Fuzzifying the Thoughts of Animats

Iztok Lebar Bajec, Nikolaj Zimic, and Miha Mraz

Faculty of Computer and Information Science, University of Ljubljana,
Ljubljana SI-1000, Slovenia,
iztok.bajec@fri.uni-lj.si,
http://lrss.fri.uni-lj.si

Abstract. In this article we present a fuzzy logic based method for the construction of thoughts of artificial animals (animats). Due to the substantial increase of the processing power of personal computers in the last decade there was a notable progress in the field of animat construction and simulation. Regardless of the achieved results, the coding of the animat's behaviour is very inaccurate and can, to someone not familiar with common physics variables like speed, acceleration, banking, etc., seem like pure black magic. Our leading hypothesis is, that by using linguistic programming based on common sense, unclear and even partially contradictory knowledge of dynamics, we can achieve comparable, if not better, simulation results. We begin the article with the basics of animats, continue with their fuzzyfication and end with the presentation and comparison of simulation results.

1 Introduction to Animats

The research field of the construction of simple artificial life was started by J. von Neumann in the mid 20th century, when he first set ground for the basics of cellular automata. The main field of his research was the modelling of the basic characteristics of living organisms, with self-replication being one of them. Von Neumann, using a few sets of crisp rules, constructed a cellular automaton, which allowed specific structures to accurately self-replicate in a finite number of time steps.

The boost in the computing power of personal computers in the last decade introduced a new field of research. This field is dedicated to the modelling and analysis of the behaviour of groups of artificial organisms. Nowadays we commonly address such artificial organisms with the term animat, which was first introduced by Wilson [7]. Representative examples of group behaviours that can be found in nature are flocks of birds, herds of sheep, schools of fish, packs of wolves, swarms of bees, ant foraging, etc.

In this article we focus our attention on the boid - a special type of animat - which was first introduced by Reynolds [5]. Observing the behaviour of a group of boids, a strong resemblance to the characteristic behaviour of a flock of birds can be sensed. Reynolds also states that the boid model can be used to simulate behaviours of herds and schools. The behaviour of every boid is based on a set of

T. Bilgiç et al. (Eds.): IFSA 2003, LNAI 2715, pp. 195–202, 2003.

geometrical expressions. On their basis every boid (every flock member) chooses the direction and speed of flight that allows him to be in the flock. From the point of view of a boid the processing of this decision takes place in discrete time steps, and from the point of view of the flock it takes place simultaneously. If the boid is by definition a model of an independent flock member, we argue the use of geometrical expressions as a source of the boid's decision. We find it contradictory from at least three points of view:

– it is hardly imaginable that animals have the ability to sense crisp accurate numerical data (such as distance) from the environment,
– it is hardly imaginable that animals have the ability to execute accurate numerical or geometrical calculations - the basis of the Reynolds model,
– the crisp numerically geometrical concept of the boid model was constructed empirically and not as a result of observation and description of the dynamics of a real flock, and is as such incomprehensible to a lay public.

In the article we first present a formal definition of the boid model and explain its background. We continue with the implementation of fuzzy logic in the decision process, and in this way generalize and upgrade the existing definitions with logic.

2 Definition of Boids

In this section we present a formal definition of the boid model. For better understanding of the definition we first present the Moore automaton [1] and the animat [2], [7] formal definitions.

Definition 1. *A Moore automaton is defined as a five-tuple* $< X, Q, Y, \delta, \lambda >$, *where* X, Q *and* Y *are finite non-empty sets representing the input alphabet, the internal states and the output alphabet respectively.* δ *is a mapping called the transition function and* λ *is a mapping called the output function:*

$$\delta : X \times Q \rightarrow Q, \tag{1}$$

$$\lambda : Q \rightarrow Y. \tag{2}$$

At any discrete time step t the automaton is in a state $q(t) \in Q$ emitting the output $\lambda(q(t)) \in Y$. If an input $x(t) \in X$ is applied to the automaton, in the next discrete time step $t+1$ the automaton instantly assumes the state $q(t+1) = \delta(x(t), q(t))$ and emits the output $\lambda(q(t+1))$.

