A Distributed Behavioral Model Using Neural Fields

Mohamed Oubbati, Josef Frick, and Günther Palm

Institute of Neural Information Processing, University of Ulm.
Albert-Einstein-Allee 11. 89081 Ulm, Germany
{mohamed.oubbati,josef.frick,guenther.palm}@uni-ulm.de

Abstract. We investigate the use of neural fields for building a distributed behavioral model enabling several agents to move in a flock. No leader is required, and each agent is implemented as an independent element that follows its own behavioral model which is composed of four steering behaviors: *separation*, *cohesion*, *alignment* and *obstacle avoidance*. The synchronized motion of the flock emerges from combination of those behaviors. The control design will be discussed in theoretical terms, supported by simulation results.

Keywords: Neural fields, autonomous agent, group behavior, navigation.

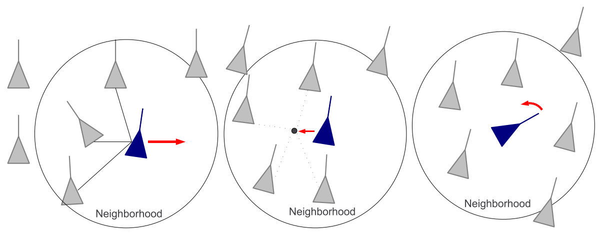
1 Introduction

The creation and evolution of a group of animals, such as flock of birds or a bank of fishes, has received attention in several research fields such as artificial life [1], virtual reality [2], data mining [3] and robotics [4]. The aim is to understand how they stay close together, never collide with each other or with obstacles, but rather create a perfectly synchronized motion. In his pioneering work, Reynolds [5] proposed a distributed behavioural model for simulating the motion of a flock of birds. Each element (*boid*) of the flock is implemented as an independent actor that follows its own behavioral model composed of few rules, such as keep distance from other members, fly toward a goal and avoid collisions. Flake [6] added a fourth rule, "*view*", that indicates that a boid should move away from any boid that blocks its view.

In this paper, we investigate the use of neural fields [7] to model some steering behaviors proposed by Reynolds in [8]. Neural fields are equivalent to continuous recurrent neural networks, in which neurons are laterally coupled through an interaction kernel and receive external inputs (stimuli). The correct choice of the parameters of the field enables the existence of one solution, called *single-peak* or *mono-modal* solution. In this solution, when an input of a stimulus is very large compared with the within-field cooperative interaction, a single-peak will be stabilized by interaction, and it remains there even if the stimulus is removed. The key idea of using neural fields in navigation is to provide sensory information as stimulus, and the position of the peak will decode the correspondant behavior [9,10,11]. In a previous work we investigated how neural fields can offer a solution for the problem of moving multiple agents as a team in formation [12]. The objective was to *acquire a target*, *avoid obstacles*, and *keep a geometric configuration* at the same time. The strategy was to design a leader which guides followers. In this work, no leader is available, and the synchronized motion of the "flock"

T. Honkela et al. (Eds.): ICANN 2011, Part II, LNCS 6792, pp. 32–39, 2011.

results from combination of every agent's steering behaviors. For an agent to participate in a flock, it adjusts its movements in accordance with the movements of its neighbors, i.e. stay close to its neighbors (*cohesion*), avoid collisions with them (*separation*) and move in their average direction (*alignment*) (Fig. 1). The global stimulus is designed by defining the relevance of each behavior relatively to the actual situation. We also consider *obstacle avoidance* with the highest priority among other stimuli. The control design will be discussed in theoretical terms, supported by simulation results.



