Emergent Navigation Behavior of Autonomous Virtual Agents Endowed with an Evolved Visual Cortex

Lílian de O. Carneiro[1], Joaquim B. Cavalcante-Neto[1], Creto A. Vidal[1], Yuri Lenon B. Nogueira[2], Arnaldo B. Vila Nova[1]
[1]Department of Computing, Federal University of Ceará, Fortaleza, CE, Brazil
Emails: {lilian, joaquimb, cvidal, barretovilanova}@lia.ufc.br
[2] Unichristus, Fortaleza, CE, Brazil
Email:yurilenon@gmail.com

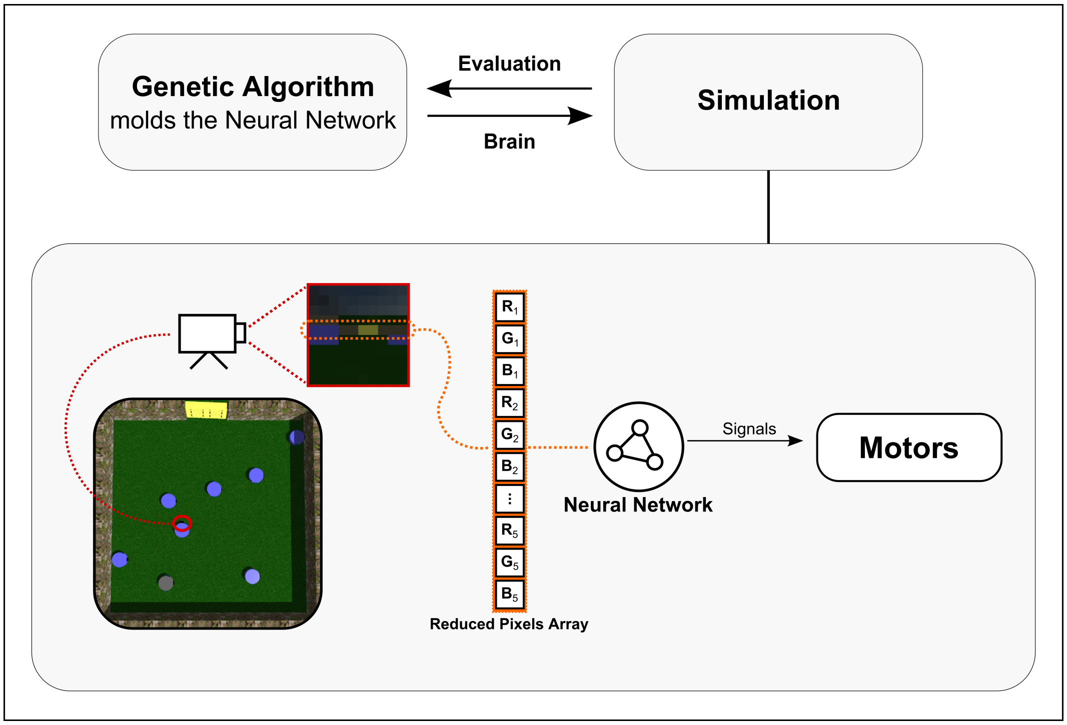

Fig. 1. Overview of the proposed model. Agent is able to move at the environment using its own vision. The proposed artificial visual cortex consists in images captured by a virtual camera. Images of 10×10 pixels in the RGB system are captured and passed to the controller. Since the agent moves only in two dimensions, it is sufficient to see the central part of the image. Therefore, only the ten pixels from the fifth row are used as vision. These, on the other hand, are reduced to five pixels using the arithmetic mean of RGB components from each two neighbor pixels.

***Abstract*—In this work, we propose a technique for generation of emergent navigation behavior in autonomous agents which are able to move at the environment using their own vision. In order to achieve this, we apply the Continuous Time Recurrent Artificial Neural Network and the genetic encoding proposed in [1] and [2]. However, we use a new sensorial description, which consists of captured images by a virtual camera, evolving an artificial visual cortex. The experiments show that the agents are able to navigate at the environment and to find the exit, in a non-programmed way and whithout requiring agent's reprogrammation, using only the visual data passed to the neural network. This technique has the flexibility of being applied in various environments, without displaying a biased forced behavior as a result of a behavioral modeling, as in other techniques.**

***Keywords*—Autonomous Agents, Genetic Algorithm, Artificial Visual Cortex.**

I. Introduction

COMPUTER aided simulation of crowd evacuation allows detailed study, without risk of death, of the behavior of a crowd during the evacuation of an environment in an emergency situation. For this reason, this type of simulation has been used by researchers interested in reproducing the global behavior of crowds as accurately as possible [3] [4]. In this type of simulation, it is possible, for example, to specify different scenarios in different situations, with a large number of people and environmental characteristics. This allows the study of complex behaviors associated to the way interactions occur among individuals of the crowd and between individuals and the environment.

Generation of natural behaviors is an open research topic and it has attracted the attention of many researchers over the years, such as [5], [6], [7], [8] and others. However, many of the proposed approaches for behavior generation require a detailed description of the simulation environment and perception maps to provide relevant information to the agents, which limits their behaviors to those expected by the programmer. Thus, since the behavior is mainly the result of the internal dynamics of the agent's mind, what actually happens in the environment may not influence their reactions, and, thus, its behavior may not be consistent with the faced situation. Furthermore, since such cognitive model defines the composition of the behavior, the resulting behaviors are very well-conducted, such as very organized queues or sudden changes in movement, losing naturalness [2]. So, researchers are increasingly seeking more natural and realistic autonomous behaviors, independent of the environment or preestablished rules.

In crowd evacuation, considering both normal and emergency situations, an agent generally has the specific goal of reaching a certain location of the simulated environment. Thus, it is important to study the dynamics of the agent movement, as well as its choice of route. According to Thalmann and Musse [9], the navigation is probably the most crucial behavior for crowds that can be simulated with a computer. Therefore, in this paper, we present a study of the movement dynamics for an agent in a variety of simulation environments. In order to achieve this, in the study of evacuations, we use the approach of behavior generation for autonomous virtual characters proposed in [1] and [2]. However, we present a new sensorial description in which we propose the evolution of an artificial visual cortex based on images captured by a virtual camera.

According to Kapadia and Badler [10], navigation in large and complex environments requires agents equipped with a mental model providing a semantically meaningful geometric representation of the environment around them. Although our proposed application simulates simple environments, the results show that an agent is able to navigate through an environment without any environmental information, using its own eyes, and find the exit in a non-programmed way. Thus, it is considered highly relevant to the crowd evacuation simulation, an approach that is able to: generate non-modeled behaviors, compatible with the situation occurring in the virtual environment; provide the possibility of exploration of new situations without the need of any cognitive modeling; and explore new environments whithout requiring agent reprogrammation.

The main contributions of this work are the emergence of navigation behavior without any kind of behavior description and the evolution of an artificial visual cortex based on images captured by a virtual camera that allows the agent to navigate through the environment using only its own vision.

