Modeling flocks with perceptual agents from a dynamicist perspective

Angel Zaldivar Pino

Manuel Gonzalez Bedia

Francisco José Serón Arbeloa

Advanced Computer Graphics Group (GIGA) - University of Zaragoza

María de Luna, 1

50015 Zaragoza, Spain

Tel. +(34) 976761949

Abstract

Computational simulations of flocks and crowds have typically been processed by a set of logic or syntactic rules. In recent decades a new generation of systems has emerged from dynamicist approaches in which the agents and the environment are treated as a pair of dynamical systems coupled informationally and mechanically. Their spontaneous interactions allow them to achieve the desired behavior. The main propo-

1

sition assumes that the agent does not need a full model or to make inferences before taking actions; rather, the information necessary for any action can be derived from the environment with simple computations and very little internal state. In this paper we present a simulation framework in which the agents are endowed with a sensing device, an oscillator network as controller and actuators to interact with the environment. The perception device is designed as an optic array emulating the principles of the animal retina, which assimilates stimuli resembling optic flow to be captured from the environment. The controller modulates informational variables to action variables in a sensory-motor flow. Our approach is based on the Kuramoto model that describes mathematically a network of coupled phase oscillators and the use of evolutionary algorithms, which is proved to be capable of synthesizing minimal synchronization strategies based on the dynamical coupling between agents and environment. We carry out a comparative analysis with classical implementations taking into account several criteria. It is concluded that we should consider replacing the metaphor of symbolic information processing by that of sensory-motor coordination in problems of multi-agent organizations.

Keywords: crowd simulation, flocking simulation, dynamical modeling, oscillators networks, artificial evolution

1 Introduction

The cost of producing animated movies is high. The fundamental objective of computer animation programming is to select techniques and design tools that are expressive enough for animators to specify what they intend, yet at the same time are powerful enough to relieve animators of the task of specifying any details they are not interested in [1]. At the highest level of abstraction, the animator becomes director. At his command are "intelligent" characters who know how to get the job done. They only need to be told what to do. It is the character's job to find a way of completing the task.

There are many issues in computer animation and simulation when it is necessary to model virtual crowds of autonomous agents, in which the motion of an animated character is choreographed based on procedural or rule-based motion techniques [2][3]. Usually, the procedural techniques are used with physics-based animation, and rule-based motion techniques are used with behavioral or intelligent animation. Symbolicism has been the prevailing paradigm in the world of virtual agents simulation, often encountering some serious drawbacks when dealing with real world problems; principally, low expressiveness, hard implementation and the limits imposed by the richness of the environment.

In this work we approach the problem from a dynamic perspective, assuming that the agent does not need a full model or to make inferences about the environment before taking actions, with simple computations and very little internal state. We argue that such a specification involves the coupling of sensory information with appropriately structured control

systems to generate action.

Our objective can be defined as modeling the locomotory behavior of virtual worlds based on perceptually controlled dynamic agents whose control architecture is a dynamical system, which will modulate informational variables captured by the visual sensing device into action variables understandable to an actuator mechanism under a sensory-motor flow.

The novelty of our proposal focuses on the design of a synthetic sensory perception model that emulates the principles of geometric projection and stimulus translation that occur in vertebrate retina, and the linking of this model with the controller through dynamic laws.

The approach aims to replace the notion of symbolic representation and processing by modeling in terms of dynamical coupling as a more appropriate metaphors for defining adaptive behavior in virtual agents.

The paper is organized as follows. The following section discusses some related works about the dynamic perspective of behaviour, perception and synchronization. We then present the methods adopted in our model, including the framework for defining an agent as an autonomous entity controlled by a dynamical system, the details of the agent perception and a description of the artificial evolution scheme to optimize the parameters of the system. Following this, we present the results, which show details of the time-series analysis of the simulations and the behaviour of the oscillators network. In order to illustrate the potential of the proposal, we carry out a comparative analysis with simulations of classical implementations under qualitative, quantitative and information-theoretic perspectives. The

4

paper closes with a discussion of the results obtained and proposals for future work.

2 Related works

A virtual agent is a system which has intrinsic goals and exhibit a behavior in a specific domain. In ideal terms, the agent must be able to carry out activities in a flexible manner. Some definitions of "intelligent" agents emphasize goal-directed behavior as the essence of intelligence. An autonomous entity observes through sensors and acts upon an environment using actuators. These characteristics suggest that an agent acts as an automaton, having a close resemblance to an animal's sensory-motor situation. According to Wilson, an animal exists in a sea of sensory signals, where only some signals are significant. The animal is capable of actions which tend to change these signals [4].

As regards with the natural resemblance Douglas et al. [5] suggest that vision should not be viewed as passive information processing but rather as an active "integrated sensory-motor event" [6][7]. The most widely adopted theory about animal perception is derived from the Ecological Psychology of J.J. Gibson [8][7]. The concept of "direct perception" assumes that an animal and its environments are inseparable pairs that should be described in terms of a relevant scale to the animal's behavior. A principal tenet of the direct perception approach is that the perceived stimulus provides adequate information for controlling behavior without further inferential processing or model construction. The animal has direct knowledge of, and also a relationship to, its environment as a result of natural laws. The

Gibsonian approach can be summarized as follows: it is more desirable to put the animal in its environment than to put the environment in the animal.

Gibson introduced the notion of optical flow as being the pattern of the apparent motion of elements in an environment caused by the relative motion of an observer. The concept describes the visual stimulus provided to animals moving through the world. The conversion of this innate ability to a computer capability is crucial in the field of machine vision. The term optical flow has been co-opted by roboticists to incorporate related techniques from the fields of image processing and navigation control, such as motion detection, object segmentation, time-to-contact information, focus of expansion calculations, luminance, motion compensated encoding, and stereo disparity measurement [9][10].

Furthermore, the concept of "affordance" has been applied in the modelling of path-planning and steering for crowds and groups [11]. Each agent perceives the environment through a set of vector and scalar fields, the egocentric property allows to compute a local space-time plan. The perception fields is used to compute a fitness measure for every possible action, defined as an affordance field.

The last two decades have also witnessed radical changes of perspective about the nature of behaviour, leaving behind some of the assumptions of computational functionalism [12][8]. Several works [13][14][15][16][12] have advanced the claim that behaviour is not well understood as symbolic manipulation or connectionist processing, but rather as complex, dynamical interactions of an agent with its environment. These ideas can be used to explain the internal processing which underlies an agent's interactions with the environment,

6

where the coupled nature of a system of equations replaces the idea of "representation". Differential equations systems can capture the agent's behavior through a multi-dimensional space under certain environmental and internal pressures, with low dimensional descriptions [15][16][17][8]. The system should be described at a scale relevant to the agent's behavior and the work of the controller is seen as an interaction of internal and external forces rather than the product of representations and sequential operations.

In the context of crowd simulation there are a series of works that have enriched our vision. A seminal paper that adopts a dynamic approach for crowds modelling is the work of Helbing and Molnar [18]. Here, a system of Non-Linear Equations captures the motion by including several force terms as acceleration, repulsion and attraction. Schuerman et al. propose an advanced abstraction of a crowd [19], the steering logic of an agent can influence the steering decisions of any agent within their sphere of influence. The work of Treuille et a.l is an attempt to bring computational crowds to the real crowds [20], which is based on continuum dynamics. A dynamic potential field integrates global navigation and obstacle without the need of an explicit description of the behaviour. The predictive approach in autonomous navigation have been used in some works, the key issue is that each agent perceives surrounding agents and extrapolates their trajectory in order to react to potential collisions [21]. With the same focus Berg and Manocha [22] propose a model in which each agent assumes that the other agents make a similar collision-avoidance reasoning. A relevant approach based on artificial life is the work of Shao and Terzopoulos [23], this paper addresses the rich complexity of real pedestrians in urban environments. The model integrates

motor, perceptual, behavioral, and cognitive components. The environment is represented using hierarchical data structures, which efficiently support the perceptual queries of the agent their behavioral responses and sustain their ability to plan their actions on local and global scales. Berseth et al. provided a broad view of parameters selection in crowds [24]. This paper investigates the effect that these parameters have on an algorithm's performance, including statistical analysis of the correlations between parameters and performance criteria. An interesting extension of Reynolds's model is the work of Hartman and Benes [25], the paper introduce a complementary force to model the leadership on the flock. The work show the drawbacks that appear in the extension of a rule base system.

Furthermore, much work has been done on dynamical systems as controllers of complex motor behaviors, particularly in the field of locomotion [26]. However, to date there has been very little research on the wider issues in the generation of embodied cognitive behaviors. One of the pioneering steps in this direction is the work of Moioli [27] in which a network of oscillators was used as the controller of simulated robotic agents engaged in some simple tasks. The model used is based on the Kuramoto model of coupled phase oscillators [28][29][30]. Their main contribution is to suggest how oscillatory networks can be provided with input and output mechanisms so that they are able to engage in sensory-motor brain-body-environment loops, allowing robust adaptive couplings between perceptive activity and environmental dynamics.

The Kuramoto model consists of a population of oscillators whose dynamics are governed as shown in equation (1).