(a) Separation (b) Cohesion (c) Alignment

Fig. 1. Steering Behaviors

2 Behavioral Model

The equation of a one-dimensional dynamic neural field (DNF) is given by

$$\tau \dot{u}(\varphi, t) = -u(\varphi, t) + S(\varphi, t) + h + \int_{-\infty}^{+\infty} w(\varphi, \acute{\varphi}) f(u(\acute{\varphi}, t) d\acute{\varphi} \qquad (1)$$

The parameter $\tau > 0$ defines the time scale of the field, $u(\varphi, t)$ is the field excitation at the position (neuron) φ and at time t. We used the symbol φ for neurons, because each neuron will encode a possible heading direction for the agent. The constant h defines the resting activity level of the neurons, and $f(u)$ is the local activation function, usually chosen as a sigmoid function. The stimulus $S(\varphi, t) \in \Re$ represents the input of the field at position φ and at time t. A nonlinear interaction between the excitation $u(\varphi, t)$ at the position φ and its neighboring positions $\acute{\varphi}$ is achieved by the convolution of an interaction kernel $w(\varphi, \acute{\varphi})$. That is, the activity change of a neuron φ depends on its actual activity level, the weighted input from other neurons, and the external input (stimulus) at its position. Depending on the parameter h, and the functions S, f, and w, equation (1) can produce a so called *single-peak* solution [7]. In this solution, when an input of a stimulus is very large compared with the within-field cooperative interaction, a single-peak will be stabilized by interaction, and it remains there even if the stimulus is removed. This interaction has also the effect to "push" the peak towards positions where the stimulus has maximum values. In our application the position of the peak decodes the movement direction, which is defined through distances and angles to the neighbors. The stimulus $S(\varphi, t)$ contains information about desired behaviors *separation*, *cohesion* and *alignment*.

2.1 Stimulus Design

We consider N_a autonomous agents moving in the plane, and each one updates its position and heading according to its neighbors. The neighbors of an agent i ($1 \leq i \leq N_a$) at time t are those which lie within a circle of radius r centered at the current position of the agent i. We denote by $N_i(t)$ the number of neighbors of the agent i at time t, such that

$$N_i(t) = \{\forall j / \|x_i^2(t) - x_j^2(t)\| + \|y_i^2(t) - y_j^2(t)\| < r\} \tag{2}$$

where $(x_i(t), y_i(t))$ is the position of the agent i at time t. The distance of an agent i to each neighbor $j \in N_i(t)$ at time t is

$$d_{ij}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \tag{3}$$

A behavior becomes active, if it holds the highest stimulus. We define λ_{min} as a threshold, such that if $d_{ij} < \lambda_{min}$, *Separation* is active. In this situation, each agent should move into the direction, which is opposite to the angle of the nearest neighbor. The behavior *Cohesion* aims to approach the agent to its neighbors as close as possible. This behavior becomes active when $d_{ij} > \lambda_{max}$, where λ_{max} is a predefined threshold distance. Finally, *Alignment* attempt to adjust the agent's direction according to the average direction of its neighbor's angle. The global stimulus is designed by defining the relevance of each behavior through the amplitudes $A_{separation}$, $A_{cohesion}$ and $A_{alignment}$ (Fig. 2). Obviously, *Separation* has the highest priority when $d_{ij} < \lambda_{min}$. On the other hand, when the agents are "too" far from each other, i.e. $d_{ij} > \lambda_{max}$, *Cohesion* behavior must take place, in order to keep the shape of the group. Finally, *Alignment* behavior occurs in the region when $\lambda_{min} < d_{ij} < \lambda_{max}$.

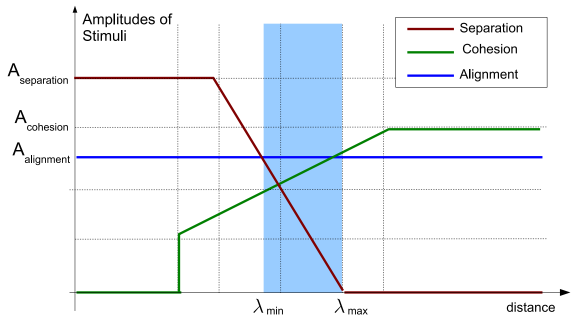


Fig. 2. Relevance of behaviors vs. distance between neighbors

3 Implementation and Results

We consider 4 agents, and each one has its own DNF. We choose the agents' headings φ_k ($k \in \{1, 2, 3, 4\}$) relative to a world-fixed reference direction as behavioral variables. By means of a codebook we use N discrete directions from $-\pi$ to $+\pi$. We discretize (1) as