The remainder of this paper is organized as follows. In Section II, we discuss the most relevant related work. In Section III, we present our model. In Section IV, we present the tests that were devised to demonstrate the usefulness of our technique, and discuss the respective results. Finally, in Section V, we draw some conclusions.

II. Related work

The coordination of the movement of groups of agents plays an important role to simulate swarms of robots, animals, and pedestrians in computer graphics and civil engineering applications [11]. Kapadia and Badler [10] argue that, to navigate in a complex environment, some semantically meaningful geometric representation of the environment is essential. Therefore, the agent needs a mental map of the environment that surrounds it and should be able to understand and use this map to achieve its goal in the best possible way.

Most related work try to provide the interpretation of what the agent perceives and the rules the agent have to follow to choose its actions. However, by predetermining the agent actions with specific rules, the movements generated are robotized, unlike real people. The ideal autonomous agent should be able to use its own perception to identify what is happening in the environment and make its own decisions by interpreting the elements that are identified through its perception.

Among the most popular techniques for crowd navigation, Pelechano et al. [11] highlights cell and graphs portal, potential fields and roadmaps. In addition to those techniques, we can also highlight those using the discrete approach, that are simple to implement, and possibly for this reason, they are among the most popular techniques. The basic idea is to use a discrete representation of the simulation environment for planning navigation.

An example of a discrete approach is Cellular Automata (CA). A cellular automaton consists of a regular uniform lattice. The state of a CA is completely specified by the values of the variables at each cell. A CA evolves in discrete time steps, with the value of the variable at one cell being affected by the values of variables at the neighboring ones [11]. The variables at each cell are updated simultaneously based on the values of the variables in their neighborhood at the previous time step and according to a set of local rules [12]. Those rules describe the (intelligent) decision-making behavior of the automata, thus creating and emulating actual behavior. Each automaton evaluates its opportunities on a case-by-case basis. Global emergent group behavior is a result of the interactions of the local rules as each pedestrian examines the available cells of its neighborhood [11]. Among the works that use the CA approach, we can refer to [13], [14] and [15].

Another approach that has been widely used is Probabilistic Roadmaps (PRMs), especially in robotics and autonomous navigation for agents. PRMs compute a simplified representation of free space by sampling configurations at random [10]. Then the sampled configurations are tested for collision, and each collision-free configuration is retained as a “milestone”. Each milestone is linked by straight paths to its k-nearest neighbors. Finally, the collision-free links will form the PRM [11]. PRMs have been used to generate navigation paths for large groups of agents simulating group behaviors such as homing, flocking [16] and shepherding [16], [17].

Cell and Portal Graphs (CPGs) were introduced by Teller [18] and represent a method of abstracting the geometry of virtual environments. In CPGs, navigation becomes a problem of getting from one node of the graph to another through a sequence of nodes and portals [11]. When employed to indoor scenes, nodes usually represent the rooms defined by their enclosing walls, and portals correspond to the doors. Outdoor environments can also be represented with CPGs where cells are pedestrian pathways and portals appear between pedestrian pathways and crossings [11]. Among the works that use this approach, we highlight [19] and [20].

In the approach based on potential field, the environment is usually discretized into a regular grid, where each cell is associated with a potential value, which is the sum of a repulsive potential generated by obstacles in the environment with an attractive potential generated by the goal. Therefore, gradient methods can be applied to find a path between any position of the environment and the goal position. Dynamic potential fields have been used to integrate global navigation with moving obstacles and people, efficiently solving the movement of large crowds whithout the need for explicit collision avoidance [21]. The works [21], [22] and [23] use potential fields.

In the approaches presented here, the agents need to receive information about the environment so they can move through it and reach their goals. These informations are transmitted to the agents by: values that indicate the shortest path to the destination setted to each cell in the discrete approach; environmental representation in a graph, in the roadmaps and cell and graphs portal approaches; or potential fields that guide the agents to their goals, in the potential field approach. The need of an environment description limits the behaviors of the agents to those provided in a fixed world, often resulting in a set of predetermined rules for a given context. Therefore, if the agent is exposed to different situations than those for which it was scheduled to face, the behaviors may not be compatible with what really happens in the environment. As a result, we obtain agents with robotized movements when compared to real people. The expected behavior would be the agent to use its perception to identify what is happening in the world around it and, based on that perception, it should make decisions, as in fact occurs in the real world. For example, CA models do not allow for contact between agents [11], generating unrealistic results in situations of high crowd density.

Seeking virtual characters with more natural and realistic behaviors or greater compatibility between the characteristics of the characters and the environments they populate, Nogueira et al. [1] [2] applied techniques inspired by the Embodied and Enactive Artificial Intelligence in order to generate emergent behaviors of virtual characters. The authors define an emergent behavior as one that is not explicitly described in any of the elements of the system, emerging as a result of the dynamic interactions of the components as well as its individual specific properties [1]. In [2], the authors describe a system based on a plastic neuromodulated network, capable of modulating itself to received stimuli, and a simple method for genetic encoding of the neural network controller. In order to test the controller, the authors used it to control a robot in a foraging task, i.e., in a search and exploitation task of food resources. As a result, the agent was able to use its own vision to achieve the goal of collecting fruit, even with the lack of sensorial data, showing a complex foraging behavior.

Although the obtained behavior is relatively simple when compared to those obtained using traditional techniques, we believe that the quest for more complex behaviors through this new approach must be pursued, so that the technique can be enhanced and lead to increasingly more realistic behaviors. For this reason, the work of Nogueira et al. [2] was used as the basis for the development of this research, in which the navigation behaviors are obtained by a technique that generates emergent behaviors, without any description of the environment or target of the pedestrian. Our technique, whose details are given in Section III, allows the application of the same controller to different simulated situations, without the need for reprogramming the character. Furthermore, generated behaviors do not reflect modeling flaws.

III. Technical Description

A. The Controller

Nogueira et al. [2] proposed that natural behaviors of virtual characters can emerge from the coupling of the dynamics of the controller with the body and the environment in which the agent is embedded. For this purpose, the controller must be able to modify and adapt itself to the environment when placed in successive contacts with the world. Aiming at this goal, the authors developed a genetic encoding to evolve a plastic neuromodulated neural network to generate the signals that control the motors of a virtual robot based on its sensory information. The genetic algorithm (GA) is then responsible for shaping the network structure and, consequently, its dynamics.