8

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^{n} k_{ij} sin(\theta_j - \theta_i), \qquad i = 1, 2, ..., n \tag{1}$$

where n is the number of oscillators, θ_i is the phase of the $i - th$ oscillator, ω_i is the natural frequency and k_{ij} is the coupling gain. This formalism provides a class of dynamical system where the phase differential $\dot{\theta}$ changes at each instant of time the oscillators phase, as shown in equation (2).

$$\theta_i(t + \Delta t) = \theta_i(t) + \dot{\theta}_i \Delta t; \ \Delta t = 1 \tag{2}$$

The oscillators are said to synchronize if the phase differences given by $\theta_j - \theta_i, \forall i, j = 1, 2, ..., n$ become asymptotically constant. Imagine these oscillators as points in a circle. The points then move with the same angular frequency and, hence, the angular distance (phase difference) between the points remains constant with time. The order parameter r is defined as shown in equation (3).

$$re^{i\psi} = \frac{1}{n} \sum_{j=1}^{n} e^{i\theta_j} \tag{3}$$

where ψ is the average phase. The order parameter $r(t)$ with $0 \leq r(t) \leq 1$ is a measure of phase coherence of the oscillator population. If the oscillators synchronize, then the parameter converges to a constant $r_\infty \leq 1$, but if the oscillators add incoherently then the order parameter r remains close to zero.

3   Mathematical framework

The standard model of our theoretical framework is an embodied and situated agent together with its environment. The agent can be described mathematically as a function that associates the sensor inputs to the motor outputs as shown in equation (4).

$$O(t) = C(I(t), e) \tag{4}$$

where $I(t)$ represents the input vector defined by the sensor readings of the agent, $O(t)$ is the output vector, C is the function which maps every possible percept sequence to a possible action, and e is the vector of evolved parameters that characterize the controller. This suggests that a specific architecture can be used to perform different tasks in a specific environment simply by modifying the configuration of its controller; in other words, subjecting it to a different evolutionary synthesis.

The analytical form of the function C will be based on equation (1), containing the system of differential equations that represents the agent's control core. This function induces correlations, thus reducing the high dimensional sensory space to a low-dimensional sub-space of actions [17][31].

Equation (4) is analogous to Gibson's principle of direct perception expressed as $\Delta F_{internal} = g(\Delta flow)$, where the change in the agent's internal forces is a function of the change in the optic flow.

In the following sections we describe our proposal, defining the structures of body, envi-

ronment and controller (unspecified in their parameter values). We first discuss a 2D model of our synthetic retina, before going on to describe 3D perception. Both models are described through three guidelines: *(a)* the sensory layer, *(b)* the iterative behavior loop and *(c)* the body structure. The evolutionary synthesis process is described in the last section.

3.1   2d retinal perception

The vertebrate retina is a light-sensitive layer of tissue interconnected by synapses, which is responsible for the conversion of patterns of light into neuronal signals. Transduction is a cascade of chemical and electrical events through which energy from environmental stimuli is converted to neural activity for the brain to understand and process. Several important features of visual perception can be traced to the retinal encoding for processing light. Motion perception is the process of inferring the speed and direction of elements in a scene based on visual inputs. This process has proven to be a difficult problem from a computational perspective, and extraordinarily difficult to explain in terms of neural processing. When an image moves across the retina, it stimulates a number of receptors sequentially. The brain reads the message and thus can determine the direction of movement, taking into account the direction in which the cells are triggered and the speed.

Mobile agents in a virtual environment are exposed to a continuous stream of ever-changing sensory stimulation, and all the elements of a virtual environment are usually geometrically defined. So, we have new circumstances for collecting information from

the environment where the optical flow can be extracted from the relative position of the observer and the elements of the environment with simple vector processing. The agent can simply concentrate on detecting coordinates, resulting in relatively simple optical flow detection. For this it is essential to have a sensor that detects motion by spatio-temporal correlation.

It must be taken into account that the nature of the stimulus must be consistent with the function of the agent in the environment, and therefore the sensory system design. In addition, the actuators must be designed according to the environment where the agent moves. The key points in this step are that the coupling of the sensor to the controller should be intuitive for the proposed architecture and the coupling of the actuators with the controller output compatible with its own dynamics.

Sensory layer: Our model consists of an agent in a two dimension environment. The sensor detects the position of a particle within its field of view, the point is projected using geometric equations to a discrete array $S_{1 \times m}$ of length m that registers the stimulus, as shown in Figure 1.

The line L_1 is defined by the particle position $P_1(x_1, y_1)$ and the projection point $P_2(x_2, y_2)$ (agent position in the space) and line L_2, being defined by two distinct points $P_3(x_3, y_3)$, and $P_4(x_4, y_4)$. This line is defined orthogonally to the velocity vector of the agent and at a empirical distance from the point P_2. The intersection point $P(x, y)$ between L_1 and L_2 determines the index i of array S that is stimulated by the projection of the particle, as shown in equation (5).

$$x = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$y = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \tag{5}$$

$$u = \frac{(y - y_3)}{(y_4 - y_3)}$$

$$i = \lfloor u * m \rfloor$$

where x and y are the coordinates of the intersection point P, u is the value of independent variable for the point P in the parametric equation of L_2 ($u \in [0..1]$) and i is the index of the array S that is stimulated by the ray L_1.
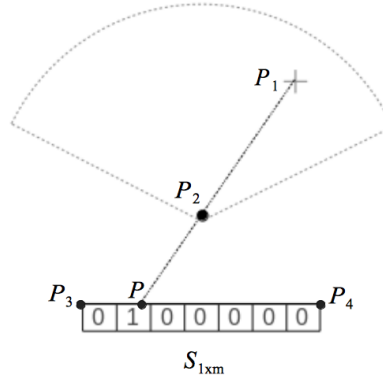
Figure 1: 2d retinal device. A particle within the field of view is projected geometrically in a sensitive array and is translated into a vector that registers the stimulus.

An individual receives signals within a small surrounding neighborhood. The neighborhood is characterized by a distance and an angle (measured from the center of the agent and with respect to the velocity vector, respectively). Signals from outside the local neighborhood are ignored. The neighborhood could be considered a model of limited perception.

The vector S can capture stimuli at various levels according to the task to be performed. The most elementary stimulation is constituted by a binary vector that does not contain a

notion of movement because it only perceives at each instant of time the static position of the particle, as shown in Figure 1. A more complex stimulus can register movement patterns similar to optical flow, as shown in Figure 2. Here, the projection of the target point on the cells is similar but the cells record the time during which the particle excites a specific cell. The time is measured from the input to the visual field of the agent. From the computational viewpoint, this corresponds to discrete values of the simulation time step. The stimulus basically contains the direction of movement and speed. The direction of movement is expressed by the ascending values for the leftward movement or descending for the rightward movement registered in the stimulus vector. The velocity is expressed by the magnitude of the values. The greater the values, the smaller the particle velocity by the visual field and vice versa. If the particle is not within the visual field, the stimulus is zero.
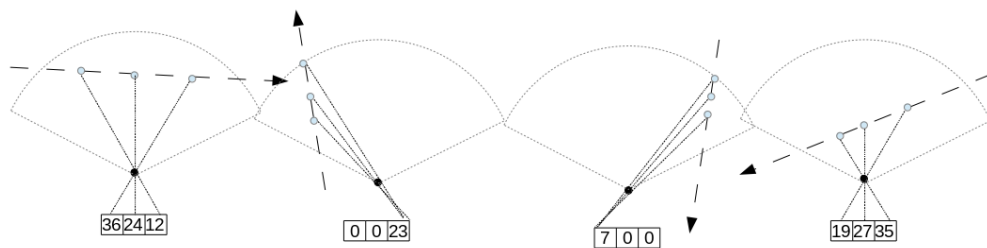

Figure 2: Every sensitive cell records the time that the particle excites a specific cell. The stimulus contains the direction of movement and speed of the particle that crosses the visual field.

When there are many particles in the visual field, the general optical flow is calculated averaging the stimulus generated by each particle independently, as shown in Figure 3. With this new concept of stimulus, the agent basically has a notion of direction and speed of the movement of agents in its environment.
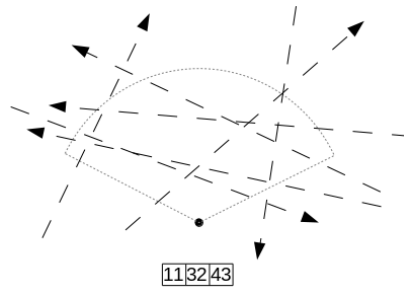
Figure 3: In order to calculate the total optical flow when the visual field is crossed by more than one particle, the optical flow vectors generated for each of the particles must be averaged. The resulting stimulus contains the average direction and the speed of the particles of the neighborhood.

Iterative behavior loop: The agent movements are controlled by a network of three coupled oscillators, all being connected to sensory stimuli, as shown in Figure 4. Each stimulus S at a given instant is modulated by a matrix $A_{m\times 3}$; this multiplication results in a three value vector $I_{1\times 3}$ which is coupled to each of the oscillators, as shown in equation (6).
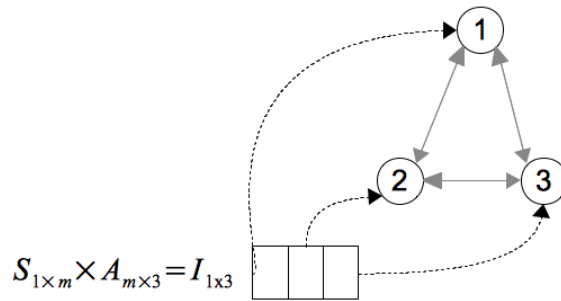

$$S_{1\times m} \times A_{m\times 3} = I_{1x3}$$

Figure 4: Each sense at a given instant is modulated by a matrix A, and the resulting vector I is coupled to each of the oscillators. This factor alters the oscillatory behavior of the network according to environmental interactions.