Definition 2. *An animat is a special Moore automaton* $A =< X, Q, Y, \delta, \lambda >$, *where the transition function* δ *is defined by a sequence of three levels of functions. These are specified by two function vectors* $\boldsymbol{P}$ *and* $\boldsymbol{S}$ *and a mapping* B *named the perception functions vector, steering functions vector and behaviour function respectively. Formally these are defined as:*

$$P_i : X \times Q \rightarrow \mathcal{P}(X),\ i = 1, ..., k, \tag{3}$$

$$S_j : \mathcal{P}(X)^k \times Q \to F,\ \ j = 1, ..., l, \tag{4}$$

$$B : F^l \times Q \to Q. \tag{5}$$

Let us sketch the appropriate picture informally. At any discrete time step t the input $x(t) \in X$ is the current state of the world, which is a finite non-empty set of animats. The three processing levels of the transition function try to imitate the behaviour of an animal. More precisely, they represent the perception, goal selection and action selection processes. The first level using the perception functions P_i (3) selects from the input $x(t) \in X$ only according to the specific perception function P_i relevant information - the observed animat's neighbours, for example - thus obtaining a vector of neighbour states. These are the input for the steering functions S_j (4), which calculate a vector of steering forces. Finally the steering forces are combined by the behaviour function B (5), and the animat's next discrete time step state $q(t+1)$ is generated.

Definition 3. *A Boid B is an animat, where the perception functions vector is defined as* $\boldsymbol{P} = (P_f)$*, the steering functions vector is defined as* $\boldsymbol{S} = (S_s, S_a, S_c)$ *and the behaviour function is defined as* B_{pa}*. The animat's state at a discrete time step t is* $q(t)$ *as defined in eq.(6), where* $p(t) \in \mathbb{R}^d (d = 2, 3)$ *is the boid's position in space,* $v(t) \in \mathbb{R}^d (d = 2, 3)$ *is the boid's velocity, r is the boid's radii of perception, m is the boid's mass, maxf is the boid's maximal force and maxs is the boid's maximal speed.*

$$q(t) = (p(t), v(t), r, m, maxf, maxs),\ \ q(t) \in Q, \tag{6}$$

$$P_f\left(x(t), q_c(t)\right) = \left\{(s_i, q_{B_i}(t)) : B_i \in x(t), D(q_c(t), q_{B_i}(t)) < r\right\}, \tag{7}$$

$$s_i = 1 - \left(\frac{D(q_c(t), q_{B_i}(t))}{r}\right)^2 . \tag{8}$$

The perception function P_f is defined with eq.(7). At any discrete time step t it, based on the input $x(t) \in X$ and the observed boid's state $q_c(t)$, returns a set of pairs $(s_i, q_{B_i}(t))$, where $q_{B_i}(t)$ is the internal state of boid B_i, whose distance from the observed boid is less then r and s_i is the level of importance of boid B_i. The importance s_i decreases with the square of distance. The distance metric $D(q_c(t), q_{B_i}(t))$ is Euclidean distance and is in the case of $d = 2$ given with eq.(9).

$$D(q_c(t), q_{B_i}(t)) = \sqrt{(q_c(t).p.x - q_{B_i}(t).p.x)^2 + (q_c(t).p.y - q_{B_i}(t).p.y)^2}. \tag{9}$$

The steering functions vector **S** is represented by three steering functions named separation S_s, alignment S_a and cohesion S_c. Stated briefly as rules, and in order of decreasing precedence, they are [5]:

– *separation*: avoid collisions with nearby neighbours - attempt to keep an "appropriate" distance from nearby neighbours,

- *alignment*: attempt to match the speed and direction of flight with nearby neighbours,
- *cohesion*: attempt to stay close to nearby neighbours - attempt to move into the centre of the nearby neighbours.

The boid's next discrete time step state $q(t+1) \in Q$ is calculated by combining these three urges - steering forces. The formal definitions of the S_s, S_a, S_c steering functions as well as of the behaviour function B_{pa} can be found in [2], [5] and will be omitted in this paper.

3 Fuzzification of Boids

In this section we will implement one of the boid's urges in fuzzy logic and through simulation show its advantages. Simulation showed [2] that the urge of alignment has the biggest influence on the boid's behaviour when it is a member of a flock. What is more, it showed that this urge is also the most suitable for the fuzzy logic implementation. As already mentioned, the foremost purpose of this urge is to match the speed and direction of flight with those of the nearby neighbours.