1) The Neural Network: The neural network used in [2] is essentially a Continuous Time Recurrent Neural Network (CTRNN) [24] with two types of neurons: standard neurons, which are the processing units; and modulatory neurons, which are responsible for modulating the changes of weights in the synapses following the Hebbian learning rule [25]. The neurons are modeled according to the following differential equation:

$$\frac{\Delta y_i}{\Delta t} = \frac{1}{\tau_i}(-y_i + \sum_{j=1}^{n} \omega_{ji} f(s_j) + I_i), \qquad (1)$$

where t is time, y_i and τ_i are, respectively, the internal state and time constant for each neuron i, ω_{ji} is the weight of the jth input synapse of neuron i, s_j is the state of the neuron linked to the jth input synapse, $f()$ is the activation function of a neuron and I_i is a constant external input to neuron i (the same I is used for all neurons). The used activation function is as follows: $f(x) = tanh(\frac{x}{2})$. This function has a continuous behavior with upper and lower limits, approaching the potential accumulation and shooting idea observed in biological neurons [26].

In addition to this standard dynamics, the autors included a neuromodulatory feature in the neural network. The neuromodulation plays an important role in neural substrates from invertebrates to the human brain, and is related to the induction of Late phase - Long Term Potentiation (L-LTP), a phenomenon of permanent growth of the synaptic contact in the brain, causing synaptic stability, being a potential candidate for explaining memory functions involving neural wiring and, consequently, learning. Another mechanism of neural wiring related to neuromodulation is the Long Term Depression (LTD), the permanent decrease of the synaptic contact ([27]).

The signals of the standard and modulatory neurons are computed according to Equation 1 exactly the same way. However, only the input signals that come from standard neurons and the weights of the respective synapses are considered for the summation in that equation.

The value of modulatory activation m acting on a neuron i is computed as follows:

$$m_i = \sum_{j \in Mod} \omega_{ji} \cdot tanh(\frac{s_j}{2}), \qquad (2)$$

where j represents all modulatory neurons connected to neuron i, ω_{ji} is the weight associated with the synapse that links modulatory neuron j to neuron i and s_j is the internal state of modulatory neuron j.

The change of synapses weight is defined by the following equation:

$$\Delta\omega_{ji} = tanh(\frac{m_i}{2}) \cdot \eta \cdot [A_{o_j o_i} + B_{o_j} + C_{o_i} + D] \quad (3)$$

where $\Delta\omega_{ji}$ is the amount of change in the synapse that links the standard neuron j to a neuron i, η is the learning rate, A, B, C and D are tunable parameters, and o_j and o_i are the activation function $tanh(\frac{x}{2})$ applied to internal states of neurons j and i respectively. Note that this rule is unstable if the parameters are such that $\Delta\omega_{ji}$ is always positive or always negative. To avoid that, the weights of all input synapses of each neuron are limited. Equation 3 differs from the Hebbian rule only on the modulatory term, i.e., $tanh(\frac{m_i}{2})$. That is, the modulatory effect changes the learning rate of the Hebbian rule.

The model presented by the autors defines some standard neurons as either afferent neurons or efferent neurons, which have no internal dynamics. The internal state of an afferent neuron is a sensor value and cannot receive input from other neurons, while an efferent neuron stores the average of the internal states of each neuron connected to it, defining the outputs of the network [26].

2) The Genetic Encoding: The evolution of Artificial Neural Networks (ANNs) enables an additional level of adaptability on the plasticity. The use of genetic algorithms allows the choice of optimal parameters in the dynamic equation, which may lead to the organization of plasticity, making it more effective. The main problem in the evolution of ANNs is the definition of a consistent encoding that allows the application of genetic operators by Genetic Algorithm. The most common mode of representation is an array of synaptic weights. However, this encoding fails to maintain the structure and consistency of network when you need to increase or reduce its size, or making topology search. Since, it is represented by a matrix, the network is always fully connected [26].

I	η	A	B	C	D

Fig. 2. Encoding of the global parameters of the network([26]).

The genetic encoding used by [2] is a simplified version of the Analog Genetic Encoding (AGE) [28] aiming to evolve a neuromodulated CTRNN. The main advantage of AGE is the fact that the genome is represented as a simple string of characters, allowing the use of traditional genetic operators such as crossover and mutation, while keeping the structure and consistency of the network. So, the chromosomes which encode the neural network are arrays of bits, with each group of 32 bits defining a gene. The first 8 bits (1 byte) of the gene are used to encode an identifier of the element that the gene represents in the network. The last 24 bits (3 bytes) specify a value that indicates a property of the decoded element.

Each individual in the population has two chromosomes: one chromosome stores the global parameters of the neural network, while the other keeps the network itself. The global parameters are the values of the variables in Equation 3 and the external stimulus of the CTRNN (Equation 1). Fig.2 shows the distribution of values in the chromosome.

The decoding process of the network's chromosome generates a list of neurons with their respective parameters and terminals that will compose the network. Then, neurons are paired by their terminals to create the synapses arriving at or leaving the neurons. This process allows evolving the neural networks using a simple canonical genetic algorithm [2] or even simulated sexual reproduction [1].

3) The Genetic Algorithm: The genetic algorithm proposed by Nogueira [26] can be divided in: into main step (selection operator), and two probabilistic steps (crossover and mutation operators).

The selection operator is responsible for choosing the individuals from the last generation that will create the new generation. Nogueira [26] uses essentially a *fitness-proportionate selection* (or roulette wheel selection), where the individuals with better evaluation have higher probability of being selected.

The crossover operator is responsible for exchange genetic data between the selected individuals, generating new individuals different from those previously tested. The method used by the authors performs the breakings of chromossomes from two individuals in the same point, permuting the correspondent pairs and creating two individuals, each one with one part of the chromossomes of the ”parents”. For each pair of selected individuals, Nogueira [26] adopts a probability of 60% for the occurrence of crossover.

After the crossover operator, the mutation operator brings the possibility of ramdomly modifying parts of chromossomes. This operator selects ramdom genes in a chromossome and changes their values, creating individuals that probably would not have been created if only crossover had been applied. Nogueira [26] applys a mutation probability of 0.1% for each gene of each individual.

4) The Simulation of Evolution: In [1], the authors demonstrated the capacity of the reported genetic encoding in the evolution of various behaviors in a population of robots by simulated sexual reproduction. The proposed controller’s model and its encoding were efficient in generating interesting emergent behaviors such as foraging, crossing and obstacle avoidance, without any description of objectives for the characters.

As discussed in the cited work, the goals of the characters were actually implicit in the system. With fruit distributed in the environment, from which the robots could get energy to survive, the Darwinian dynamics could act ensuring that the best adapted individuals (individuals able to stay more time alive, eating more fruits) would have more possibilities to reproduce and consequently spread their genes. Given this fact, it is possible to interpret the objective of the agent as a restriction imposed by the environment, which imposes (or guide) its adaptation to the environment. This restriction is a result of the environment itself and of the robot’s body conditions, as well as of the new situations created by competition with the evolutionary advancement of the characters.