At each time step, the oscillators network iterates seeking synchronization. The phase differential from a node i is calculated following equation (6). The model is inspired by the Kuramoto equations in such a way that the frequency of each node is the result of the sum

15

of its natural frequency of oscillation with the scaled value of the sensory input related to that node.

$$\dot{\theta}_i = \omega_i + I_i + \sum_{j=1}^{n} k_{ij} sin(\theta_j - \theta_i), \qquad i = 1...3 \tag{6}$$

where I_i is the value of the sensory inputs at instant t for the oscillator i, θ_i is the phase of the $i-th$ oscillator, ω_i is the intrinsic frequency of the oscillators, and k_{ij} is the coupling factor from the $j-th$ to the $i-th$ oscillator. At each iteration the phase from a node i is calculated following equation (2).

The network coupling in our model allows that the natural activity of each oscillator modulates the global activity of the network. The outputs of the system are defined as a function of the phase differences of two particular oscillators (see equation (7)). In this way, the behaviour emerges from the dynamical interplay between the neural system of an agent, its body, and the environment, creating a sensory-motor loop.

Here, the result of the control process influences the operation of the process itself in the next time step as negative feedback. Negative feedback tends to make a system self-regulating. It can produce stability and reduce the effect of fluctuations [32]. Negative feedback loops in which just the right amount of correction is applied in the most timely manner can be very stable, accurate and responsive. It decreases the effects of unpredictable influences from the system's environment. This property allows our model to operate in any realistic environment with many sources of noise or surprises.

Body structure: Embodiment constitutes a physical grounding of the agent, which is forced to face the kind of real-world issues organisms have to deal with. As a result, the morphology of the body often plays a key role in the generation of behavior. Thus, the body not only imposes constraints on the behavior of an agent, but it allows an agent to exploit the world by means of sensory-motor coupling with the environment. The idea of situatedness means that the agent is embedded in a world. Its whole interaction with the environment is controlled by the agent itself [33][17][34]. So, the environment is a rich source of constraints and opportunities for the agent.

The body structure of our agent is modeled abstractedly as a circular body with two diametrically opposed motors which can move forward with different velocities[35][27]. The activation of the motors is defined, for the right and left motor, respectively as a_r and a_l. Thus, the sine of the relative phase between oscillators 1 and 2 activates the right motor while the relative phase of oscillators 1 and 3 activates the left motor, as shown in equation (7). A phase-sensitivity function is applied to each of the phase differences to reduce instabilities in these differences caused by the resetting of the phase of each oscillator when it exceeds 2π.

$$a_r = sin(\theta_1 - \theta_2) + 1.0$$
$$a_l = sin(\theta_1 - \theta_3) + 1.0$$

(7)

The position of the agent in the environment is characterized geometrically by direction γ in scalar form in the range $[0, 2\pi]$, and a vector position (x_n, y_n). At each iteration the

17

new position is calculated according to equations (8).

$$v_t = (a_r + a_l)$$

$$v_a = (a_r - a_l)/2.0$$

$$\gamma = \gamma + v_a \tag{8}$$

$$x_n = x_n + v_t \cdot cos(\gamma)$$

$$y_n = y_n + v_t \cdot sin(\gamma)$$

where v_t is the magnitude of displacement, v_a is the angle variation, γ is the new direction angle (normalized) and x_n and y_n is the new position in x and y axes respectively. Due to the abstract design of the body, the agent has only two directions to turn (right or left) in a time step.

The formalism described above provides a dynamical system which can be parameterized by specifying the modulation matrix, the specific values for the frequency of the oscillators, and the coupling factor between the oscillators ($[\omega_1, \omega_2, \omega_3, A_{m \times 3}, k_{3 \times 3}]$). For example, the exact encoding for defining an agent whose retina has three cells is a vector that contains the three frequency values of the oscillators ω_i, the modulation matrix A of three by three for a total of 9 elements and the coupling matrix k, where nine values are needed for a three oscillators network. This configuration uses $21 (3 + 9 + 9)$ genes of 8 bits each, for a total of 168 bits.

3.2 3d retinal perception

Sensory layer: In order to extend the concept of the synthetic retina proposed in the previous section for 3d spaces, we designed a sensor that can see inside a conical viewing volume, which is defined by the angle d and depth l as shown in Figure 5. Using geometric equations, the particles within the volume are projected to the corresponding receptor cell and translated into a matrix, similarly to the 2d retina model. If there is no stimulus within the volume, the matrix is zero.

The line L is defined by the particle position P_1 and the projection point P_2 (agent position in 3-dimensional space). The plane S is defined by a normal vector N that coincide with the velocity vector of the agent (normalised) and point P_3 defined at a empirical distance from $P2$ on the plane.

Consider the point P the intersection between the line L and the plane S. The equation of a plane can be written as $N \cdot (P - P_3) = 0$. The equation of the line L (point P on the line passing through points P_1 and P_2) can be written as $P = P_1 + u(P_2 - P_1)$. The intersection of these two occurs when $N \cdot (P_1 + u(P_2 - P_1)) = N \cdot P_3$. Solving for u gives as shown in equation (9).

$$u = \frac{N \cdot (P_3 - P_1)}{N \cdot (P_2 - P_1)} \tag{9}$$

where u is the value of independent variable for the point P in the parametric equation of L (evaluating any of the components in the vector equation).

On the other hand, a plane may be described parametrically as the set of all points as shown in equation (10)

$$P = P_3 + sV + tW \qquad (10)$$

where s and t range over all real numbers, V and W are given linearly independent vectors defining the plane, and P_3 is the vector representing the position of an arbitrary point on the plane. Solving the system of equations we obtain the parameters s and t. After a simple process of transformation and discretization based on the position and dimension of the plane in the space, $s\prime$ and $t\prime$ will define the cell of the matrix S that is stimulated by the ray L.

The matrix can capture stimuli at various levels according to the task to be performed, from an elementary stimulation that does not contain a notion of movement to a more complex stimulus as an optical flow.

Iterative behavior loop: The coupling of the sensing device with the controller within the iterative behavior loop has the same philosophy as that described in relation to the 2d space. The agent's movement is controlled by a network of three coupled oscillators, all being connected to sensory stimuli as shown in Figure 6. The retinal matrix is transformed into a vector R product of the concatenation of its rows. Each sense is modulated by a matrix A at a given instant. This multiplication results in a three value vector I which is coupled to each of the oscillators, in the same way as shown in equation (6). At each iteration the
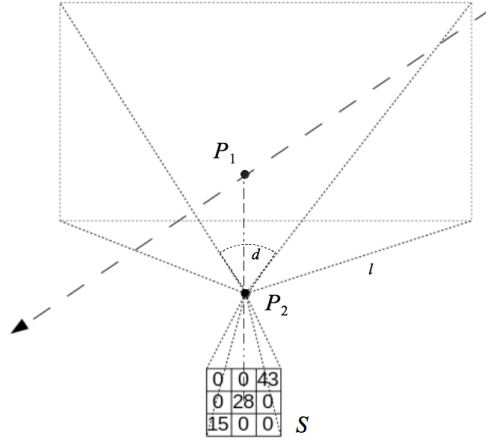
Figure 5: Synthetic retina model in 3d spaces. Here, the particle goes through the view volume, which is projected onto a sensitive plane for activate the corresponding cell in each time step.

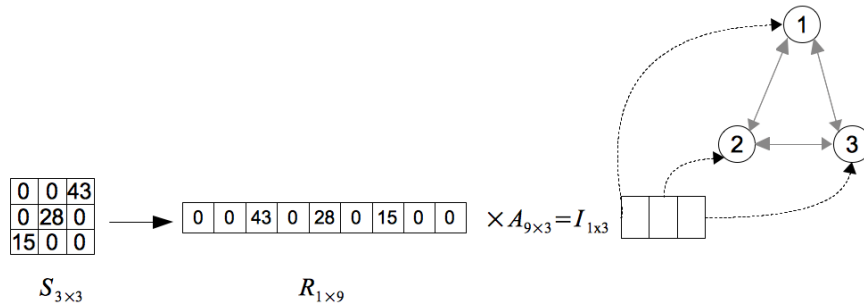phase from a node i is calculated following equation (2).

Figure 6: 3d retinal device. Each sense at a given instant is modulated by a matrix A. The resulting vector I is coupled to each of the oscillators. This factor alters the oscillatory behavior of the network according to environmental interactions.

Body structure: The body structure of our agent is modeled abstractedly as a flying entity which can change its direction of flight in 3d space as a function of two real values that express x and y axis rotation, corresponding to the pitch and yaw rotation angles, respectively. The outputs of the system can then be defined as the sine of the phase differences of two particular oscillators, as shown in equation (11).

21

$$r_x = sin(\theta_1 - \theta_3)$$
$$r_y = sin(\theta_1 - \theta_2)$$

<div align="right">(11)</div>

where θ_i is the phase of the $i-th$ oscillators in a time step t, and r_x and r_y are the rotation angles in the local system of the agent (pitch and yaw rotation), respectively. A phase-sensitivity function is applied to reduce instabilities in the value of the phase differences.

The position of the agent in the environment is characterized geometrically by a direction vector and a position vector. At each iteration the new direction vector is calculated using the equations of rotation of a vector in the space according to the angles r_x and r_y. Subsequently, the new position is calculated by displacing the current position point on the path to the new direction vector, using the parametric equation of the segment.

The genetic encoding for the 3d agent can be represented by the vector ($[\omega_1, \omega_2, \omega_3, A_{9\times3}, k_{3\times3}]$).