$$\tau \dot{u_k}(\varphi_i, t) = -u_k(i,t) + \sum_{j=1}^{N} w(\varphi_i, \varphi_j) f[u_k(\varphi_j, t)] + S_k(\varphi_i, t) + h \tag{4}$$

and we chose $N = 60$ neurons, which means that each direction N decodes a step of 6^o, and the interaction kernel as:

$$w(\varphi_i, \varphi_j) = 2.8 \cdot \left(1 - \left(\frac{d(\varphi_i, \varphi_j)}{6}\right)^2\right) \cdot exp\left(-\left(\frac{d(\varphi_i, \varphi_j)}{2 \cdot 6}\right)^2\right) + h \tag{5}$$

where $d(\varphi_i, \varphi_j)$ defines the distance between neurons φ_i and φ_j. The global inhibition $h = -1.3$ allows only one localized peak on the field. Based on the stimulus design described earlier, we implement the behaviors as (Fig. 3)

Separation:

$$s_{separation}(\varphi_i) = \max_k \left(A_{separation}\left(1 - \frac{2 \cdot d(\varphi_i, \phi_{sep})}{30}\right)\right) \tag{6}$$

where φ_{sep} is is the position on the field decoding the opposite direction to the nearest neighbor.

Cohesion:

$$s_{cohesion}(\varphi_i) = A_{cohesion}\left(1 - \frac{2 \cdot d(\varphi_i, \phi_{middle})}{30}\right) \tag{7}$$

where φ_{middle} is the position on the field decoding the direction to the middle of the neighbors.

Alignment:

$$s_{alignment}(\varphi_i) = A_{alignment} \cdot \left(1 - \frac{2 \cdot d(\varphi, \phi_{flock})}{30}\right) \tag{8}$$

where φ_{flock} is the position on the field that decodes the flock direction.

At each situation the *flock* stimulus is chosen as

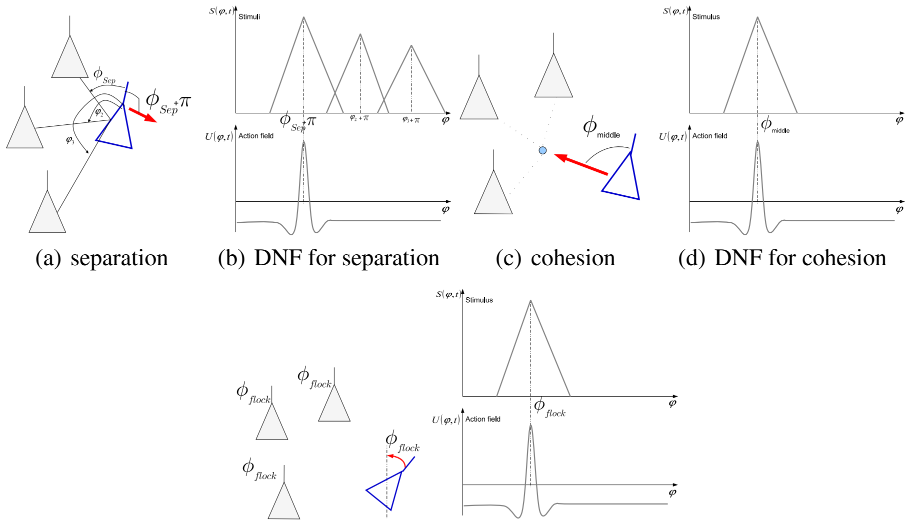
$$S_{flock} = \max(S_{separation}, S_{cohesion}, S_{allignment}) \tag{9}$$

The behavior *obstacle avoidance* is chosen as a Gaussian function centered at the direction of an obstacle φ_O

$$S_{obstacle}(\varphi_i) = C_O e^{-\sigma_O(\varphi_i - \varphi_O)^2} \tag{10}$$

where C_O and σ_O are positive constants, and σ_O defines the range of inhibition of an obstacle. The global stimulus of each agent will be then

$$S_{global} = S_{flock} - S_{obstacle} \tag{11}$$



(a) separation (b) DNF for separation (c) cohesion (d) DNF for cohesion

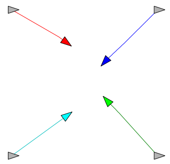
(e) alignment (f) DNF for alignment

Fig. 3. Behaviors and their representation on the DNF.