The GA works similarly, with only one of the constraint variables defined by the programmer: the evaluation or fitness function. This function guides the agent to perform the desired behavior, but not specifically define its format, i.e., how exactly the movements occur. Since each different task to be learned by the GA uses different factors, the fitness function varies with the tasks. For the proposed navigation task, to consider the distance from the agent to the exit and the time that the agent takes to reach the exit could be enough, but it can change according to each task and scenario. For instance, if there are objects harmful to the agent in the scenario, those objects can be considered in the fitness function. The movements are achieved by the character living in the world in which it is inserted, and not predetermined externally by the programmer/animator. So, the coherence between behavior and environmental situations is sustained, the main quality of emergent behaviors. It is also possible to perceive the emergence of other goals besides the one explicitly described in the fitness function [2].

B. The Artificial Visual Cortex

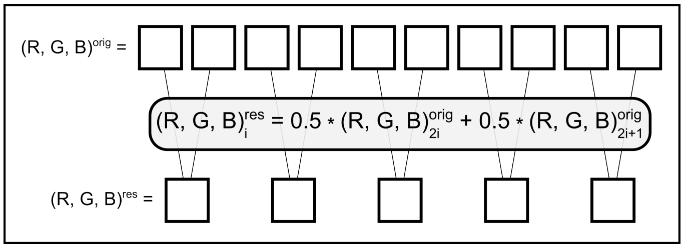
$(R, G, B)^{orig}$ =
$(R, G, B)_i^{res} = 0.5 * (R, G, B)_{2i}^{orig} + 0.5 * (R, G, B)_{2i+1}^{orig}$
$(R, G, B)^{res}$ =

Fig. 3. Pixels selection of the artificial visual cortex. The central portion of the image captured by the virtual camera is reduced to five pixels by the arithmetic mean of RGB components from each two neighbor pixels. In the given formula, $(R,G,B)^{orig}$ is the array of the ten central pixels of the captured image and $(R,G,B)^{res}$ is the array of the five pixels resulting from the reduction, with i being the index of each pixel.

Based on the controller presented in Section A, we analyze the generation of characters that guide themselves using their own vision to find the exit from a room. For this purpose, we used the same neural network and the same genetic encoding applied to a canonical AG, but with a new sensorial description. To achieve the specified objective, this paper proposes the evolution of an artificial visual cortex based on images captured by a virtual camera (Fig. 1). Images with 10×10 pixels resolution in the RGB color system are captured and given to the controller. Since the displacement of the agent occurs only in two dimensions, it is sufficient for the agent to see the central portion of the image and, therefore, we use only the ten pixels from the fifth row as vision. These pixels, on the other hand, are reduced to five using the arithmetic mean of RGB components of each two neighbor pixels (Fig.3) in order to reduce the amount of information to be processed. Since the number of afferent neurons depends on the number of pixels passed to the agent, we reduce the number of pixels to simplify the Neural Network, trying to accelerate the learning of the navigation task without great loss of information. The number of considered pixels from the captured image must be coherent with the scenario and the amount of perceptible elements. For instance, if the height of the agent is significant, more lines of pixels should be considered, since the door could be in the lower lines of the image. The resulting colors are then given as input to the neural network, which has, therefore, sixteen afferent neurons, with fifteen neurons of vision (five pixels with three colors each) and one neuron of energy sense (in the same way specified in [2]).

IV. Tests and Results

The tests were performed with a computer equipped with a 2.20 GHz Intel Core i7 processor, 8.0 GB of RAM and a NVidia GeForce GT 540M graphics card with 2 GB of RAM. The program was written in C/C++ . Our simulation was developed with the Irrlicht 3D Engine[1], with physics provided by the Bullet Physics Engine[2]. The tests were performed to analyze the applicability of the technique in the study of navigation behavior of one agent in a variety of simple simulation environments with the presence of obstacles.

To our knowledge, there is no work in the literature that deals with emergent navigation behavior aided by an evolved visual cortex. Therefore, at this point, we feel that it is pointless to try to compare our results with the preprogramed navigation behaviors found in the literature.

Using the proposed technique, the configuration of an evolution simulation should follow three steps: **the scenario description**, which defines all the elements needed by the task; **the agent description**, which defines the agent's body, its motors, its perception system and its chromosomal rules; and **the fitness function** definition, considering all the relevant variables for the simulation in question. From the informed chromosomal setting, the GA generates the first population (Generation 0) randomly. Each individual is then sequentially placed in the simulation scenario and its performance is evaluated according to the fitness function. After all the individuals of Generation 0 are evaluated, the individuals with better evaluations are selected by the GA to create the new population (Generation 1). Likewise, the individuals of Generation 1 are evaluated and the loop can continue until it reaches a convergence of evaluation, without significant improvement among the Generations.

Each one of those steps are described for the proposed test and the results are presented as follows.

A. Description of Scenario

The simulations scenarios consist of a square area bounded by walls and a single door. The door position is randomly defined among eight possible positions and its position changes for each evaluated individual in the population. The random positioning of the door whenever a new individual is evaluated makes the genetic algorithm to adapt the neural network to act completely independent of where the exit is, avoiding a "memorization" of the door's position, which would generate individuals that would always move in just one direction. Because an agent uses its own vision to navigate through the environment, the colors of objects are chosen to provide visual contrast and to provide visual differentiation to the agent. Thus, the walls have a gray texture and the door is yellow.

B. Description of the Agent

The agent is represented by a Khepera-like virtual robot [29]. It has a cylindrical body with a black box that plays the role of an eye. In addition to the agent under brain evolution process, the environment is populated with other robots that do not move, working as obstacles that can be moved by the agent) at the environment. All robots are randomly positioned. The agent who is evolving has an amount of energy that is reduced proportionally to the strength of the signals generated by the network. If the energy is exhausted, the agent dies, ending the evaluation of the its brain and restarting the scenario

[1] http://irrlicht.sourceforge.net/

[2] http://bulletphysics.org/

with the next virtual brain to be analyzed. In each generation, 100 brains are analyzed.

The robot has two motors: one to move forward or backward and another to make left or right turns. Each motor is controlled by one efferent neuron (for more details see [1]). When the first motor receives a positive value, the robot moves forward and when it receives a negative value, the robot moves backward. With a positive value, the second motor makes the robot turn right, otherwise, it causes the robot to turn left.

The actions of the motors are not physically accurate, since the amplitude of the signals is directly applied to the robot's speed. However, the interactions among the bodies of the robots, i.e., the reaction forces, are properly calculated, providing the possibility of a robot to move others.

C. Fitness Function

The fitness function, f_f, used for adaptation and generation of the desired behavior is given by:

$$f_f = \frac{1}{t} + \frac{1}{d}, \tag{4}$$

where t is the time the agent takes to reach the door and d is the final distance between the agent and the door. Thus, the agent with best evaluation will be the one that came closest to the door in the shortest possible time. The choice of this function was taken in order to allow the evaluation of individuals that could not be able to reach the door, providing a certain degree of value to the simple approach to evolution. Some individuals may have a satisfactory behavior, but cannot reach the door for many reasons, such as be blocked by other robots or by insufficient energy. So, considering individuals that approach the door without reaching it, we have a gradual evolution of the population towards a behavior of seeking and getting near and near to the exit door.