3.3   Evolving a dynamic agent

At this point we have a prototype of embodied and situated agents intimately linked to a specific environment. Their parameter values are unspecified. However, we have no clues for providing behavior or specific methods for designing complex systems with many interacting parts. In complex systems modeling, it is more fruitful to focus on the most simple substrate. Thus, a good method is to use an open-ended synthesis system where some parameters of the model are not specified in the design because this is very difficult to do by hand [36][13][37]. Living systems are dynamical systems designed through natural Darwinian

evolution, so it makes sense to consider the artificial equivalent to this design process using evolutionary techniques [38]. Here, the designer plays a passive role, the genetic algorithm is responsible for finding the parameters to execute the behavior. For this it is necessary to model the agent-environment interaction in a parameterized agent and weighting the desired performance on the fitness function [39][38][40].

In our case, genetic algorithms make it easier for us to explore and exploit the complex dynamics of sensory-motor loops in greater depth [41]. Also, partly due to its stochastic nature, evolution can find several different (and equally viable) solutions. When evolution is complete, the agents with positive results are taken and the way in which they solve the task is analyzed. There is nothing preventing the use of evolutionary algorithm for optimizing both the body and the controller of the agent. This allows us to explore the possibilities of the agent in its environment in greater depth and reduces the severity of the problem of selecting an appropriate body structure a priori [27].

According to a computationalist conception, the attitude of an agent towards environmental changes can include a variety of mechanisms that produce adaptive changes in an individual during its lifetime. However, the dynamicist perspective of adaptability changes drastically. In a changing and unpredictable environment, an agent controller cannot always be designed beforehand, but a suitable controller may be evolved depending on the situation of the environment. If a species evolves in a changing environment there will be evolutionary pressure in favor of individuals with the ability to learn during their lifetime. The ability to learn allows individuals to maximize their adaptability to the environment. This is known

23

as Baldwin's effect. Baldwin suggested that any change, feature or custom that lasts long enough also involves a genetic change [42][43].

The last step of our methodological scheme is the evolutionary synthesis process for each task and the simulation of the system behaviour. The following sections outline a series of experiments using the prototypes of autonomous agent described previously in order to show the potential of this framework.

4   Analysis of the results

Our approach consists of producing some diversity and then applying selective pressure to keep the best solutions and discard the others. The design of the fitness function is based on simulating the behavior of the agent in the environment long enough to evaluate its performance according to the desired role. We are thus facing a problem-specific creative process. This should be approached from a heuristic perspective, observing the parameters of the scope of our problem, reasoning and making adjustments according to trial and error.

The first experiment focuses on providing flock behavior to the agent in 2d space. The second experiment expands this to collective behaviors in 3d space. Each experiments is described through two guidelines: *(a)* the evolutionary synthesis process and *(b)* simulation and analysis of the results.

4.1 2d flock model

*Evolutionary synthesis:*The evolutionary mechanism used in this experiment is striking due to its simplicity. The flock behaviour should be weighted on the fitness measure and must be based on quantities that are available in the agent simulation. We have used the average of the deviation of each agent with respect to the average direction of their neighbors. This parameter can be interpreted as a measure of the alignment of the entire flock, tending to 0 when the flock is perfectly aligned. This can be formalized as shown in equation (12).

$$\overline{\delta}(t) = \frac{\sum\limits_{i=1}^{N} \left| \overline{D}(t) - D_i(t) \right|}{N} \tag{12}$$

where $\overline{\delta}$ is the average deviation angle of the flock with respect to the average direction, $\overline{D}$ is the average direction angle of the flock, D_i is the direction angle of the $i - th$ agent (normalized scalar) and N is the number of individuals in the flock. The fitness function was defined as minimizing the average deviation angle of the flock with respect to the average direction in a time period T according to equation (13).

$$min \; \frac{\sum\limits_{t=1}^{T} \overline{\delta}(t)}{T} \tag{13}$$

So, to evaluate each individual in the evolutionary process is to evaluate the quality of a flock where all agents are specified identically for long enough. Each individual is initialized in random initial conditions, synthesizing the adaptability whatever the initial conditions.

The algorithm type used was a simple generational and elitist algorithm that ensures that the best individual passes to the next generation. We use a population size of 80 in a maximum of 100 generations. The tournament selector was used as the parent selector, and uniform crossover as the recombinator type. The recombination probability was fixed at 0.75 and the mutator probability at 0.1, with one locus mutator as the mutator operator. In 80 generations approximately the agents successfully performed the task, obtaining the best fitness value in the order of $[10^1..10^2]$.

Simulation and results: Figure 7 represents different flock simulation states where all agents are configured with the genotype obtained in the evolutionary process, from the initial state when the flock motion is chaotic to the 75 iterations when the agents achieve alignment with their neighbors. At the micro level each agent is capable of rotating in the direction of the neighbours' movement. The macro effect is a flock behavior, seeing as the seemingly intelligent behavior of a flock is restricted to evolve the elementary behavior of an individual. The stimulation of the agents at each instant of time is calculated by averaging the stimulus generated by the flow of each agent that is inside its visual field, as was described in section 3.1.

Looking at the time series of the parameter under optimization (see Figure 8), we can appreciate that from iteration 100 the value tends to 0. This means the flock is perfectly aligned.

Another commonly used flocking parameter that quantitatively illustrates the formation of flocks is the degree of alignment of the flock in terms of the absolute direction versus
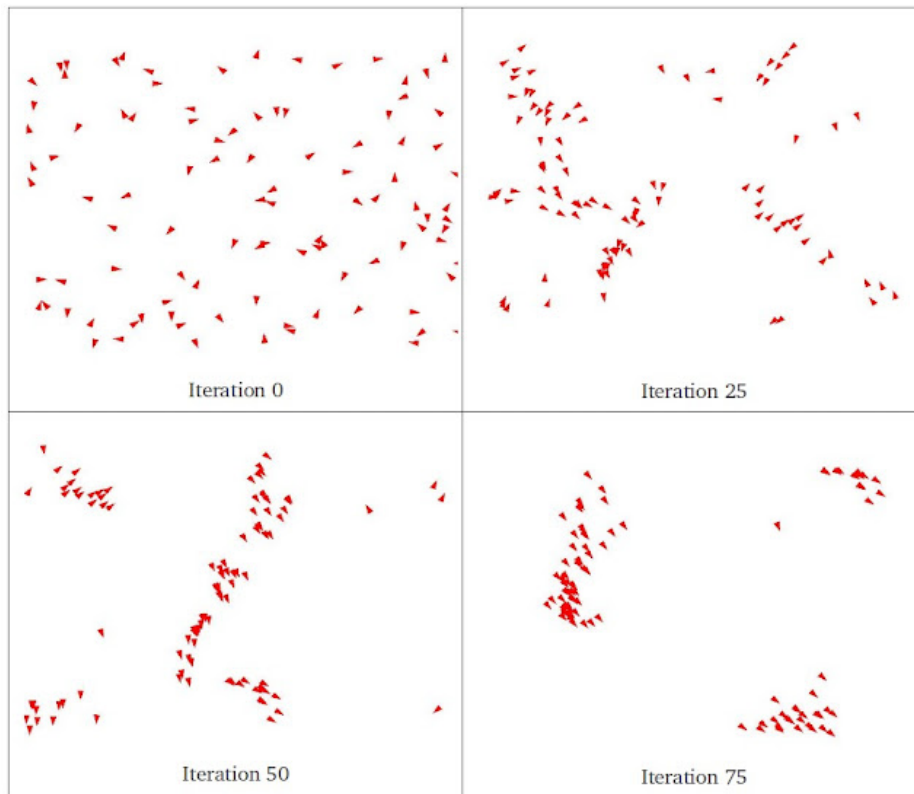
Figure 7: Four states of a flock simulation, where each individual is encoded to rotate in the direction of the optical flow perceived from its neighborhood. Over the 100 iterations the flock is acceptably aligned.

time of each agent, as shown in Figure 9. Note that when the flock organization occurs the steering angle is equal for all agents.

From the point of view of the synchronization dynamic in the controller of each agent, the first thing to remark is that the order parameter of the oscillator network (see equation (3)) of all agents converges to 1 as a measure of phase coherence of the oscillator population, the synchronization emerging as the essence of sensory motor coordination (see Figure 10a). Another interesting point to note is that synchronization occurs independently in each agent. However, the global order parameter including all oscillators of the simulation also
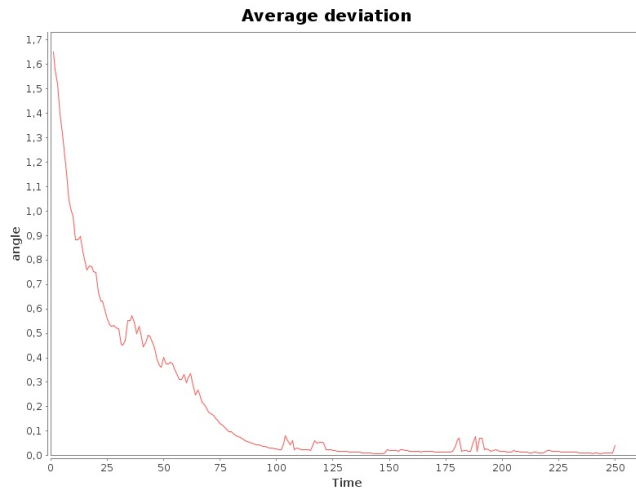
Figure 8: Chart of average deviation of each agent respect to average direction of neighbors versus time. The trend to 0 reveals an aligned flock.

converges to 1 (as shown in Figure 10b). This result suggests that the oscillators of the system are synchronized with each other. Such behaviour occurs because the population of oscillators is connected by means of dynamical interaction in the environment.