According to the definition of the animat, the result of the steering function is a steering force, which is, according to the definition of the boid, the force required to achieve the desired next discrete time step state. The alignment steering function S_a is defined with eq.(10), where N_f is the set of neighbour states returned by the perception function P_f:

$$S_a(N_f, q_c(t)) = \left(\frac{1}{|N_f|} \sum_{(s_i, q_{B_i}) \in N_f} (q_c(t).v + s_i(q_{B_i}(t).v - q_c(t).v)) \right) - q_c(t).v. \quad (10)$$

The alignment steering function is concerned only with the speed and direction of flight of the nearby neighbours and ignores their positions. The vectors $q_c(t).v$ and $q_{B_i}(t).v$ therefore represent the observed boid's and the neighbour's velocity vectors. The velocity vector gives the relative position changes per coordinate axis in the Cartesian coordinate system and thus describes the boid's speed and direction of flight at time step t. The main distinction of the alignment urge is its predictive collision avoidance [5]. This is mainly caused by the reason that if a boid does a good job aligning with its nearby neighbours, it is unlikely that it will collide with any of them in the near future.

Let us represent s_i - the level of importance of boid B_i - with a linguistic variable Imp composed of three fuzzy sets (LOW, MEDIUM and HIGH). To continue, we shall use a linguistic variable $OrDiff$ composed of three fuzzy sets (LEFT, SAME and RIGHT) to represent the relative difference of the direction of flight between the observed boid and boid B_i. Finally, we shall use a linguistic variable $SpdDiff$ composed of three fuzzy sets (SLOWER, SAME and FASTER) to represent the relative difference of the speed of flight between the observed boid and boid B_i.

Let us declare the linguistic variables $OrChng$ and $SpdChng$, which can be decomposed into the same set of terms as $OrDiff$ and $SpdDiff$, and represent the desired orientation and speed changes respectively. Then the following rules give the fuzzy alignment steering function:

```
if (OrDiff is SAME) then (OrChng is SAME),
if (Imp is LOW) and (OrDiff is RIGHT) then (OrChng is SAME),
if (Imp is MEDIUM) and (OrDiff is RIGHT) then (OrChng is RIGHT),
if (Imp is HIGH) and (OrDiff is RIGHT) then (OrChng is RIGHT),
if (Imp is LOW) and (OrDiff is LEFT) then (OrChng is SAME),
if (Imp is MEDIUM) and (OrDiff is LEFT) then (OrChng is LEFT),
if (Imp is HIGH) and (OrDiff is LEFT) then (OrChng is LEFT),
if (SpdDiff is SAME) then (SpdChng is SAME),
if (Imp is LOW) and (SpdDiff is FASTER) then (SpdChng is SAME),
if (Imp is MEDIUM) and (SpdDiff is FASTER) then (SpdChng is FASTER),
if (Imp is HIGH) and (SpdDiff is FASTER) then (SpdChng is FASTER),
if (Imp is LOW) and (SpdDiff is SLOWER) then (SpdChng is SAME),
if (Imp is MEDIUM) and (SpdDiff is SLOWER) then (SpdChng is SLOWER),
if (Imp is HIGH) and (SpdDiff is SLOWER) then (SpdChng is SLOWER).
```

4 The Results of Experiments

The two graphs in Fig. 1 at point (x, y) give the alignment steering force in the case when the observed boid is at location (x, y) travelling away from the centre with speed $maxs$ and its only neighbour is at location (0,0) travelling in the positive y direction with speed $maxs$. The left graph stands for the crisp implementation of the function, whereas the right stands for our fuzzy logic implementation. It can be seen that even with a simple set of fuzzy logic rules, for which we did not use fitting or other forms of automatic generation, we get a remarkably similar mapping. It can also be noticed that with the fuzzy implementation distant neighbours have less impact on the change of direction and speed of the observed boid (see outer perimeter in Fig. 1). The similarity of the two implementations is even more evident in the metrics, which will be presented later on.

To test the quality of the alignment steering function based on fuzzy logic we run a simple experiment. The experiment included fifty boids in an uninteresting environment; an environment without obstacles. Every boid had random initial position, speed and direction. The other parameters of their states were fixed and equal for all boids. We ran 2000 steps of the simulation, where at each frame we measured the cumulative number of collisions, the number of flocks, average flock speed, average flock speed variation, average flock direction and average flock direction variation. In the paper we will, for reasons of limited space, present only the graphs of the most interesting metrics, the others will be only commented upon.