Because a fitness function is defined depending on the task to be learned and on the relevant elements in the environment, even the same task could be done with distinct fitness functions. For example, for the simulation presented here, the fitness function could consider only the agents that reached the door, ignoring the agents that identify the door and walk toward it but, for some reason, are unable to reach it. We decided to consider the agents that walk toward the door even if their do not reach it, because the evolution seems to be better this way.

D. Generation of Emergent Navigation Behavior

The neural network was put to evolve in order to observe the emergence of the navigation behavior of an agent in different scenarios. The fitness function gives each agent a value that defines how much it was able to satisfy the objective of reaching the room's exit door. Thus, in each generation, the agent that came closest to the door in the shortest possible time, receives a better evaluation than the other agents, and their chromosomes have a higher possibility of being selected for inclusion in the next generation. In order to show the results presented in this paper, we selected the individual with best evaluation after 65 generations, when it is possible to observe satisfactory navigation behavior.

Fig. 4 shows three sequences of images displaying a particular agent walking through the environments. In each sequence, the first image shows the initial position of the agent (highlighted in red). The other images show how the agent moves through the environment. Analyzing the images, we can observe how the agent uses its own vision to orient itself (it is always seeing something that can be used to guide it through the environment). When the door is not in its visual field, or cannot be seen clearly, the agent moves around the scene, turning in different directions, in a recognition task of the environment that surrounds it. This can be seen in pictures 1 and 2 from areas A, B and C of Fig.4. That kind of behavior can make the agent move greater distances than the ideal. It is possible that the agent behave differently, if the total displacement were considered in the fitness function. When the door appears in the visual field, the agent tends to move toward it as shown in frames 3 and 4 of areas A, B and C in Fig.4.

In order to show that the agent is able to guide itself through the environment using its own vision, in Fig.5, we show the visual field of the agent. Fig.5 is composed of 10 frames showing from the initial position of the agent until the moment it approaches the door. At the top left of each picture, one can see what the robot is seeing during its movement into the environment.

We observe that, when the door can be clearly seen, the agent identifies it and moves toward the

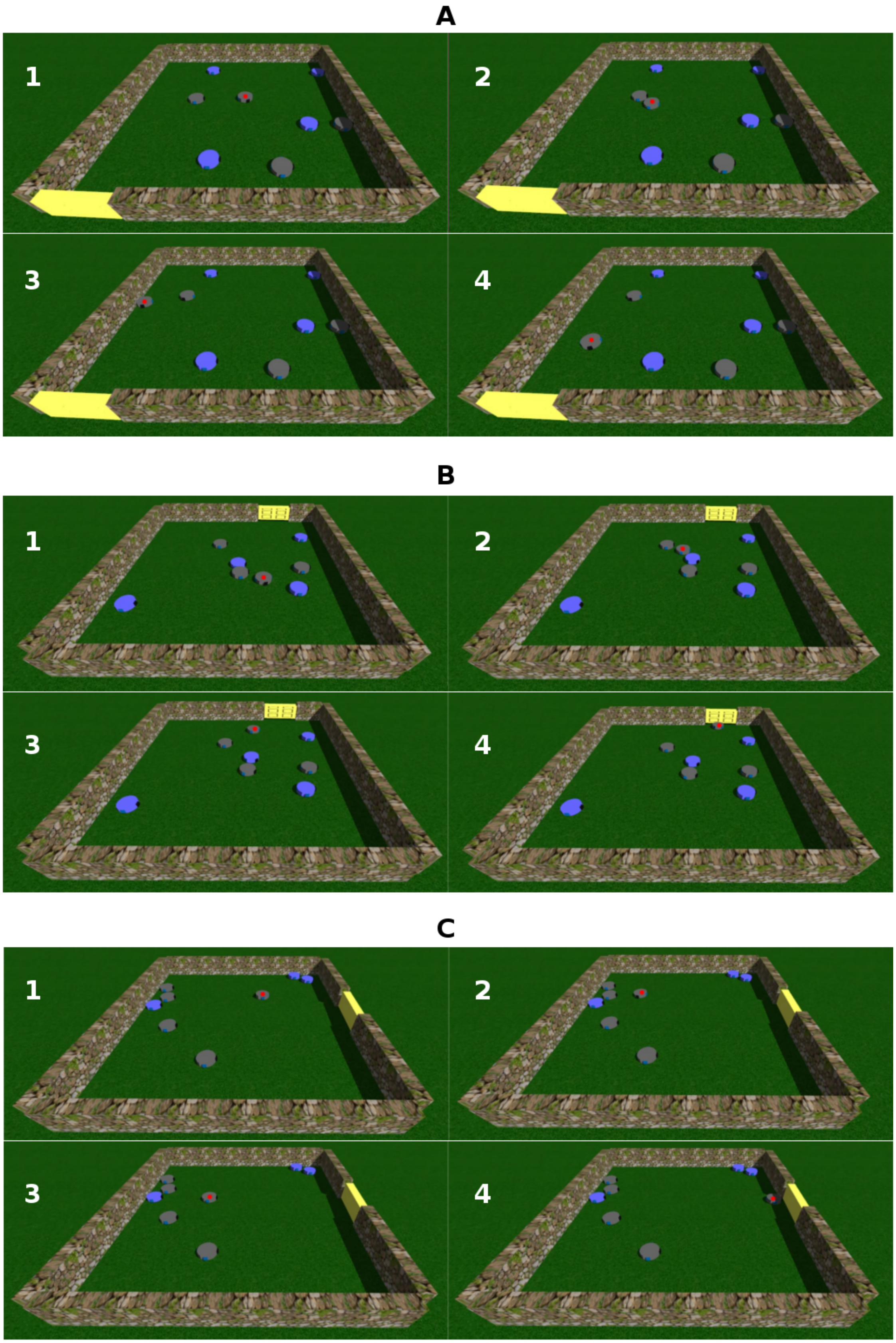

Fig. 4. Navigation Behavior. The cases A, B and C consist of four pictures showing the trajectory of the agent endowed with the evolving brain (highlighted with red dot). In Case C, we note that the agent moves away from the door since it is looking in the opposite direction to it. As the agent turned, it sees the door and moves toward it.

