Prey and predator: synthetic senses in a simulated marine environment

Marco Conti*- 09262805
Trinity College Dublin

Abstract

The marine environment is a popular setting for physics simulation, autonomous agent and behavioural animation papers since [Tu and Terzopoulos 1994a]. Virtual fishes can be defined as autonomous agent that interact in real time with a dynamic environment. They are based on a spring-mass physical locomotion model and on a behavioural model based on roles (prey, predator) and sensors.

While extensive work has been done in modeling sophisticated virtual fishes, this paper focuses on the sensor system and reaction to environmental stimuli. This paper comes with a demo illustrating different sensors used by a fish to identify a stalking predator and a simulation of the chase once the predator attacks.

Keywords: autonomous agents, behavioural animation, virtual fish, synthetic vision

1 Introduction

The original *artificial fishes* were first introduced in [Tu and Terzopoulos 1994a] as autonomous agents with a non-trivial interaction and behavior model. This framework has been expanded in subsequently papers and many different specific subtopics have been investigated. In this paper, we want to investigate the result of a combination of a sensor system and an awareness model to mimic the prey-predator behavior in this environment. As suggested in [Stephens et al. 2003], we will take into account the vision, which is relatively limited in an underwater environment, and mechanoperception provided by the lateral line of the fish, that can detect vibration and hence movement and sounds. This last sense can extend its range farther than sight and is used by predators to identify possible prey before actually spotting them.

In our demo we won't focus on an accurate physical simulation of the fish or on the reproduction of different high-level behaviours, but we will implement a simple scenario where a prey fish will actually try to escape an attacking predator. The success of this action will depend on the ability of the prey to detect the predator before it attacks, as if the prey spots the predator from a distance, it could escape. In this demo the initial positions of prey and predator are randomly chosen and the final outcome is not predefined.

The detection of a predator is based on a multi-state model to take into account the awareness level of the prey. This awareness level will influence the future perceptions and actions of the prey, e.g. if the prey detects something unusual in an area, it will not approach that area even if it cannot clearly detect a predator. This awareness model is inspired by [Leonard 2003], and is based on the sum of the sight and mechanoperception senses.

In the next section we will cover the state of the art on virtual fishes simulation in real time and virtual agents percepts. We will then move to describe the design of the demo and the different choices that have been made in the implementation. Finally we will discuss the results achieved.

2 State of the art

One of the first and most influential papers about *artificial fishes* is the aforementioned [Tu and Terzopoulos 1994a]. In this paper, fishes are categorised as prey, predators and pacifists, and a special behavior is defined for each class. Complexity of the behavior model in that paper extends up to modeling mating rituals. The artificial fishes are embodied in a physical model capable of locomotion, thanks to a mass and spring system used to simulate the fish muscles, and movement is obtained in accordance to hydrodynamics laws. Physical simulation of the locomotion has been largely explored both in [Tu and Terzopoulos 1994a] and [Stephens et al. 2003] and a connection between low-level movements and high-level movement behaviours as in [Reynolds 1999] has already been proposed. There is therefore a fair amount of material on physical simulation so there is no need for our paper to investigate this aspect.

[Reynolds 1999] and [Reynolds 1987] can be considered the main references regarding low-level movement behavior. These papers outline a different number of movement behaviours, and inspired the popular *OpenSteer* ([Reynolds 2009]) framework.

Artificial fishes' sensor system have also been proposed and explored in [Tu and Terzopoulos 1994b] and [Stephens et al. 2003], while a scenario closely related to the one proposed in our demo has already been explored in [Funge et al. 1999]. The paper implements a competition between a prey (mermaid) and a predator (shark) in a underwater environment. The mermaid and the shark can take advantage of the sight sense to detect the opponent, but only the mermaid has the ability to come up with a plan to evade the faster shark, by hiding behind rocks.

A game oriented sensor system based on different perceptions and awareness levels, featuring a sight sense based on viewcones, has been employed in the game *Thief: the Dark Project* (Looking Glass Studio), as described in [Leonard 2003]. *Thief: the Dark Project* has been defined as a 3D stealth-based game, where the player has to avoid detection rather than simply killing opponents. A perception system based on graphically represented viewcones has also been made popular in the *Commandos* (Pyro Studios) games series.

*e-mail: contim@tcd.ie

3 Design

3.1 Environment

The environment in which the demo is settled is a simplified cylindrical model of an oceanic seabed. The bottom side of the cylinder is the sand bed, and includes a seaweed forest where predators can hide and stalk preys passing by. The seaweed affects vision, reducing the distance fish can be spotted, while it doesn't impede movement or vision from inside the forest to the outside.