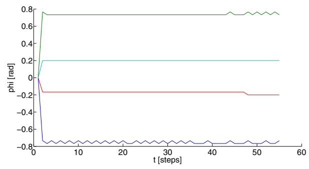
After the stabilization of the field, the most activated neuron decodes the movement direction to be executed:

$$\varphi_{peak} = arg_{\max}\{u_k(\varphi_i)|i \in [1, N]\} \tag{12}$$

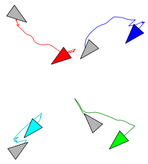
Two experiments are designed to test the validity of the behavioral model with DNFs. In eah experiment the initial position and orientation are assigned beforehand. The first experiment (Fig. 4) consists in observing the agents' behavior in an obstacle-free environment. At the beginning the agents started with the *cohesion* behavior, in order to reach their desired positions in the flock (Phase 1). As they were too close to each other, their global stimuli were dominated by the *separation* stimulus, which brought each agent away from its neighbors (Phase 2). Once the flock was stabilized, each agent tried through *alignment* to follow the global direction of the group (Phase 3). To test robustness of the model we forced one agent to change its direction (Phase 4). The model could recover to this "perturbation", and the agents adjust their positions and orientation according to this new situation, and maintain the flock motion. The global path is illustrated in Fig. 4. (i). The second experiment (Fig. 5) shows the case of flock motion with obstacle avoidance. There is an important distinction between *obstacle avoidance* and *separation* behavior. *Separation* is used to steer away from the predicted future position caused by the *cohesion*, whereas *obstacle avoidance* takes action only when a nearby obstacle is detected by the agent's sensors. Fig. 5. (c) shows how the contribution of all stimuli provide the appropriate heading directions, which permits to avoid the obstacle. After passing the obstacle the stimulus of obstacle avoidance is removed, and the group continues its movement by combining the three flock behaviors. The global path is illustrated in Fig. 5. (g).