4.2 3d flock model

In section 4.1 we described an experiment where an agent uses the movement patterns collected from its 2d environment to perform autonomous collective tasks. The agent was sensitive to the direction of the flow that crossed its field of view, turning in the same direction. Here, we describe a version of this experiment in 3d space using the corresponding synthetic retina model.

Evolutionary synthesis: In the previous experiment a macro-measure of the collective behavior (flock aligning) was used to evaluate the evolution quality. Although it is feasible
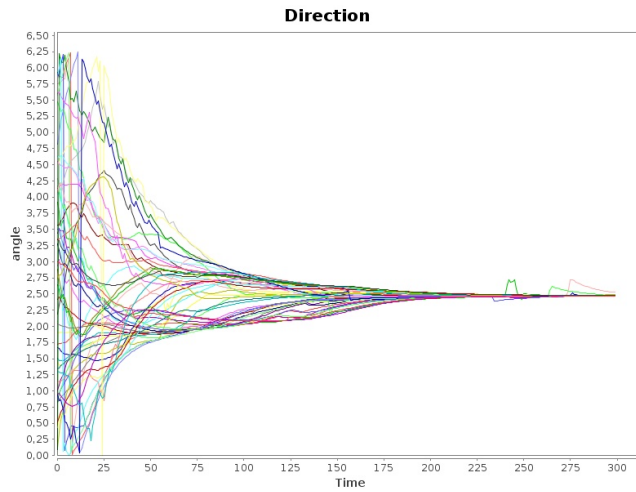
28

Figure 9: Absolute direction versus time in a flock simulation. When the flock organization occurs the steering angle is equal for all agents.

to use the same mechanism for 3d space, we decided to use a micro-magnitude. Like the 2d agent, the individual behaviour of the 3d agent must be able to rotate in the direction of movement of its neighbors. Therefore, the fitness function was designed by placing an agent in a random position and direction in the environment. Its visual field was crossed by a number of particles of random direction and velocity, one behind the other. In this process it can be seen how well the actuators of the agent rotate in the direction of the particle that crosses its visual field. In order to ensure that the agent performs the desired behavior correctly, several particles are simulated in each evaluation to average its performance. The objective function can be formalized as shown in equation (14).

$$min \ \frac{\sum_{i=1}^{N} \tau_i}{N} \tag{14}$$

where N is the number of particles used in the experiment, τ is the angle between vectors
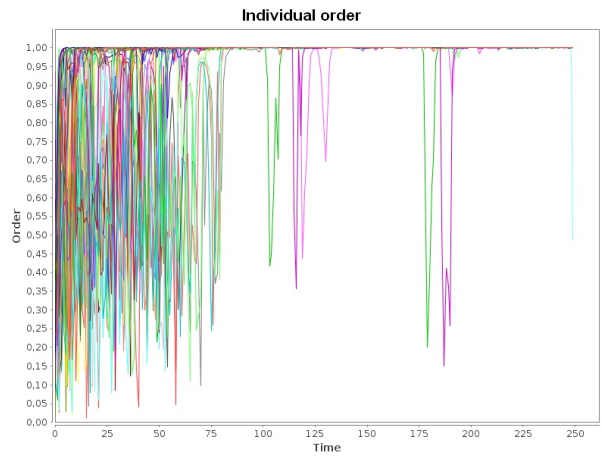
(a)
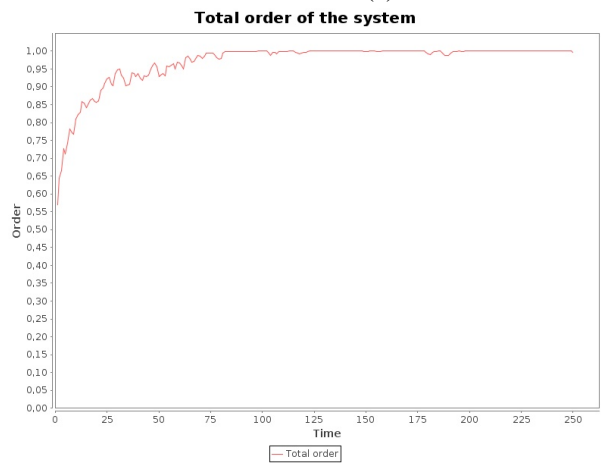
(b)

Figure 10: (a)Order parameter of all oscillator networks of the system. (b) Global order parameter including all oscillators of the system.

$\vec{P}$ and $\vec{A}$, $\vec{P}$ represents the projections in the retinal plane of the vector that describe the flow motion of the $i - th$ particle and $\vec{A}$ represents the projections in the retinal plane of the vector that describe the agent motion in response to this stimulus, as shown in Figure 11.
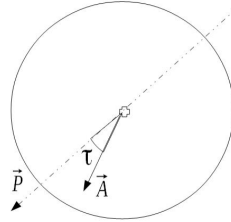

Figure 11: Angle τ, formed by projections in the retinal plane of the vector that describe the flow motion $\vec{P}$ and the projections in the retinal plane of the vector that describe the agent motion $\vec{A}$.

The average of the performance evaluations of the agent with all the particles is the fitness value of the individual. After obtaining an agent able to orient its direction based on the motion flow detected by its sensing device, we try to extend this behavior to a whole flock. Remember that when there are many particles in the visual field, the general optical flow is calculated averaging the stimulus generated by each of the particles independently, thereby each agent basically has a notion of the direction and the speed of movement of its neighbors.

Simulation and results: Figure 12 shows the simulation of a flock whose agents are configured with the best individual obtained by the mechanism described above. The stimulation of each agent is calculated by averaging the stimulus generated by the flow of each neighbor within its visual field. The depth parameter of the 3d retina was fixed at a relatively small value, recreating a limited perception model in an animal flock. Over 600 iterations

31

the agents achieved alignment with their neighbors. The figure shows that the agents are aligned with their close neighbors, giving the sensation of self-organization.

Also, we see that the parameters that quantitatively illustrate the formation of flocks used in the 2d-experiment look similar. In particular, the choreography of the flocks have great similarities with the flocks described by Reynolds [44][45]. Reynold's model is a classic in the simulation of artificial swarms. He described a flock as a general class of polarized, non-colliding, aggregate motion of a group of individuals, based on a set of rules.

4.3 Comparison of the results

In order to make a deeper comparative analysis of our proposal with classical implementations, we compared our model with Reynolds's model from three different perspectives. First, under qualitative criteria; second, measuring quantitatively certain criteria of resemblance of the simulated flocks with natural flocks; and finally, making use of some "information theoretic" measures observed in the temporal dynamics of the simulation, to quantify two concepts inherent to the field of multi-agent organizations, the degree of autonomy and emergence of each of the models.

A qualitative comparison of both models is summarized in the table 1.

A method to estimate the flocking ability is to turn to counting the cumulative number of collisions between agents and to observe the temporal dependency of the number of stragglers and the number of flocks. In order to carry out a quantitative comparison, we also
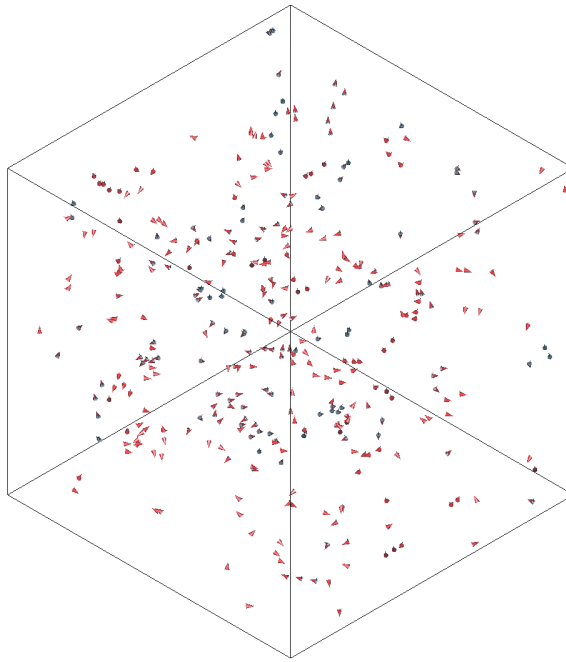
(a)
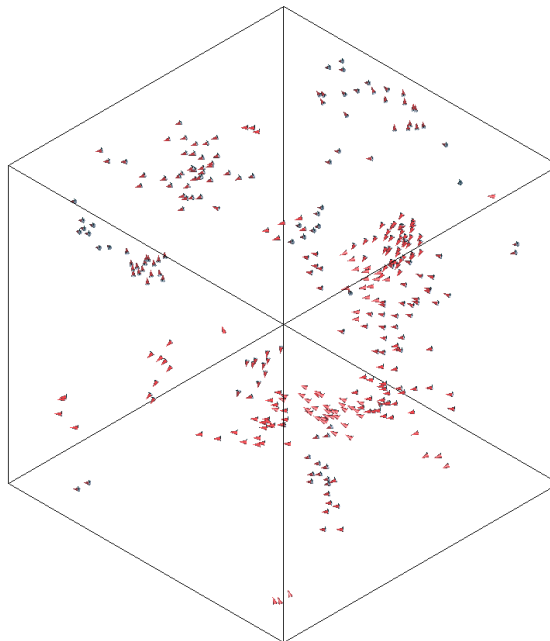
(b)

Figure 12: Two simulation state of a flock of agents configured to rotate in the direction of the perceived optical flow. The second frame shows how the agents are aligned with their close neighbors, giving the sensation of self-organization.