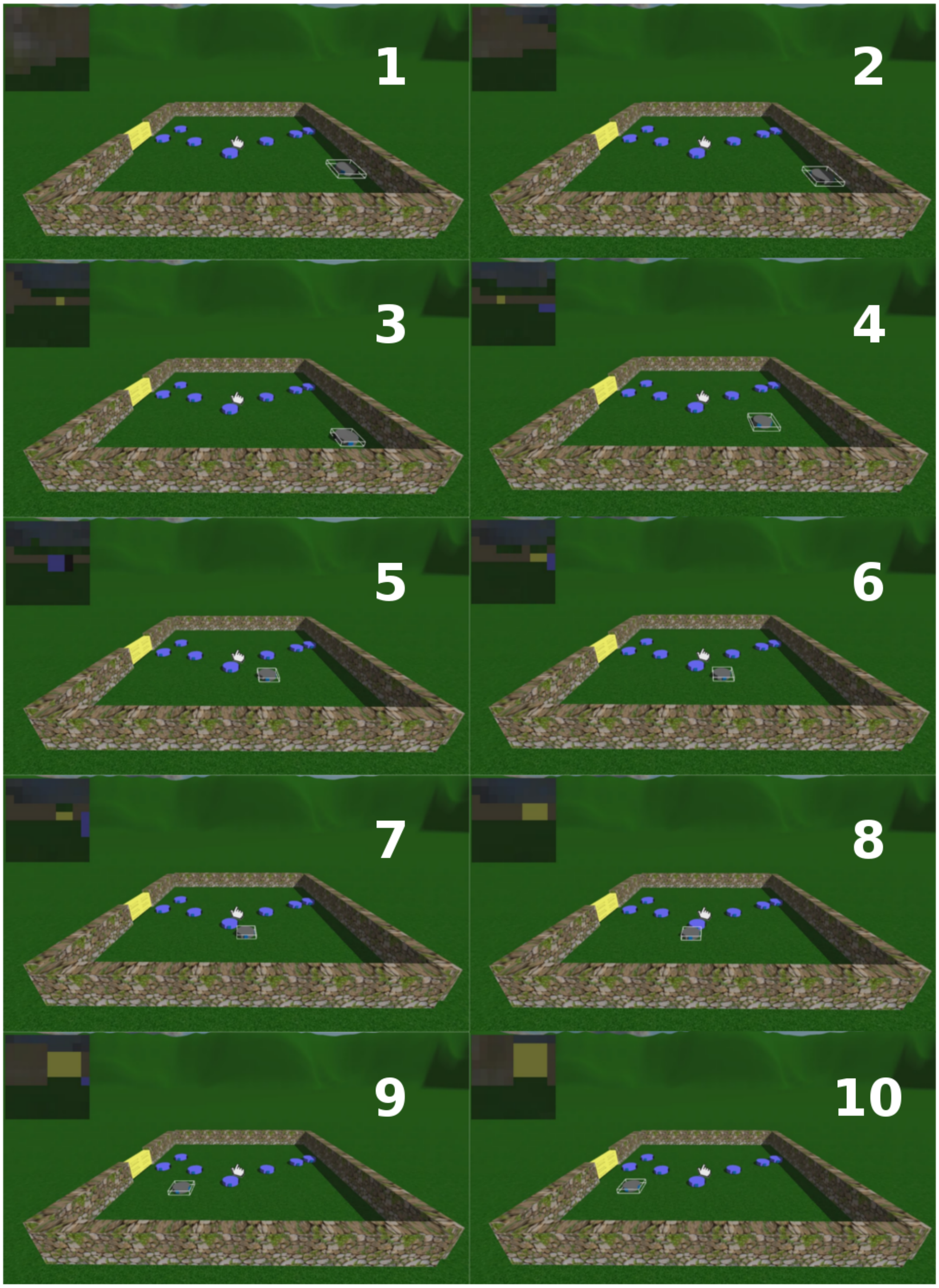

Fig. 5. Locomotion of one agent guided by its own vision. In the top left part of each picture we can see the visual field of the selected agent.

door guided only by its vision. It is noteworthy that the agent is not endowed with memory and, therefore, if it sees the door and something blocks its vision during the movement, the agent starts to seek the exit again. However, we can note that, when the view of the door is blocked (Frame 5), the agent vary its motors to keep a slightly similar direction to the one it was following until the door was visible again (Frame 6).

From those results, we can say that the genetic algorithm was able to evolve the neural network to control the robot in the navigation task, as can be seen in Fig.6. The graph in Fig.6 was obtained from the arithmetic mean of the average evaluation of one hundred individuals in each generation, considering five runs. Moreover, in each of those runs, the number of individuals that reached the door in each generation were counted. Fig.7 depicts a plot indicating the average number of individuals that reach the door in each generation.

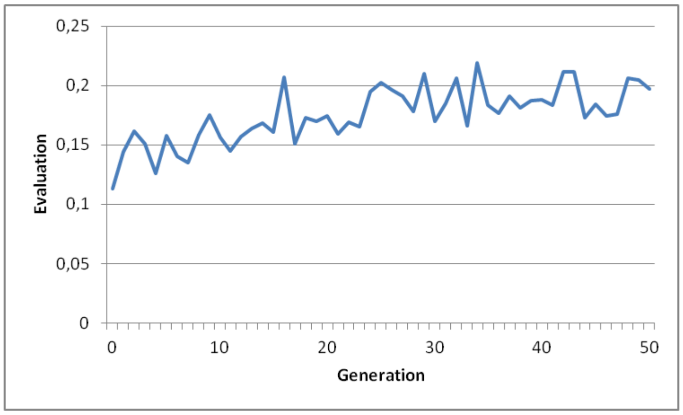

Fig. 6. Evolution of the average evaluation in each generation considering five runs.

Furthermore, we consider the best and the worst runs of five runs performed, resulting in the graph of Fig. 8. This graph takes into account the evolution of the average evaluation of one hundred individuals in each generation, taking the first 50 generations. The blue curve represents the result related to the best run, and the red curve is related to the worst run. Analyzing the graph, we can say that even in the worst run, the genetic algorithm was able to evolve the neural network to control the robot in the navigation task. Thus, the technique proved to be robust in the study of emergent navigation behavior of an agent in a variety of simple simulation scenarios.

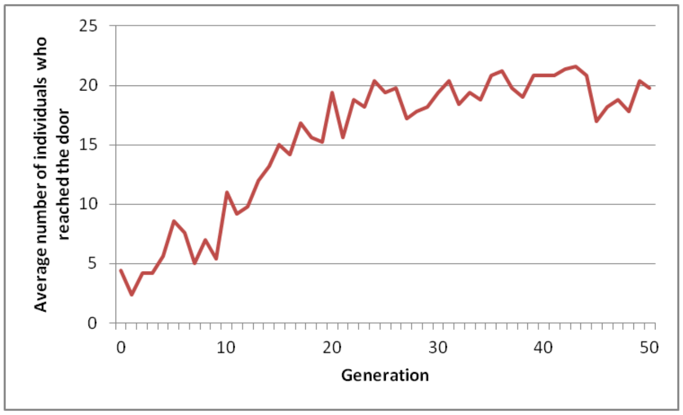

Fig. 7. Average number of individuals who reached the door in each generation considering five runs.

