Predator-Pray Model in Water

Almir Heralic´[1] and Björn Franzon [2]

Complex Adaptive Systems, Chalmers University of Technology

1 Almir Heralic, 811118-5776, almir@etek.chalmers.se
2 Björn Franzon 811107-4871, quack@etek.chalmers.se

Abstract

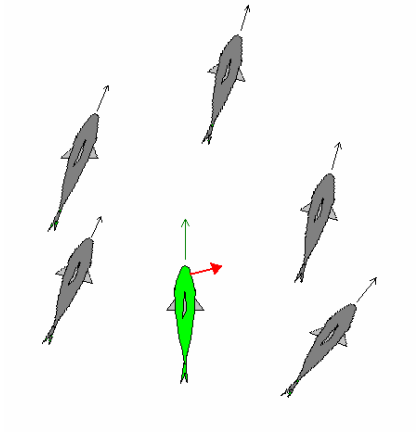
We wanted to create a model of animals acting in the real world and to reproduce some of there behavior in our simulations. The solution we came up with was to simulate a predator-pray model in water where fishes are hunted by simple predators which are just swimming after them and another more advanced predator that can hide in the sand and surprise the fishes with a sudden strike.
An agent based model was used to create the different species behavior. Fishes are obeying 4 simple rules while the predator is following the hunting behaviors. The mechanism that will choose the right behavior for the predator is evolved using Genetic Algorithms. Compared to the simple predator which just chases the fishes our predator was approximately 8 times more efficient.

1. Fish model

All fishes start with a random position, heading and speed and in every move a small random change of the current heading and speed are made. The agent based model for the fishes uses the four rules alignment, separation, attraction and escape. The first three rules are set up to create the flocking behavior and the last one to survive against the predators. All fishes has two sight ranges where one is how far the fish can see its neighbors and one shorter where the fish distinguish between friendly fishes and predators.

Alignment

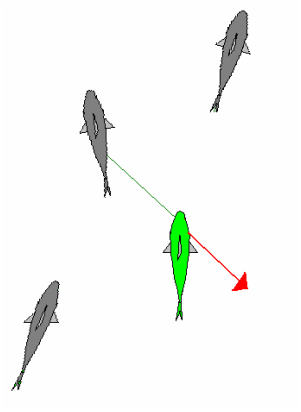
Alignment makes the fish adapt to its surrounding neighbors speed and heading by calculating the average value of those and with a certain percentage change its current velocity towards it. An example of how alignment works can be seen in *Picture 1* below.



Picture 1: Alignment

Separation

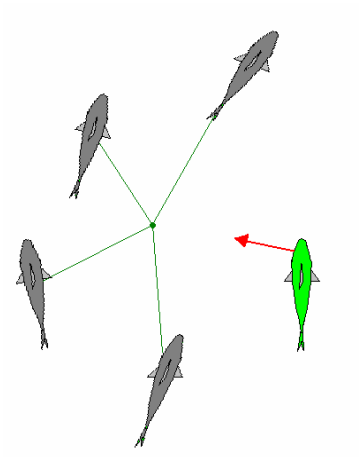
Separation is used to keep the fishes at a distance from each other to avoid collisions. It is also used in combination with alignment and attraction to simulate a comfortable distance as for example humans have in normal speak-conversations. The separation is calculated by the neighboring vision radius divided by the distance to the nearest fish as *Picture 2* shows.



Picture 2: Separation

Attraction

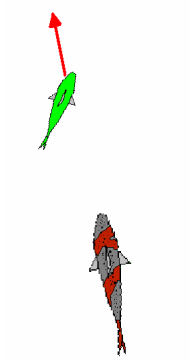
The attraction rule helps the flock to keep together where fishes in the edge get a vector towards the center of the flock, see *Picture 3*. Attraction is given by the average position of neighboring fishes and can there by not have a too large factor of changing towards this direction because fishes in the front of the flock will have an opposite attraction heading.



Picture 3: Attraction

Escape from predator

To simulate the escaping behavior of the fish the radius of finding predators is used and if a predator is spotted this behavior totally takes over. *Picture 4* shows a fish that is trying to escape from a predator. The fish changes to maximum speed and steers away from the closest predator with a given factor. If a fish goes into the escaping behavior it continues escaping until a given time of no enemies seen has passed.