The cumulative number of collisions is the same in both cases, although the collisions in the case of the fuzzy logic implementation happen earlier in the

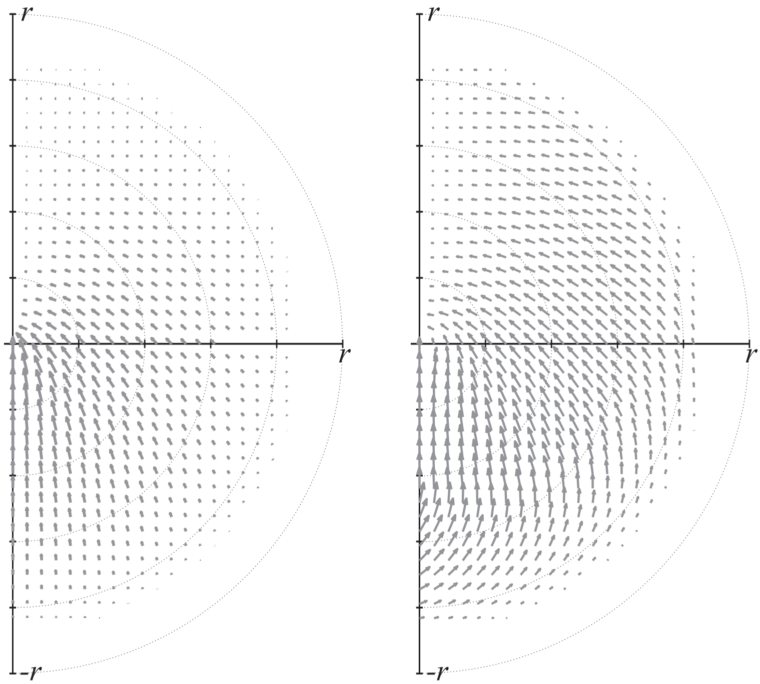

Fig. 1. Graph of the alignment steering function (left) crisp, (right) fuzzy implementation.

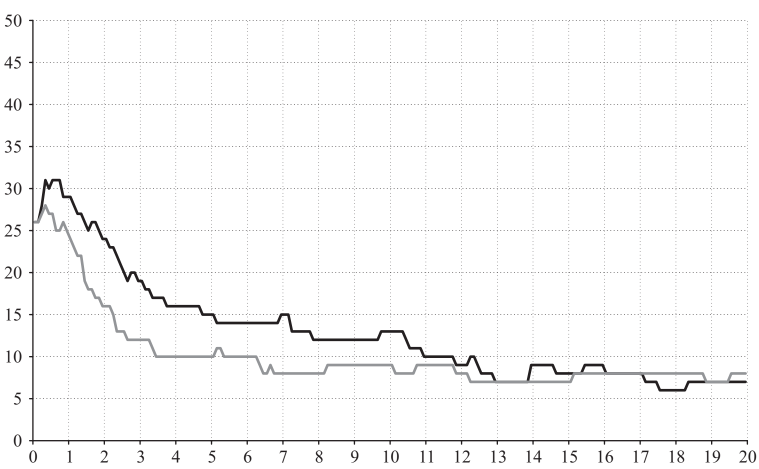

Fig. 2. Number of flocks: (black) crisp, (grey) fuzzy.

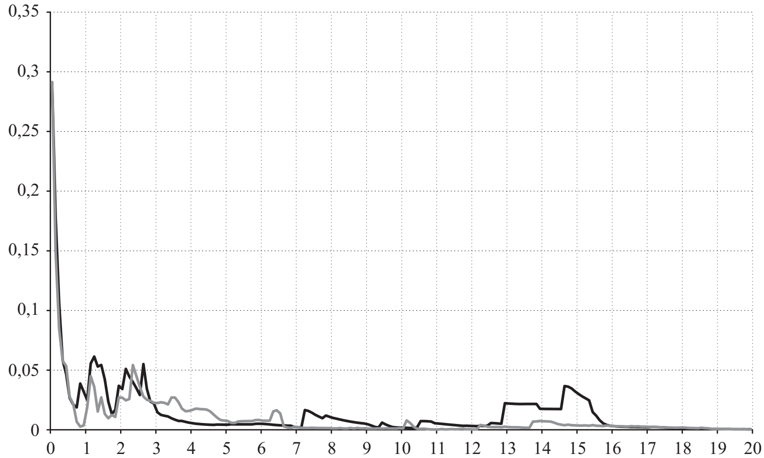

Fig. 3. Average flock speed variation (black) crisp, (grey) fuzzy logic.