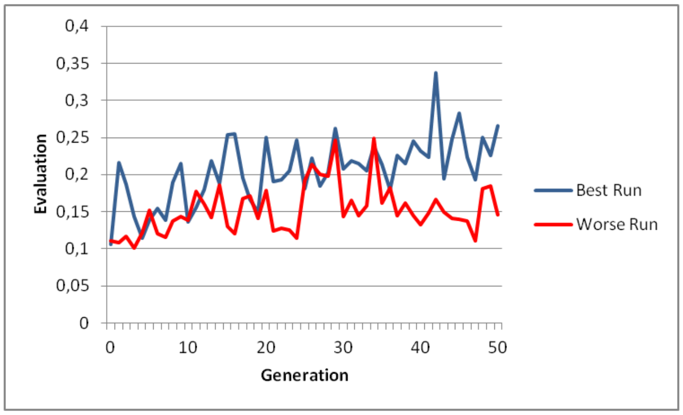

Fig. 8. Evolution of the average evaluation in each generation, taking the best and the worst of five runs.

In order to validate our model, we analyzed the behavior of the agent when a new situation is presented to it. For this, we have changed the environment to the one the agent was already adapted. We selected one of the simulation scenarios used in the previous test and added some fixed obstacles. These obstacles consist of five pillars placed in the environment, including one in front of the door.

For running this test, we selected the individual with best evaluation after 65 generations of the previous example, i.e, the same individual of the previous case study was used in this new test, without the need to evolve new brains to the task of identifying and reaching the door in a scenario with new elements (the pillars). As the pillars have the same texture of walls, the agent is able to identify them as obstacles. If we had inserted new objects with textures different from those the agent is adapted, the agent would need to adapt to these new objects, i.e, to learn how distinguish them from others.

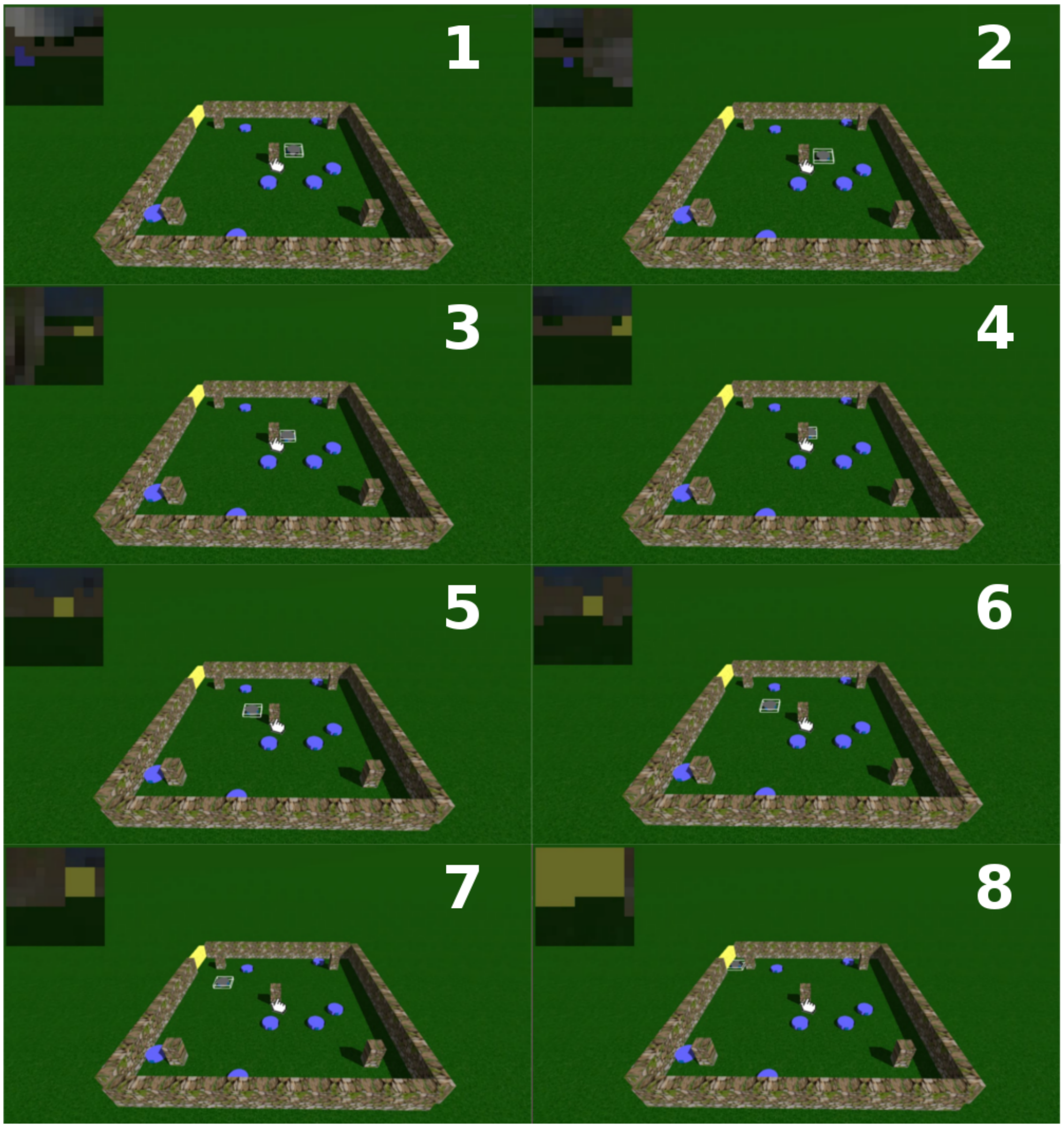

Fig. 9. Agent's behavior when a new situation is presented to it. The agent reaches the door successfully avoiding the columns.

In the Fig. 9 we show the behavior of the agent in this new test. Fig. 9 consists of 8 frames that show the movement from the initial position of the agent until the moment it reaches the door. At the top left of each picture one can see what the robot is seeing during its movement into the environment.

Analyzing the image sequence, we can observe that the agent, even in a new situation, is able to guide itself toward the door. Note that at the beginning of the movement, a column is in the field of view of the agent. When the agent sees the door, it avoids the column and moves toward the exit. We can also note in Frame 5 that the column located in front of the door is already in the field of view of the agent, but, because of the simplified vision the column and the wall seems to be the same object.

Even when the position and the number of obstacles change, the agent is able to identify the door, as shown in Fig. 10. Moreover, when the number of obstacles is sufficiently large that occupies a significant portion of the avaliable area in which the agent can move, it is forced to push the other robots that are blocking its way to the door, as shown in frames 5, 6, 7 and 8 of Fig. 10.

The presented results in this work show that a plastic controller is really able to produce an intimate coupling between an agent and its environment.