Picture 4: Escape from predator

2. Predator model

The predators start as the fishes, with random initialized position, heading, and speed. Furthermore, in every move a small random change of the current heading and speed is made.
There are two types of predators in the simulation. The first one obeys a very simple predator behavior namely, the simple predator swims around randomly and if it sees a fish in its vision radius it tries to catch the closest one by swimming in maximum speed towards it. If a fish spots a predator it will immediately swim away with its maximum speed, which is however set to exact the same as the predators (see table 1). This will result in predators hardly catching any fishes at all. Still there is a possibility of this simple predator to catch a fish and that is when two predators attack the same flock but from opposite directions. But since predators are not aware of each other and therefore not cooperate to evolve such behavior, this is a rare sight.

Even though these predators hardly ever catches a fish they contribute to a more interesting fish behavior where these predators makes the fishes split up in smaller flocks and thereby create flock escaping phenomena.

Advanced predator model

For this predator, we had another approach. Instead of giving the predator rules to follow and hand code the exact moment when they are to be applied, we evolved a behavior choosing mechanism by means of evolution. This means that the predator has learned to choose the most appropriate behavior in a certain situation so that the catch is maximized. It can choose between:

- hide in the sand
- swim around
- attack

If the predator chooses to hide in the sand, it is of course not seeable by the other fishes and can thereby wait for the right moment to attack.

What kind of input data is the predator getting?
When reading of its environment the predator looks for the shortest distance to a visible fish. Note that the predator has much longer vision radius than the fishes (see table 1). On the other hand it has a vision angle of only 90 degree while the fishes can see all-around.
Apart from the distance the predator is also looking for the direction that the spotted fish is having. Hopefully this will result in a predator choosing only to attach fishes that are swimming towards it rather then away from it, because then the chances of a catch will increase since the fishes cannot turn away instantaneously.

Utility manifold method

In the utility manifold method [1], each behavior B_i is assigned a utility function U_i which depends on the values of the state variables of the agent (the predator), i.e. the distance to the nearest fish and the difference between the predator and that fish's heading. The behaviors are divided into two categories. Task behaviors, which are directly related to the task of the

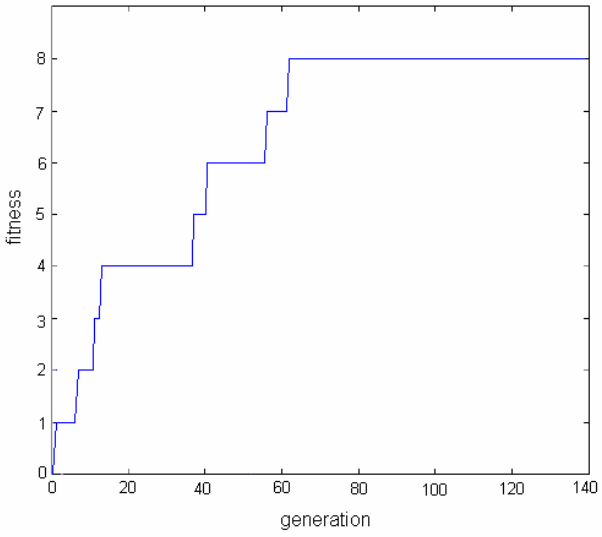
predator, and increase its fitness i.e. attack behavior, and the auxiliary behaviors which do not increase the fitness but are still necessary for the fitness to increase in the longer run.

The task is now to evolve these utility functions so that the fitness, i.e. number of caught fishes, is maximized. We will use a utility function of the form:

$$U_i(d,\alpha) = a_0 + a_1 \cdot d + a_2 \cdot \alpha + a_3 \cdot d^2 + a_4 \cdot d\alpha + a_5 \cdot \alpha^2$$

where d and α are the distance to the nearest fish, and the difference between the predator and that fish's heading.