Two types of agents are present in the environment: *prey* and *predator*. The *prey*'s goal is to survive swim around the environment and avoid any contact with the predator; it can achieve this either by avoiding being detected by the predator or by fleeing once it is detected by the predator. On the other hand the *predator* will try to detect the prey and overtake it.

3.2 Sensors

The virtual fishes perception system for the purpose of this paper is based on two senses: sight and mechanoperception. These two senses will generate different stimuli that the autonomous agents can receive. Three levels of stimuli are considered:

no stimuli

some activity (the perception system detects something but the source is unknown)

agent detected (positive identification of another agent)

The sum of the information collected by the perception system is added to the agent knowledge. The agent will modify its behavior according his knowledge of the environment and its AI model.

3.2.1 Sight

The proposed fish model has a cyclopean vision, i.e. the sight sense receives data from a single point positioned in front, with a field of view defined by a spherical angle. While [Tu and Terzopoulos 1994a; Stephens et al. 2003] suggest the use of a spherical angle, in this paper we integrate the concept of *viewcones* defined [Leonard 2003]. Instead of having a single spherical field of view with a given distance, we use different fields of view to model different resolutions in peripheral vision. These viewcones will be modelled as 3D cones centred on a fixed point between the fish's eyes, and the different viewcones will have a different maximum distance of perception. Figure 1 illustrate the different viewcones of the predator.

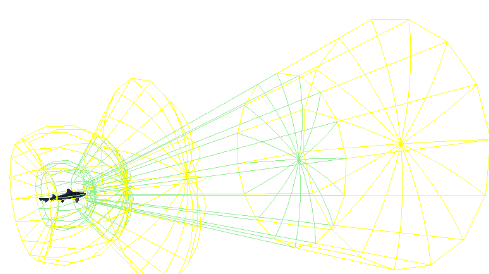

Figure 1: *Different fields of view for the predator. Green and yellow represent different resolutions*

If an object falls inside one of the viewcones of the agents, the object is possibly visible to the agent. The distance the agent can see is affected by the seaweed, and an object inside a seaweed forest might not be spotted even if inside the normal view range. Additionally, since eyes catch moving objects more easily, we also add a function of object velocity in the evaluation of the visibility score ν.

We consider two thresholds τ_1 and τ_2 expressing the fish's reactivity to the sight sense. The stimulus S perceived is then:

$$\mathrm{S}_{sight} = \begin{cases} \text{agent detected} & \text{if } \nu < \tau_1 \\ \text{activity detected} & \text{if } \tau_1 < \nu < \tau_2 \\ \text{no stimuli} & \text{if } \nu > \tau_2 \end{cases} \tag{1}$$

3.2.2 Mechanoperception

The mechanoperception is a sense that responds to mechanical pressure. For an animal immersed in water, this means detecting the vibration propagating in the liquid, in a similar way as surface animals can detect sound (vibrations travelling trough the air). While in most surface animals the hearing sense is mainly located in the ears, fishes can take advantage of the lateral line, an organ situated along both sides of the body which detects vibration carried by the water. The mechanoperception can therefore be roughly described as a underwater hearing sense, capable of detecting fishes swimming or otherwise producing vibrations. This can include, over short distances, detecting the heart or gill muscles contracting.

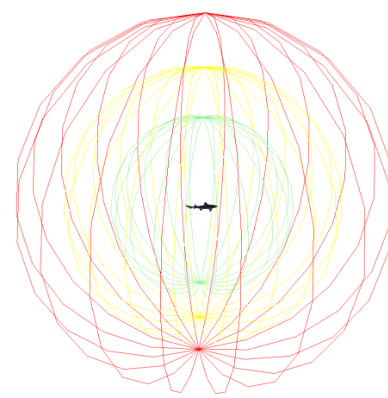

Figure 2: *Mechanoperception spheres. Different colours represent different resolutions*

In the demo we try to apply to this sense the concept of viewcones already considered with sight. While we assume that the lateral lines don't have a directional resolution (so that the detection can be performed in any direction without any penalty) we model different levels of resolution in this sensor system as spheres. We define three radii τ_1, τ_2 and τ_3 centred on the fish's body, as illustrated in figure 2. The detection of another fish will eventually occur according to the distance δ between the sensing agent and the target T of the perception.

$$\mathrm{S}_{mech.} = \begin{cases} \text{agent detected} & \tau_1 < \delta < \tau_2 \text{ and T is moving} \\ \text{agent detected} & \delta < \tau_1 \text{ even if T is not moving} \\ \text{activity detected} & \tau_2 < \delta < \tau_3 \text{ and T is moving} \\ \text{activity detected} & \tau_1 < \delta < \tau_2 \text{ even if T is not moving} \\ \text{no stimuli} & \text{otherwise} \end{cases} \tag{2}$$

3.3 Behaviour

Prey and predator react differently to different awareness levels generated by the sensor system. As already outlined, the possible stimuli perceivable are: *no stimulus*, *activity detected* or *agent detected*. We model the two fishes' AI as finite state machines with the following states: *wandering*, *aware*, *and chasing/fleeing*, illustrated in figure 3. The fourth state, *dead*, is only reachable by the prey when the predator bites it.