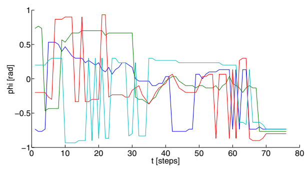
(a) Phase 1: Cohesion



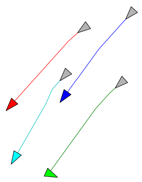
(b) Phase 1: Headings vs. time



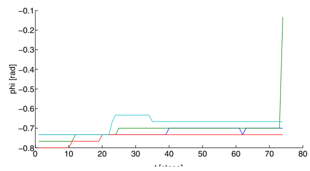
(c) Phase 2: Separation



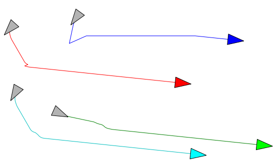
(d) Phase 2: Headings vs. time



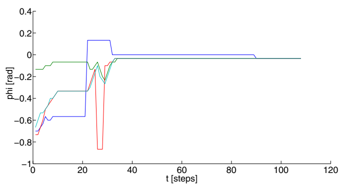
(e) Phase 3: Alignment



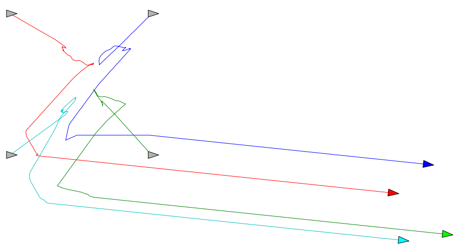
(f) Phase 3: Headings vs. time



(g) Phase 4: Test of changing direction

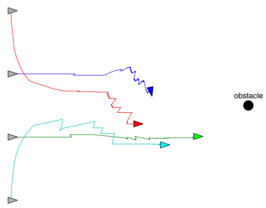


(h) Phase 4: Headings vs. time

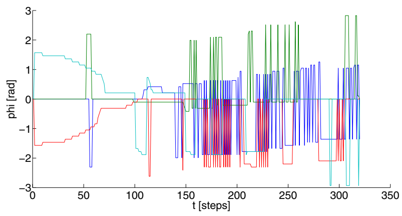


(i) Global path

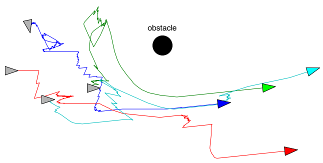
Fig. 4. Flock Motion



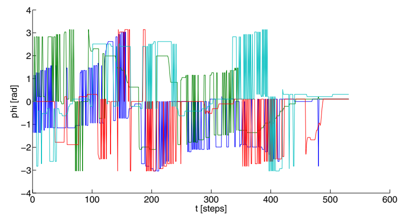
(a) Cohesion



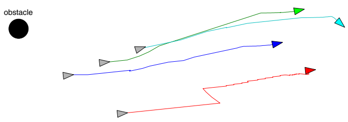
(b) Headings vs. time



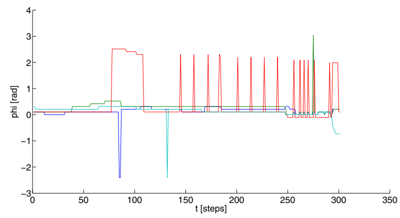
(c) Obstacle avoidance



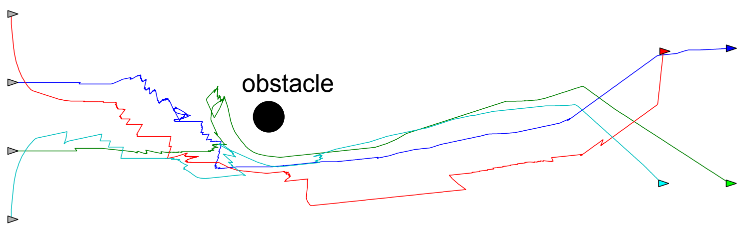
(d) Headings vs. time



(e) Recovering from obstacle avoidance



(f)



(g) Global path

Fig. 5. Flock Motion with Obstacle Avoidance

4 Conclusion

We designed a behavioral model for flock motion using DNFs. No leader was available, and the synchronized motion emerged from combination of every agent's steering behaviors *separation*, *alignment*, *cohesion* and *obstacle avoidance*. These behaviors were first transformed to separate stimuli entries, and then combined in a global stimulus by assigning a situation-based priority to each behavior. The simulation tests demonstrate the feasibilty of the approach, but a real-world implementation is still needed to confirm the effectiveness of the results achieved. Actually, we are working on implementing this design on 3 small real robots to test its performance in uncertain environments with real-time contraint.

References

1. Terzopoulos, D., Tu, X., Grzeszczuk, R.: Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. Artificial Life 1, 327–351 (1994)
2. Ulicny, B., Thalmann, D.: Towards interactive real-time crowd behavior simulation. Computer Graphics Forum 21(4), 767–775 (2002)
3. Abraham, A., Grosan, C., Ramos, V.: Swarm Intelligence in Data Mining. SCI, vol. 34. Springer, Heidelberg (2006)
4. Beni, G.: From swarm intelligence to swarm robotics. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics 2004. LNCS, vol. 3342, pp. 1–9. Springer, Heidelberg (2005)
5. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. SIGGRAPH Comput. Graph. 21, 25–34 (1987)
6. Flake, G.W.: The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation. MIT Press, Cambridge (2000)
7. Amari, S.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biol. Cybern. 27, 77–87 (1977)
8. Reynolds, C.W.: Steering behaviors for autonomous characters. In: Game Developers Conference, pp. 763–782 (1999)
9. Schöner, G., Dineva, E.: Dynamic instabilities as mechanisms for emergence. Developmental Science 10(1), 69–74 (2007)
10. Oubbati, M., Palm, G.: A neural framework for adaptive robot control. Journal of Neural Computing and Applications 19(1), 103–114 (2010)
11. Sandamirskaya, Y., Schöner, G.: An embodied account of serial order: How instabilities drive sequence generation. Neural Network 23, 1164–1179 (2010)
12. Oubbati, M., Palm, G.: Neural fields for controlling formation of multiple robots. In: International Symposium on Computational Intelligence in Robotics and Automation, pp. 90–94. IEEE Computer Society Press, Los Alamitos (2007)