In order to get the appropriate value of the constants a_i, we have used Genetic algorithms [2] to evolve them. In a GA many potential solutions are encoded in strings, called chromosomes. These are evaluated in a specific task and a fitness measure is assigned. Genetic operators such as crossover are used to combine genes of the best solutions (with highest fitness) to form new solutions, which are inserted into the population. Mutation is another genetic operator that provides new information for evolution to work with. Each iteration of the genetic algorithm consists of evaluation of solutions, crossover and mutation and is called a generation. In our case the chromosome consists of the a_i constants. Since the evaluation time is rather long (few seconds) the population of individuals is chosen to be 50. Selection method used here is steady-state tournament selection, i.e. four different, randomly picked individuals compete against each other in pairs. The two winners get to breed and their offspring replace, after undergoing mutation, the two looser individuals in the population. This way the population is gradually changed towards an optimum predator.



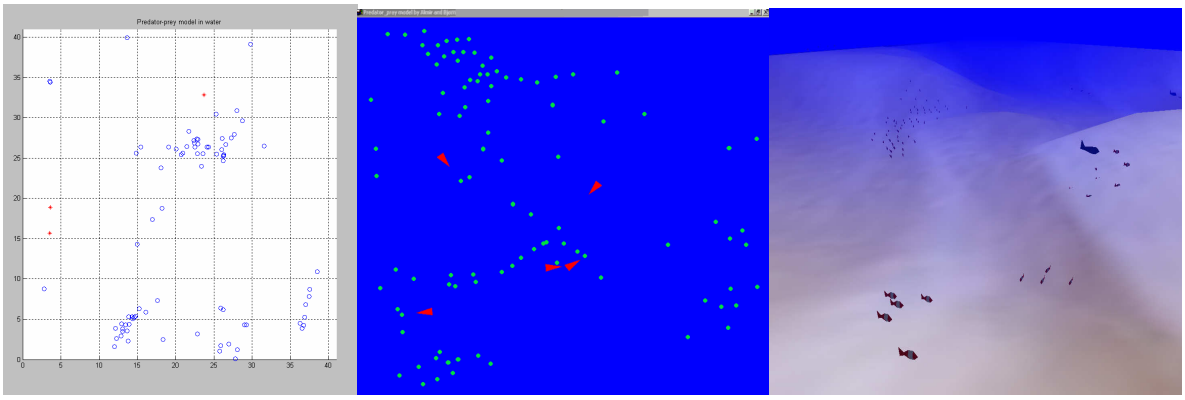
Picture 5: Fitness as a function of evaluated generations

A result of the training run is shown in the *Picture5* above. We see that the fitness is increasing very fast in the beginning and then converges towards a local optimum. The reason of this convergence is partly the short simulation time and partly the fact that the population size is somewhat low. Newer the less this new predator is much more efficient in hunting than the simple predator model described earlier, which for the same simulation time catch on average just one fish. During the simulation, the strength of the evolved hunting behavior was seen each time the predator ambushed the fishes. It was particularly interesting to see predator

sometimes not attacking even if the fishes was quite near it. The reason why the predator did not attack in these situations was of course the heading of the fishes, which in these cases was always away from the predator and not towards it.

3. Implementation

We started out with a 2-D implementation in Matlab and when a satisfactory result of the different species was given, the same simulation was implemented in C combined with OpenGL. After this the code was changed to even work in a 3-D environment as well. The three implementation steps can be seen in *Picture 6* below.