33

	Reynolds's model	Our model
Flocking behavior	Each member is controlled by a relatively simple set of rules that operate locally with symbolic manipulation of the information.	Each member is controlled by a dynamic system resulting from evolutionary algorithms with a sensory-motor coordination approach.
Action selection	The rules base can lead to conflicting behavior. The controller negotiates among the various demands produced as a result of the perception.	The internal network modulates informational variables to action variables in a sensory-motor flow without inference.
Resemblance with natural flocks	Flight formations emerge as well as the presence of somewhat erratic, unsystematic behaviour, strictly dependent on the parameters. Repeating patterns can be detected.	A high sense of dynamism and improvisation.
Perception	Each agent has a perception of center, velocities and direction of the nearby flockmates.	The perception device is an optic array allowing capture flexibly stimulus from the close environment.
Individual parameters	Area of influence; weight of each rule; how fast they can turn, accelerate or decelerate; the field of vision; among others. Values are placed empirically.	Basically the dynamical system parameters. The perception device and the actuators can also be parameterized.
Complexity	The processing complexity is n-squared. Even when interactions are limited to some k nearest neighbors, it is still necessary to find them in the total population.	N-squared complexity because each agent must reason about each of the other agents, even if only to decide to ignore them.
Easy to extend	Extending the rules base from a micro perspective.	Modeling the agent-environment interaction and weighting the desired performance on the fitness function based on micro or macro magnitudes.
Changing environment	The agent should include prior adaptation mechanisms, such as increasing the rule base. Harder to treat in a changing and unpredictable environment.	A suitable controller is evolved depending on the situation of the environment, providing the ability to learn during its lifetime.
Relationship with the environment	The rules base is totally dependent on quantities that can be perceived in the environment.	Agent dynamically coupled to the environment. Only perceives and acts. The functioning of the environment is irrelevant.
System abstraction	The system should be described at a scale relevant to the individual agent's behavior.	The system can be described at individual or collective level.

Table 1: A qualitative comparison of some relevant criteria for Reynolds's model and our model.

use the method followed by Bajek [46][47] [48] based on starting both models (Reynolds's model and our model in 3d space) with identical initial conditions in terms of number of individuals, initial position and direction. Then, some parameters of the simulations are measured during the runs:

- Collisions: when an individual violates the vital area of another individual. The agents are confined using a circular invisible boundary, when an agent crosses this boundary occurs a collision.

- Stragglers: individuals who do not belong to any flock. Stragglers are agent with a

fluctuating neighborhood, they do not manage to join the group throughout the simulation.

- Flocks: number of groups that are generated in the simulation. Each group has specific average direction and speed.

The comparison is based on averaging each of these parameters in several runs for both algorithms, as shown in Table 2. After five runs of experiments, the quantitative data do not show significant differences between the algorithms. Both models behaved very similarly in avoiding collisions. Reynolds's model, summed over the whole population of 50 agents, averaged 2 collisions in 750 simulation steps versus a 1.6 in our model with identical conditions. In both case, collisions occurred randomly throughout the simulation and were mostly head-on collisions, caused by the merging of flocks and were caused in circumstances where there was no scape route. Stragglers in this set of experiments did not persist for long periods of time. At the beginning of the simulation some agent can be isolated but usually doesn't take long before it is attracted by a group. Our model generated a smaller number of flocks, quite possibly due to the difference in the limited perception model. The perception model is determinant in this experiments, in both methods the parameters can be adjusted in many ways with successful results. So, there is a wide range of configurations that resulting in similar behaviour to a greater or lesser extent depending of the values.

We have also estimated the degree of uniform distribution in both models by measuring the nearest neighbour distances (see Figure 13a). We can see that this factor drastically

|	Reynolds's model			Oscillator model		
	collisions	stragglers	flocks	collisions	stragglers	flocks
1	0	3	2	3	2	3
2	2	4	4	0	3	1
3	2	1	3	1	0	2
4	5	1	5	1	1	1
5	1	3	4	3	5	2
avg	2	2.4	3.6	1.6	2.2	1.8

Table 2: A comparison of the number of collisions, number of stragglers and number of flocks after 1000 simulation steps in five experiments for both models.

decreases to the extents that align the flock. Reynolds's model is slightly perturbed, possibly due to the imbalance of the rules of aggregation and repulsion. From the viewpoint of convergence, no significant differences exist. Figure 13b shows the average speed of the flock as the level of alignment. This equally reveals that although in the Reynolds simulation the speed changes abruptly due to the polarization rules, there are no significant differences between the two algorithms.

The ability to measure a phenomenon is an essential step toward its effective scientific description. Recent studies in complexity have generated a variety of models that address particular aspects of the behavior of complex systems. These models have made significant contributions to our understanding of the notion of organization [49]. Some "information theoretic" measures have been used to quantify the degree of constraint inherent in the organization of each model. Concepts of emergence and autonomy are recurrent in artificial life and related cognitive and behavioral sciences. Both have been used as quantitative measures based on the framework of multivariate autoregression and, specifically, Granger causality [50].
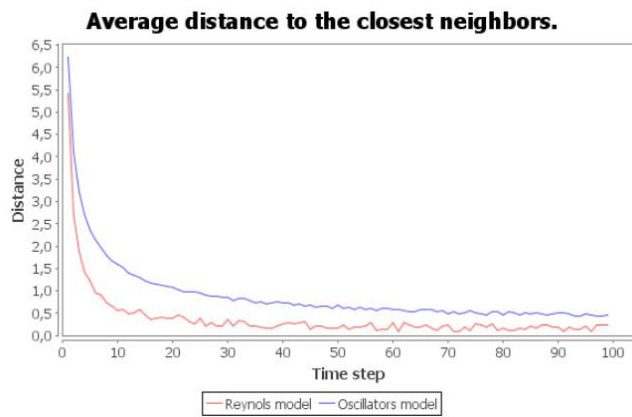
(a)
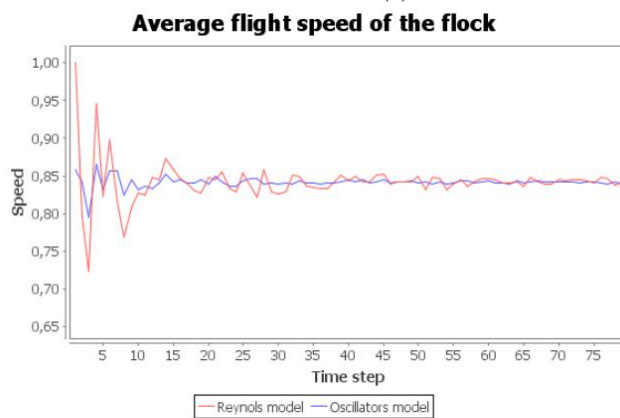
(b)

Figure 13: (a) Comparison of the nearest neighbour distances and (b) the average flight speed of the flock.

Granger causality is a statistical interpretation of causality according to which a signal A causes a signal B if information in the past of A helps predict the future of B, over and above predictions based on the past of B alone [51][52]. To illustrate G-causality, suppose that the temporal dynamics of two time series $X_1(t)$ and $X_2(t)$ (both of length T) can be described by a bivariate autoregressive model as shown in equation (15).

$$
\begin{aligned}
X_1(t) &= \sum_{j=1}^{p} A_{11,j} X_1(t-j) + \sum_{j=1}^{p} A_{12,j} X_2(t-j) + \xi_1(t), \\
X_2(t) &= \sum_{j=1}^{p} A_{21,j} X_1(t-j) + \sum_{j=1}^{p} A_{22,j} X_2(t-j) + \xi_2(t)
\end{aligned}
\tag{15}
$$

where p is the maximum number of lagged observations included in the model, A contains the coefficients of the model, ξ_1 and ξ_2, are the residuals (prediction errors) for each time series. If the variance of ξ_1 is reduced by the inclusion of the X_2 terms in the first equation, then it is said that X_2 G-causes X_1. Assuming that X_1 and X_2 satisfy covariance stationarity, the magnitude of this interaction can be measured by the log ratio of the prediction error variances for the restricted (R) and unrestricted (U) models, see equation (16).

$$
gc_{2 \rightarrow 1} = log \left(\frac{var(\xi_{1R(12)})}{var(\xi_{1U})} \right)
\tag{16}
$$

where $\xi_{1R(12)}$ is derived from the model omitting the $A_{12,j}$(for all j) coefficients in the first equation, and ξ_{1U} is derived from the full model. Importantly, G-causality is easy to generalize to the multivariate (conditional) case in which the G-causality of X_1 is tested in

the context of multiple variables $X_2, , X_N$ ($X_i \neq X_j$ for all X_{ij}). In this case, X_2 G-causes X_1 if knowing X_2 reduces the variance in X_1's prediction error when the activities of all other variables $X_3, , X_N$ are also included in the regression model.

We adopt a simple conception of autonomy as the degree of self-determination of a system. A variable is autonomous to the extent that it helps predict its own future, as compared to predictions based on past states of a set of external variables. An autonomous system should not be fully determined by its environment. Having a practically applicable measure of autonomy is useful for gaining insight into mechanisms underlying apparently autonomous behavior, as well as into selective pressures that can lead to increases or decreases in autonomy.

The G-causality provides a means of operationalizing autonomy as self-determination, or self-causation. The G-autonomy measure asks whether the prediction error of X_1 is reduced by inclusion of its own past, given a set of external variables $X_{2,...,N}$. Put simply, a variable is G-autonomous to the extent that it is dependent on its own history and that these dependencies are not accounted for by external factors.