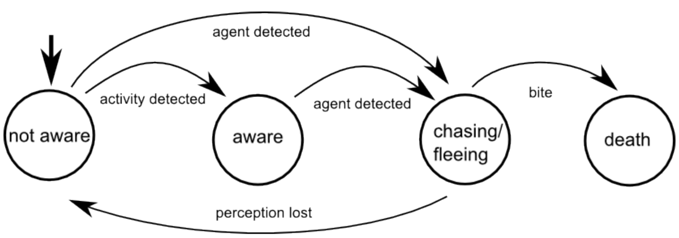

Figure 3: *Finite-state machine of the agents*

An agent will be initialised in the *wandering* status. In this status, the prey will simply swim from one end to another of the environment, and will eventually leave the environment if it reaches the opposite side without encountering the predator. The predator will instead slowly swim through the environment, performing a wandering behavior ([Reynolds 1999]), and if it enters the seaweed forest, it stops there for a random time before resuming the wandering behavior.

If the prey detects some activity, it will move to the *aware state*. The assumption is that the predator is actively looking for another living being to eat, so it will be curious to investigate any activity detected. In case of multiple sources, the closest one will be investigated. On the other hand, the prey is suspicious and afraid of meeting a predator, so will try to avoid any contact with unknown sources of stimuli by slowly swimming away, because it can possibly be a potential danger. Again, in case of multiple sources, the prey will swim away from the closest. It's worth noting that, since the detection is affected by the velocity of the agents, the fishes will always try to swim at the slowest speed with which they feel safe.

In the event of a positive identification, the prey will change to the *chasing/fleeing* state. The prey will simply run away from the source at full speed, aiming for any side of the cube, as it will be considered safe if it exits the environment before the predator catches it. The predator reaction to a positive identification is a bit more complex, as it takes into account the chance of waiting for the prey to get closer. The predator will remain in the *aware* state and stay still pointing at the closes prey until it decides to start the chase.

The predator will sprint and start to chase the prey only if is confident that the prey is close enough to outrun (i.e. closer than a given fixed distance δ) or if the prey is moving away from the predator. If the prey is moving towards the predator, the predator will wait to have a higher chance of success in a second moment. Once the chase has started, the two agents will perform the pursuit and evasion behavior seen in [Reynolds 1999].

We also modelled a sort of intelligence value for the agent. As in real life, similar initial situations can lead to different outcomes, we decided to have the predator *estimating* the distance to the prey instead of knowing the exact value. This decision results in a more realistic predator making some evaluation mistakes once in a while by starting the chase too early and failing to catch the prey.

3.4 Knowledge representation

The stimuli generating from the senses helps to build the agent knowledge base. This knowledge base maintains information on the position of the other agents. According to the last perception received, the information available to the agent has different uncertainty levels:

- If the agent is currently detected, its exact position and velocity is stored
- If some activity is detected, the position is stored, with an added uncertainty radius that represent the vagueness of the stimuli received
- If the agent was detected some moments ago but is now out of reach, his estimated position is updated

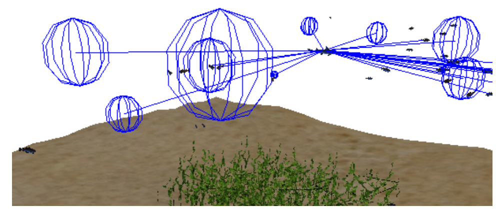

Figure 4: *Knowledge representation of the predator in a particular moment*

The estimation of an agent position, when it can no longer be directly detected, is based on the time passed and the last known velocity. The position has an uncertainty radius that grows as time is passing, and the position is updated according to the last known velocity. The velocity also affects the growing ratio of the uncertainty sphere. A representation of this knowledge base is shown in figure 4. After some time has passed, old information is dropped because the accuracy is too low to be of any use.

3.5 Movement and physical model

The fishes are not being modelled as the spring-mass system designed in [Tu and Terzopoulos 1994a], but as a simple point. The point is centred on an approximation of the agent's 3D model. The two fishes will therefore be very similar to the *boids* described in [Reynolds 1999] and [Reynolds 1987].