Picture 6: Three implementation steps: Matlab, OpenGL 2D and OpenGL 3D

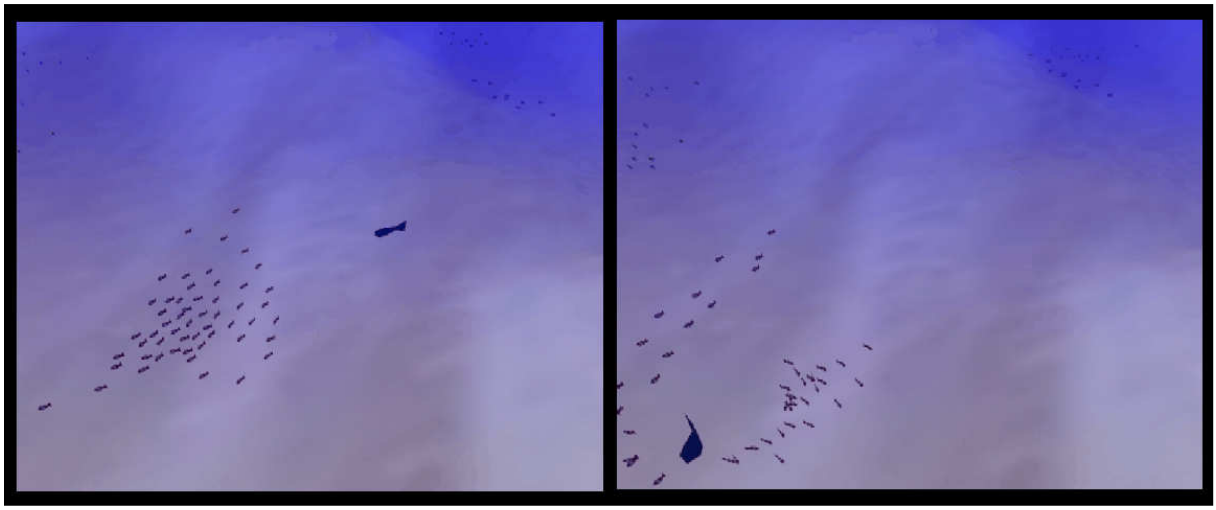
The 3-D model is actually a 2-D model where all individuals stay at a certain distance above the sand. The model uses periodic boundary conditions for the horizontal plane. The 3-D graphics model uses a height map with a sand texture for the ground. The fishes and the predators are created using 26 polygons per individual. Furthermore a blue fog is added for the illusion of water. The population of fishes and predators are constant, meaning that the predators don't die if they don't catch any fishes. On the other side, if a fish gets caught a new one will come up at a random spot.

Different parameter values chosen for this simulation can be seen in table 1 below.

	Fish	Predator
Normal speed	0.5	1.0
Max speed	1.6	1.6
Vision radius	15	20
Threat radius	7	-
Vision angle	360 ^{o}	90 ^{o}
Alignment speed dependence	70%	
Alignment angle dependence	60%	
Separation dependence	3 %	
Attraction dependence	5%	

Table 1 Constants used when the model is updated

The high alignment dependence shows that it is the most important rule to get a flocking behavior even though separation and attraction is necessary. Giving the fishes and predators same max speed forces the predators to surprise the pray in order to actually catch it.
A flock escaping phenomena that occurred was that it looks like the fishes create a road through the flock for the predator to swim at. *Picture 7* shows the predator road phenomena after the attack from the predator (the simple model).



Picture 7 Predator road phenomena

4. Conclusion and discussion

To get a good behavior in the model many combination of constants had to be made. The range where the simulation gives the "true" look is very narrow. For example small changes in Separation and Attraction dependence easily give individual fishes oscillations or on the other hand compact clustering.

The reason of implementing a 3-D world was to actually see how powerful the model was to create a simulation of the real world. For example to be able to see the world from the eyes of a fish or a predator really gave a good impression and as it all turned out we were more than satisfied with the result. Some part of the illusion of a real world could not be seen in the 2-D model and also a lot of new thoughts of improvements were given when the 3-D model were finished. Since the fishes are only moving in two directions there is more work to do on this model but the current code is such that it should not be a major challenge.

As for the predators we are satisfied with the performance of the evolved one but will in future work try to involve the others to evolve cooperative behaviors. Furthermore the real predator-prey model is to be implemented meaning that the population size will vary depending on e.g. how good the predators are as hunters. However, we think that our evolved predator could be successful in that model since it conserves a lot of energy while lying still in the sand.

Another very interesting case to analyze would be to find the steady states of this model. What influence dose different flocking behavior have on the fishes chance to survive? An intuitive answer to this question would be that predators have difficulties finding fishes when they cluster up in flocks instead of doing random walks.

References

[1] Holland, J. 1975: Adaptation in natural and Artificial Systems. Ann Arbor, MI, USA: The University of Michigan Press.

[2] Wahde, M. A method for behavioural organization for autonomous robots based on evolutionary optimization of utility functions, J. Systems and Control Eng., 217, pp. 249-258, 2003