Recalling equation (15), X_1 is G-autonomous if the coefficients in A_{11} are jointly significantly different from zero. This can be tested by performing an F-test of the null hypothesis that $A_{11} = 0$, given assumptions of covariance stationarity for X_1 and X_2. By analogy with G-causality, the G-autonomy of X_1 with respect to X_2 is given by equation (17).

$$ga_{X_1|X_2} = log \left(\frac{var(\xi_{1R(11)})}{var(\xi_{1U})} \right) \tag{17}$$

where $\xi_{1R(11)}$ is derived from the model omitting the coefficients $A_{11,j}$ (for all j) in the Granger equations.

G-autonomy can be validated by its application to time series that have different self-causation characteristics by construction. In our test, three experimental conditions are compared:

1. Agents move randomly

2. Agent behavior is determined by Reynolds rules.

3. Agent behavior is controlled by an evolved network of oscillators.

Each condition of the model was run for 10 trials of $2,000$ time steps each (following the evolution in condition 3). From each trial, 20 time series were generated, corresponding to the trajectories of the agents. To prevent edge effects, each time series $(a_1, , a_{20})$ consisted of the average of the horizontal and vertical displacement from the middle of the environment. Each time series was then first-order differenced [i.e., $a(t) \to a(t) - a(t-1)$] in order to ensure covariance stationarity. Each resulting data set was used to construct a multivariate autoregressive model of order $p = 4$. Each model was then used to calculate the G-autonomy values (see equation (17)). G-autonomies were averaged across the 10 trials in each condition, as shown in Figure 14.
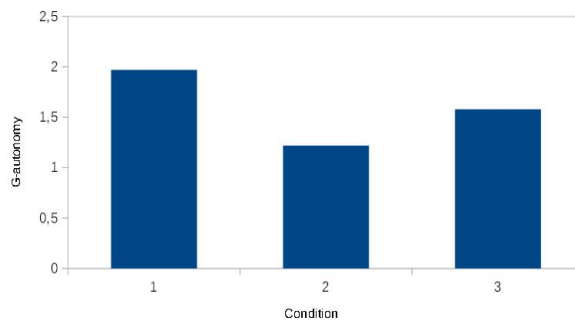
Figure 14: Agents G-autonomy (averaged across the 10 trials) for each condition. All comparisons are statistically significant (two-tailed t-test, $p < 0.001$). (Condition 1-Agents move randomly, 2-Agent behavior is determined by Reynolds rules, 3-Agent behavior is controlled by an evolved network of oscillators.)

We can see that random movement (condition 1) produces high G-autonomy, because random movement in this model is not a white noise process; rather, it implies that the agent changes direction unpredictably, so that the best predictors of future agent positions are recent past positions. The Reynolds model (condition 2) produces medium G-autonomy, because the agent position is now well predicted by the past positions of a flock. However, the G-autonomy value in our model (condition 3) is higher than in condition 2 because in this condition it is driven both by the position of nearby agents and by the internal state of the oscillators network, also suggesting that evolutionary adaptation in the model leads to increased G-autonomy. A possible explanation for this is that the evolved controller has the opportunity to take the internal state into account at every time step. G-autonomy in condition 3 remains lower than in condition 1 because agents' positions can still be partially predicted by flock position. So, the G-autonomy measure quantifies and amplifies the concept of autonomy as self-determination.

On the other hand, an emergent property is in a qualitative sense "more than the sum" of its component parts. G-emergence is a statistical measure of the extent to which an emergent process is both autonomous from and dependent upon its underlying causal factors [50]. G-emergence is a continuous version of weak emergence, in which a macro property is weakly emergent to the extent that it is not deducible from micro-level observations. G-causality and G-autonomy together provide the necessary ingredients for operationalizing G-emergence. This measure captures the basic intuition about weak emergence that involves dependence on underlying processes, and that at the same time involves autonomy from underlying processes. According to G-emergence, a macro variable M is emergent from a set of micro variables m if and only if:

i) M is G-autonomous with respect to m.

ii) M is G-caused by m.

A simple measure for the G-emergence of M from m is therefore given by equation (18).

$$ge_{M|m} = ga_{M|m} \left(\frac{1}{N} \sum_{i=1}^{N} gc_{m_i \rightarrow M} \right) \tag{18}$$

This value will be zero either if M is independent of m or if M is fully predicted by m. There are other more sophisticated methods that detect nonlinear dependencies and which can be less sensitive to noise. However, for the present purposes this method is

preferable because it is simple to describe and to implement, statistical significance can easily be assessed, and it supplies an explicit formula for G-emergence.

In both models (Reynolds's model and our model), the flock seems to have a shape and trajectory of its own, which transcends those of the individual agents. Here, a simple agent simulation is used to test whether visually compelling flocking correlates with high G-emergence of the center of mass of the flock (the macro variable) with respect to the trajectories of the individual agents (the micro variables).

The objective is to measure the G-emergence of the center of mass of a flock with respect to the individual agents. $N = 20$ agents are simulated in a toroidal square environment. The agents are randomly initialized with positions and velocities.

Three different conditions were tested:

1. Agents move randomly

2. Agent behavior is determined by Reynolds rules.

3. Agent behavior is controlled by an evolved network of oscillators.

For each condition the simulation was run 25 times with each run lasting 5000 time steps. For each run the x, y coordinates of each agent and the global center of mass were recorded. Several preprocessing steps were carried out prior to calculation of the G-emergence. Each coordinate pair was transformed into a single variable reflecting distance from the center of the environment. The first 50 data points were removed to eliminate initial transients,

and each resulting time series was transformed into its zero-mean equivalent. Finally, each time series was first-order differenced in order to ensure covariance stationarity. Following preprocessing, for each run in each condition the G-emergence of the center of mass was computed using a model order $p = 5$.

Figure 15 shows the mean G-emergence of the center of mass in each condition. All values of G-emergence in conditions 2 and 3 were significant ($P < 10^{-5}$ for G-autonomy and G-causality, two-tailed t-test); those in condition 1 were not.
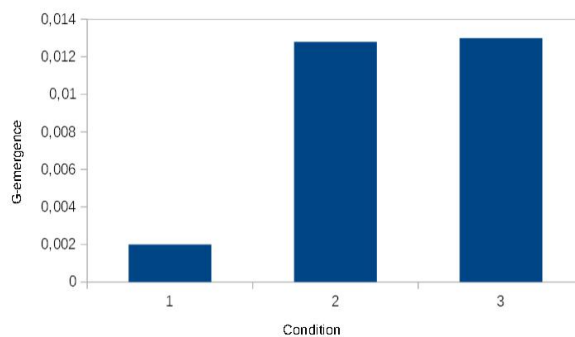

Figure 15: G-emergence of the center of mass of a flock for all conditions. (Condition 1-Agents move randomly, 2-Agent behavior is determined by Reynolds rules, 3-Agent behavior is controlled by an evolved network of oscillators.))

This confirms the prediction of a high G-emergence value for compelling flocking. There is no statistical difference between the two methods. The macro property is highly emergent to the extent that it is not deducible from micro-level observations. This measure is based on a statistical interpretation of causality. It sidesteps conceptual pitfalls such as competition among micro and macro causes, and it provides an objective and graded assessment of the non-triviality of micro-to-macro inferential pathways.

5 Discussion and future work

Our approach is inspired by Gibson's concept that there is sufficient information available in the agent-environment interaction to control the agent's behavior without further inference or reconstruction. We try to demonstrate that the perception-action dynamics in an agent provide a non-inferential source of information which can deal with problems for which other computation techniques (planning, mapping, reasoning, etc.) can be limited.

We have presented an experimental simulation framework capable of modeling behaviors of embodied and situated agents whose control architecture is based on oscillator networks. The internal network modulates the stimulus signals to action signals under a sensory-motor flow. The perception device of the agent is designed as an optic array emulating the principles of motion perception in the animal retina. We use an intuitive design methodology where the model is designed in abstract dynamic terms thanks to the evolutionary approach, allowing agents to be designed without structural components predefined in the design process. The behavior emerges thanks to three factors: suitable devices of perception and action for the environment, a controller perfectly coupled to the sensory-motor loop capable of generating coordination dynamics for the problem, and a fitness function that synthesizes the complexity and diversity of agent-environment interaction that establishes selective pressure in the evolution process of the agent.

Our model is applied in two flocking experiments in 2d and 3d environments. We carry out a comparative analysis of our proposal with Reynolds's model under qualitative, quan-

titative and information-theoretic points of view. In the qualitative aspect our model stands out above classical implementations by its high sense of dynamism and improvisation in the simulation. Our optic array model allows stimulus to be flexibly captured from the environment compared to a perception closely linked to specific behavior in the Reynolds model. The differential equations system can capture the agent's behavior with low dimensional descriptions of each individual, and the system can be described at a scale relevant to both individual behavior and collective behavior, showing a high scalability capacity. Our agent is dynamically coupled to the environment (it perceives and acts), as opposed to a rules-based model which depends entirely on the problem-environment. Our design methodology is effective when we have no clues to provide behavior. The keys of our approach are the agent-environment interaction and the fitness function. Once this has been achieved, the genetic algorithm is responsible for finding the parameters to execute the behavior. Also, our model is marked by an inherent adaptability. The individuals evolve in a changing environment with an evolutionary pressure in favor of agents with the ability to learn during their lifetimes.

A quantitative measure of some parameters of resemblance with natural flocks in identical initial conditions for both models shows a great similarity in the simulations.

