, this can be thought of as the eye-height of the fielder. Given these constraints equations 1 become:

$$\begin{aligned} y_b &= \frac{h-y_0}{x_{f0}-x_0}(x_b - x_f) + h \\ y_b &= \frac{h-y_0}{z_{f0}-z_0}(z_b - z_f) + h \end{aligned} \quad (2)$$

What is actually needed is a set of equations which relates optic variable available to the fielder to changes in the fielder's velocity. Rearranging and differentiating 2 gives:

$$\begin{aligned} \frac{d(x_b - x_f)}{dt} &= \frac{x_{f0}-x_{b0}}{h-y_0}\frac{dy}{dt} \\ \frac{d(z_b - z_f)}{dt} &= \frac{z_{f0}-z_{b0}}{h-y_0}\frac{dy}{dt} \end{aligned} \quad (3)$$

$d(x_b - x_f)/dt$, $d(z_b - z_f)/dt$ and dy/dt are the components of the velocity of the ball relative to the fielder. These are available to the fielder as the human visual system is sensitive to velocities. If we take h to be the eye height of the fielder $x_{f0} - x_{b0}$, $z_{f0} - z_{b0}$ and $h - y_0$ are the initial components of the position of the ball relative to the fielder and so are also known to the fielder. Equations 3 hold if the fielder is travelling along the correct path such that the optical trajectory of the ball is linear. We can use the error in these equations to produces a suitable acceleration which will keep the fielder roughly on the right path:

$$\begin{aligned} a_x &= -\left(\frac{d(x_b - x_f)}{dt} - \frac{x_{f0}-x_{b0}}{h-y_0}\frac{dy}{dt}\right) \\ a_z &= -\left(\frac{d(z_b - z_f)}{dt} - \frac{z_{f0}-z_{b0}}{h-y_0}\frac{dy}{dt}\right) \end{aligned} \quad (4)$$

Equations 4 are the basis of an algorithm for catching a ball. All the quantities are immediately available in the graphical database with differences used to calculate derivatives. This means the accelerations are simple and quick to calculate (the inner loop has three differences and a multiply).

In most cases the fielder caught the ball, being underneath it when it was at height h. The path taken by the fielder was different to that taken by the fielder using Chapman's method. In this case the fielder took a convex path, first moving away from the ball and only moving towards it near the end of the path. This is consistent with McBeath, Shaffer and Kaiser's experimental findings ??. The path tends to get very convex if h and y_0 are too close as the y term (which is quadratic in t) tends to dominate so my program does not choose y_0 until the ball is clearly higher than h. This is fairly realistic for deep fielder who would not start running until the ball is fairly high. A nearer fielder would need a different strategy for catching shorter balls which do not rise far above h.

5 Further Work

Dannemiller, Babler and Babler [3] argue that the Linear Optical Strategy can be reduced to keeping the angle of elevation constant if the acceleration of horizontal angle was also kept constant. This would seem to suggest another algorithm. It would be interesting to investigate this.

More generally, the methodology of psychologically-based behavioural animation could be applicable to a wide range of problems and we intend to attempt more complex examples.

6 Conclusions

A new methodology for designing algorithms for behavioural animation has been presented and an example of its use has been described. The use of experimental psychology to design behavioural animation has a number of theoretical advantages and it has proved valuable in a simple case. The psychological theories described in this paper lend themselves to implementation by computer and the algorithms they produce are computationally efficient and effective. This all suggests that this could be a useful methodology though it is yet to be seen if it will scale to more complex problems.

References

[1] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *ACM SIGGRAPH*, pages 47–54, 1995.

[2] Seville Chapman. Catching a baseball. *American Journal of Physics*, 36(10):868–870, October 1968.

[3] James L. Dannemiller, Timothy G. Babler, and Brian L. Babler. On catching a flyball. *Science*, 273:256–257, July 1996.

[4] Zoltan Dienes and Peter Mcleod. How to catch a cricket ball. *Perception*, 22:1427–1439, 1993.

[5] Micheal K. McBeath, Dennis M. Shaffer, and Kaiser Mary K. How baseball outfielders determine where to run to catch fly balls. *Science*, 268:569–573, April 1995.

[6] Micheal McKenna, Steve Pieper, and David Zeltzer. Control of a virtual actor: The roach. In *ACM Symposium on Interactive 3D Graphics*, pages 165–173, 1990.

[7] Peter McLeod and Zoltan Dienes. Do fielders know where to catch the ball or only how to get there? *Journal of Experimental Psychology: Human preception and performance*, 22(3):531–543, 1996.

[8] Hansrudi Noser, Olivier Renault, Daniel Thalmann, and Nadia Magnenat-Thalman. Navigation for digital actors based on synthetic vision, memory and learning. *Computers and Graphics*, 19(1):7–19, November 1995.

[9] Olivier Renault, Nadia Magnenat-Thalmann, and Daniel Thalmann. A vision-based approach to behavioral animation. *The journal of visualization and Computer Animation*, 1(2):18–21, 1990.

[10] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *ACM SIGGRAPH*, pages 25–33, 1987.

[11] Demetri Terzopoulos, Tamer Rabie, and Radek Grzezczuk. Perception and learning in artificial animals. In *Artificial Life V: Proc. Fifth International Conference on the Sythesis and Simulation of living Systems*, May 1996.

[12] James T. Todd. Visual information about moving objects. *Journal of Experimental Psychology: Human preception and performance*, 7:795–810, 1981.

[13] James R. Tresilian. Perceptual information for the timing of interceptive action. *Perception*, 19:223–239, 1990.

[14] James R. Tresilian. Study of servo-control strategy for projectile interception. *The quaterly journal of Experimental Psychology*, 48A:688–715, 1995.

[15] Xiaoyuan Tu and Demitri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *ACM SIGGRAPH*, pages 43–49, 1994.