simulation. The graph in Fig. 2 shows the number of flocks. As we can see, the fuzzy logic implementation in this case generates the flocks faster than the crisp implementation. Both versions shows similar signs of being unable to keep the flocks together, which would, in an uninteresting environment, not be expected. Nevertheless this deficiency is less evident in the fuzzy logic implementation. Therefore we can conclude that due to the almost identical tendencies of both implementations they behave almost identically where the fuzzy logic implementation has a slight lead over the crisp implementation.

The graph in Fig. 3 shows the average flock speed variation. As we can see the graph also shows similar tendencies of both implementations. Nevertheless it looks as if the fuzzy logic implementation gives better results since the last part of the graph has no turbulences (see frames 120-160).

The graph in Fig. 4 shows the average flock direction variation and similarly as Fig. 3 shows almost identical tendencies of both implementations. The earlier mentioned superiority of the fuzzy logic implementation is even more evident in this graph (see frames 75-185).

5 Acknowledgements

The work presented in this paper was done at the Computer Structures and Systems Laboratory, Faculty of Computer and Information Science, Ljubljana, Slovenia and is part of the Ph.D. thesis being prepared by I. Lebar Bajec.

6 Conclusion

In this paper we explore the use of fuzzy logic as a tool for the construction of artificial animals (animats). We limited our research to the construction of a

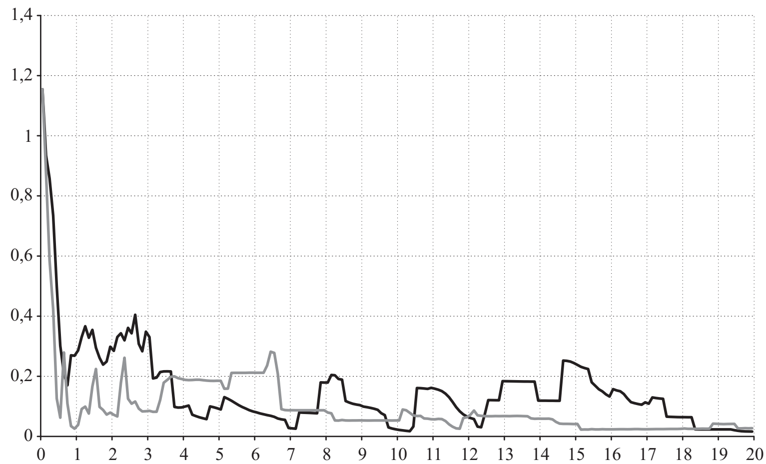

Fig. 4. Average flock direction variation (black) crisp, (grey) fuzzy logic.

boid - a special type of animat. In our case, analogous to other fields of modelling [3], [4], [6], the fuzzy logic approach results as a more suitable and user friendlier than the traditional crisp numerical approaches. We introduced a set of simple linguistic rules that describe the boid's urge of alignment when in a flock. The behaviour of a group of boids that uses these rules is comparable to the behaviour of a group of boids that uses the original geometrical function. This proves that a flock member can base its decisions purely on unclear evaluations of its environment and linguistic rules even without the knowledge of the Newton's laws of motion.

References

1. Kohavi Z.: Switching and Finite Automata Theory. McGraw-Hill, Inc., (1978).
2. Lebar Bajec I.: Computer model of bird flocking. MSc Thesis, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, (2002).
3. Mraz M., Lapanja I., Zimic N., Virant J.: Notes on fuzzy cellular automata. Journal of Chinese Institute of Industrial Engineers, Vol.17, No.5, 469-476, (2000).
4. Mraz M., Zimic N., Virant J.: Intelligent bush fire spread prediction using fuzzy cellular automata, Journal of Intelligent and Fuzzy Systems, Vol.7, 203-207, (1999).
5. Reynolds C.W.: Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics (SIGGRAPH87 Conference Proceedings), Vol.21, No.4, 25-34, (1987).
6. Virant J.: Design considerations of time in fuzzy systems. Kluwer Academic Publishers, (2000).
7. Wilson S.W.: Knowledge Growth in an Artificial Animal, Proceedings of ICGA'85, Pittsburgh, PA, 16-23, (1985).