Some statistical measures have been used to quantify the degree of emergence and autonomy of our model versus Reynolds's model. The G-autonomy value in our model is higher than the rules-based model because the evolved controller has the opportunity to take the internal state into account at every time step besides the external conditions. Our

agent position is partially predicted by flock position rather than being totally predicted as in the Reynolds's model. However, there is no statistical difference in G-emergence between the two methods, both showing high values. In both models, the macro property is highly emergent to the extent that it is not deducible from micro-level observations.

On the other hand, our approach allows full integration of environmental, corporal and neural factors as well as its complex dynamic interaction within the same explanation framework. It is therefore possible to analyze patterns of control activity, variability of actuators and sensors of the system in a coupled way. Thus, we can see that:

- The system exhibits circular causation in a negative feedback loop closed through the environment. This contradicts the classical notion of linear causation of behavior by stimuli, in which environmental stimuli are thought to cause behavioral responses, mediated by intervening cognitive processes.

- The negative feedback that occurs in agent-environment interaction gives our agent certain properties such as self-regulation and stability, reducing the effect of fluctuations. Observing its internal variables, we can see patterns of stability, accuracy, sensitivity and a reduction of the effects of unpredictable influences.

- The results of the experiments have demonstrated that our agent controls neither its own behavior nor external environmental variables, but rather its own perceptions of those variables. Actions are not controlled, they are varied so as to cancel out the effects of unpredictable environmental disturbances. So, "behavior is the control of

47

perception".

Our model does not target any specific biological mechanism; it only shows the potential of oscillatory controllers within a continuous time-closed sensory-motor loop for complex adaptive tasks. Our approach provides new ways of thinking about behavior at another level of abstraction or from a different paradigmatic perspective. The dynamicist focus has helped us to replace the notion of symbolic representation and processing by modeling in terms of dynamical coupling without the need of an explicit encoding on the side of the agent as a more appropriate metaphor for defining adaptive behavior in virtual agents.

Our future research priority will be to extend the model to perform more complex tasks of locomotory behavior and perception in changing environments. Also, we are interested in the perception of patterns, shapes and sizes from the environment. Another important issue is the study of dynamic patterns of control activity and the variability of actuators and sensors.

Acknowledgement

References

[1] Rick Parent. *Computer Animation: Algorithms and Techniques (Google eBook)*, volume 2007. Morgan Kaufmann, 2007.

[2] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. *Virtual Crowds: Methods, Simulation, and Control*, volume 3 of *Synthesis Lectures on Computer Graphics and Animation*. Morgan & Claypool Publishers, 2008.

[3] Adriana Braun, Bardo E J Bodmann, and Soraia R Musse. Simulating virtual crowds in emergency situations. *Proceedings of the ACM symposium on Virtual reality software and technology VRST 05*, page 244, 2005.

[4] Stewart W Wilson. Knowledge Growth in an Artificial Animal, 1985.

[5] R J Douglas, K A Martin, and J C Nelson. The neurobiology of primate vision. *Baillieres Clinical Neurology*, 2(2):191–225, 1993.

[6] J J Gibson. *The perception of the visual world*, volume 98. Houghton Mifflin, 1950.

[7] J J Gibson. Visually controlled locomotion and visual orientation in animals. *British journal of psychology London England 1953*, 49(3):182–194, 1958.

[8] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2):141–160, June 2009.

[9] Steven S Beauchemin and John L Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[10] K.R.T. Aires, A.M. Santana, and A.A.D. Medeiros. Optical flow using color information: preliminary results. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1607–1611. ACM, 2008.

[11] Mubbasir Kapadia, Shawn Singh, William Hewlett, Glenn Reinman, and Petros Faloutsos. Parallelized egocentric fields for autonomous navigation. *The Visual Computer*, 28(12):1209–1227, 2012.

[12] David Vernon, Giorgio Metta, and Giulio Sandini. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *IEEE Transactions on Evolutionary Computation*, 11(2):151–180, April 2007.

[13] Esther Thelen and Linda B Smith. *A dynamic systems approach to the development of cognition and action*, volume 512. MIT Press, 1994.

[14] S. S. Robertson, A. H. Cohen, and G. Mayer-Kress. *Behavioural chaos: Beyond the metaphor* . Cambridge: MIT Press., 1993.

[15] Tim Van Gelder. What Might Cognition Be, If Not Computation? *Journal of Philosophy*, 92(7):345–381, 1995.

[16] Timothy Van Gelder and Robert F Port. It's about time: An overview of the dynamical approach to cognition. In T Van Gelder and R Port, editors, *Mind as Motion Explorations in the Dynamics of Cognition*, volume Mind as mo of *Bradford Books*, pages 1–43. MIT Press, 1995.

[17] Randall D Beer. Dynamical Systems and Embedded Cognition. In K Frankish and W Ramsey, editors, *The Cambridge Handbook of Artificial Intelligence*, chapter 12. Cambridge University Press, 2011.

[18] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[19] Matthew Schuerman, Shawn Singh, Mubbasir Kapadia, and Petros Faloutsos. Situation agents: agent-based externalized steering logic. *Computer Animation and Virtual Worlds*, 21(3-4):267–276, 2010.

[20] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *ACM Transactions on Graphics (TOG)*, 25(3):1160–1168, 2006.

[21] Sébastien Paris, Julien Pettré, and Stéphane Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, volume 26, pages 665–674. Wiley Online Library, 2007.

[22] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1928–1935. IEEE, 2008.

[23] Wei Shao and Demetri Terzopoulos. Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 19–28. ACM, 2005.

[24] Glen Berseth, Mubbasir Kapadia, Brandon Haworth, and Petros Faloutsos. Steer-fit: Automated parameter fitting for steering algorithms. *Eurographics Association, Copenhagen, Denmark, V. Koltun and E. Sifakis, Eds*, 2014.

[25] Christopher Hartman and Bedrich Benes. Autonomous boids. *Computer Animation and Virtual Worlds*, 17(3-4):199–206, 2006.

[26] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420, 2007.

[27] Renan C Moioli, Patricia A Vargas, and Phil Husbands. Exploring the Kuramoto model of coupled oscillators in minimally cognitive evolutionary robotics tasks. *Framework*, 2010.

[28] Juan Acebrón, L. Bonilla, Conrad Pérez Vicente, Félix Ritort, and Renato Spigler. The Kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1):137–185, April 2005.

[29] Ziya Kalay. A Review On Synchronization Through The Kuramoto Model. *Physics*, 87131(2):1–8, 2007.

[30] S Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1-4):1–20, September 2000.

[31] Xabier E Barandiaran and Ezequiel Di Paolo. Artificial Mental Life. *Artificial Life*, 11:1994–1994, 2008.

[32] Arkalgud Ramaprasad. On the definition of feedback. *Systems Research and Behavioral Science*, 28(1):4–13, 1983.

[33] Randall D Beer. The Dynamics of Active Categorical Perception in an Evolved Model Agent. *Adaptive Behavior*, 11(4):209–243, 2003.

[34] Fred Adams. Embodied cognition. *Phenomenology and the Cognitive Sciences*, 9(4):619–628, 2010.

[35] Bruno A Santos, Xabier E Barandiaran, and Phil Husbands. Metastable Dynamical Regimes in an Oscillatory Network Modulated by an Agents Sensorimotor Loop. *Rain*, 2011.

[36] Rodney A Brooks. Intelligence without representation. *hilosophical Transactions of the Royal Society - Series A: Mathematical, Physical and Engineering Sciences*, 47(1811):139–159, 1991.

[37] W J Freeman, L J Rogers, M D Holmes, and D L Silbergeld. Spatial spectral analysis of human electrocorticograms including the alpha and gamma bands. *Journal of Neuroscience Methods*, 95(2):111–121, 2000.

[38] Inman Harvey, Ezequiel Di Paolo, Rachel Wood, Matt Quinn, and Elio Tuci. Evolutionary Robotics: A New Scientific Tool for Studying Cognition. *Artificial Life*, 11(1-2):79–98, 2005.

[39] György Buzsáki. *Rhythms of the Brain*, volume 54. Oxford University Press, 2006.

[40] William Bechtel. Representations and cognitive explanations: Assessing the dynamicist's challenge in cognitive science. *Cognitive Science*, 22(3):295–318, 1998.

[41] David E Goldberg and John H Holland. Genetic Algorithms and Machine Learning. *Machine Learning*, 3(2):95–99, 1988.

[42] J Mark Baldwin. A new factor in evolution. *The American Naturalist*, 30(354):441–451, 1896.

[43] Geoffrey E Hinton and Steven J Nowlan. How learning can guide evolution. *Complex Systems*, 1(1):495–502, 1987.

[44] Craig W Reynolds. Computer animation with scripts and actors. *ACM SIGGRAPH Computer Graphics*, 16(3):289–296, July 1982.

[45] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, 1987.

[46] Iztok Lebar Bajec, Miha Mraz, Nikolaj Zimic, and Information Science. Boids with a fuzzy way of thinking. *Science*, 25, 2003.

[47] Iztok Lebar Bajec, Nikolaj Zimic, and Miha Mraz. Fuzzifying the Thoughts of Animats. *Knowledge Creation Diffusion Utilization*, pages 195–202, 2003.

[48] Iztok Lebar Bajec. *Fuzzy Model for a Computer Simulation of Bird Flocking*. PhD thesis, University of Ljubljana, 2005.

[49] James P. Crutchfield. The calculi of emergence: computation, dynamics and induction, 1994.

[50] Anil K Seth. Measuring autonomy and emergence via Granger causality. *Artificial life*, 16(2):179–196, 2010.

[51] Anil Seth. Granger causality, 2007.

[52] Adam B Barrett, Lionel Barnett, and Anil K Seth. Multivariate Granger causality and generalized variance. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 81(4 Pt 1):041907, 2010.