We modelled the prey to be faster than the predator, and so even if the predator can take advantage of the initial faster sprint, this will vanish after a few seconds. The predator is then able to catch the prey only if it's smart enough (e.g. hiding in the vegetation). Of all the behavior defined in [Reynolds 1999], the ones implemented in this demo are *wandering*, *seek*, *flee*, *pursuit* and *evade*.

The fight between prey and predator is resolved with a simple collision detection. If the predator comes in contact with the prey, we consider the prey to be eaten.

4 Implementation

The demo is implemented in C# using the XNA framework [Microsoft Corporation 2007], as opposed to the initial plan of using OpenGL. This decision was taken early in the development phase as the XNA environment proved to be more easy to use, allowing for faster development.

The environment in which the agents operates is a 3D simulation of the seabed, where the agents can move freely in all directions inside a cylinder representing the boundaries. The predator and prey are represented with a 3D model of a shark and a fish, available from [Toucan Corporation 2008]; we added some simple skeletal animation to them using Blender 3D. The underwater environment is recreated with simple static models.

While OpenSteer ([Reynolds 2009]) was an appealing framework to implement movement behaviours, we considered the additional effort of including it in our simple demo to be greater than the effort required to implement the trivial behaviours requested.

During the demo some information can be displayed by pressing keys 1 to 7 on the keyboard; these will turn on and off the visualisation of the fishes' senses and knowledge representation, while pressing the 0 key will activate/deactivate the predator's AI. Different initial scenarios, with different numbers of fishes and deactivated AIs, can be shown by pressing the F1 to F5 keys. The last scenario (F5) allows the user to control the prey movement with the XBoX 360 controller.

The simulation is able to run in real time with 100 fishes on a desktop computer and the outcomes are determined by the rules outlined in this paper, resulting in a realistic simple animation of a prey-predator interaction. The only scripted action is the initial setup of the scene.

A video of the demo can be found at: *http://www.youtube.com/watch?v=pb1hKTFWgY0*

4.1 Issues

The demo was initially designed to have only two agents and we later decided to have more than one prey. As result of this, we had less time to polish this scenario and some unrealistic behaviour can be seen in the predator while in the *aware* state. Even if there is nothing wrong with the behaviour itself (constantly changing the focus from one prey to another, according to which one is the closest), the animation resulting is not smooth, as the predator seems to jitter as it faces the current target. This can easily be fixed by including a delay in the rotation.

It's also worth nothing that due to the limited size of the environment, chases can't go on forever. Eventually the agents will reach the border of the environment and turn around to avoid the collision. This will result in a fleeing prey slowing down to turn and therefore the predator will always catch a fleeing prey after some time. However if we remove the world limits, the faster prey will successfully flee by simply increasing the distance from the predator up to the point where the predator is not able to detect the prey anymore.

4.2 Improvements

The demo might use a better wandering steering behaviour, as after a while all the fishes tend to concentrate on the ceiling or the bottom of the environment. Removing the world boundaries but somehow keeping the models visible should improve the behaviour.

Adding physical simulation to the fishes, as proposed in the original papers, will probably also add to the credibility of the simulation, as well as adding more fish and environment models like rock, sunken ships and vegetation. But the main improvement regarding the credibility of the fishes would result from using more realistic sense parameters (orientation, field of view, distance) coming from biological studies and having some sort of collaboration between the prey (e.g. flocking behaviour).

Last but not least, having already implemented a user-controlled prey in one of the scenarios, one can consider the idea of creating a simple stealth game out of this demo.

References

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 29–38.

LEONARD, T. 2003. Building an ai sensory system: Examining the design of thief: The dark project. In *GDC 2003*.

MICROSOFT CORPORATION, 2007. Xna. *http://www.xna.com/*.

PATTERSON, S., 2009. The sneaky history of stealth games. Gamesradar US, *http://www.gamesradar.com/f/the-sneaky-history-of-stealth-games/a-2009020393535662028*, (retrieved 12 Dec 2009).

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 25–34.

REYNOLDS, C. 1999. Steering behaviors for autonomous characters. In *proceedings of Game Developers Conference 1999*, Miller Freeman Game Group, San Francisco, California, 763–782.

REYNOLDS, C., 2009. Opensteer: steering behaviors for autonomous characters. *http://opensteer.sourceforge.net/*.

STEPHENS, K., PHAM, B., AND WARDHANI, A. 2003. Modelling fish behaviour. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, ACM, New York, NY, USA, 71–78.

TOUCAN CORPORATION, 2008. Tuocan virtual museum. *http://www.toucan.co.jp/indexE.html*.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: physics, locomotion, perception, behavior. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 43–50.

TU, X., AND TERZOPOULOS, D. 1994. Perceptual modeling for the behavioral animation of fishes. In *Pacific Graphics '94: Proceedings of the second Pacific conference on Fundamentals of computer graphics*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 185–200.