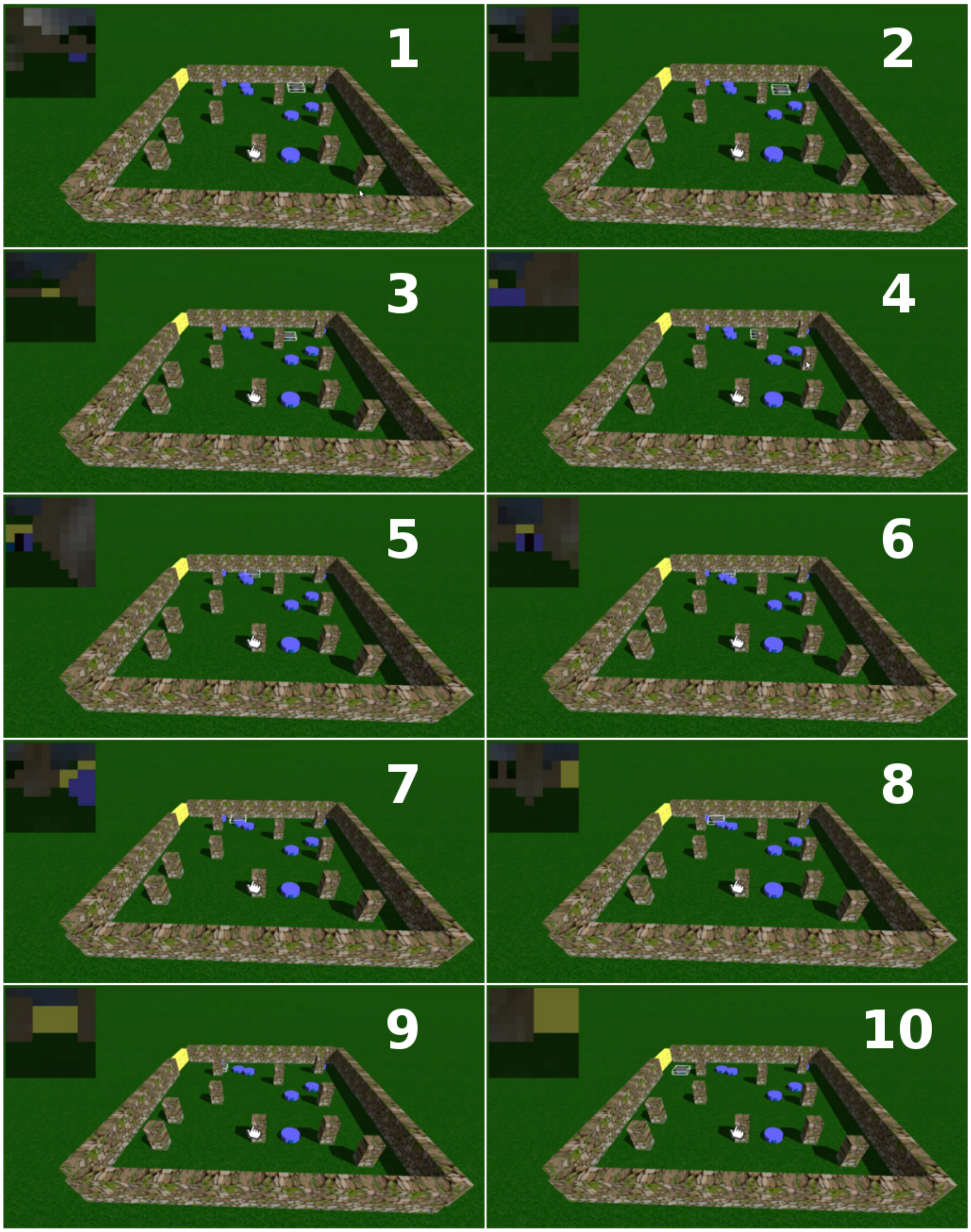

Fig. 10. Agent's behavior in a situation with more pillars. As the number of obstacles reduces the avaliable area for locomotion, the agent pushes the robots on the way to the door, as can be noted in frames 5, 6, 7 and 8.

V. Conclusion

In this work, we study the dynamics of motion of one sigle agent in simple environments. We generate individuals that are able to guide themselves using their own vision in the task of finding the exit from a room. For this purpose, we applied the same neural network and the same genetic encoding applied to a canonical GA used in [1] and [2], but with a new sensorial description, promoting the evolution of an artificial visual cortex based on images captured by a virtual camera.

Tests showed that the GA was able to successfully evolve the neural network for controlling the robot in navigation tasks. The agents were able to navigate through the environment and find the exit, in a non-programmed way, using only visual information passed to the neural network, even when a new situation is presented, as in the insertion of pillars in the environment. This can facilitate the application to various environments, without displaying possible flaws of a behavioral modeling as in other techniques.

The possibility of using agents that adapt to different environments and that can interact with objects initially unknown to them, reveals new ways for simulation of more natural behaviors. Although the results have been generated from simulation of relatively simple environments and agents represented by robots, we believe that the proposed approach has the potential of simulating environments with greater complexity and with more complex virtual characters, as well as of exploring other behaviors observed in real crowds without the need for modeling and deterministic programming. Future works are heading in that direction. Moreover, we desire to provide the agent with a memory architecture, so it can perform better in situations where the vision of the door is blocked by obstacles during the simulation. Also, the visual cortex can be upgraded to satisfy situations in which robots have different heights or situations in which the robots have to identify different objects with the same color.

Acknowledgment

This work was supported by Brazilian funding agencies Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

[1] Y. Nogueira, C. Brito, C. Vidal, and J. B. Cavalcante-Neto, "Emergence of autonomous behaviors of virtual characters through simulated reproductions," in *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems*, vol. 12. Cambridge: MIT Press: Advances in Artificial Life - ECAL 2013, 2013, pp. 750–757. [Online]. Available: http://dx.doi.org/10.7551/978-0-262-31709-2-ch107

[2] Y. Nogueira, C. Brito, C. Vidal, and J. B. Cavalcante-Neto, "Evolving plastic neuromodulated networks for behavior emergence of autonomous virtual characters," in *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems*, vol. 12. Cambridge: MIT Press: Advances in Artificial Life - ECAL 2013, 2013, pp. 577–584. [Online]. Available: http://dx.doi.org/10.7551/978-0-262-31709-2-ch083

[3] L. Z. Yang, J. L. Zhao, and T. Y. Fang, "Simulation of the kin behavior in building occupant evacuation based on cellular automaton," *Building and Environment*, vol. 40, no. 3, pp. 411–415, March 2005. [Online]. Available: http://dx.doi.org/10.1016/j.buildenv.2004.08.005

[4] J. Shi, A. Renb, and C. Chenb, "Agent-based evacuation model of large public buildings under fire condition," *Automation in Construction*, vol. 18, no. 3, pp. 338–347, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.autcon.2008.09.009

[5] S. Musse and D. Thalmann, "Hierarchical model for real time simulation of virtual human crowds," *Visualization and Computer Graphics*, vol. 7, no. 2, pp. 152–164, Apr 2001.

[6] N. Pelechano, J. M. Allbeck, and N. I. Badler, "Controlling individual agents in high-density crowd simulation," in *SCA '07 Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 99–108.

[7] W. Shao and D. Terzopoulos, "Autonomous pedestrians," *Graphical Models*, vol. 69, no. 5-6, pp. 246–274, 2007.

[8] D. Terzopoulos, "Artificial life and biomechanical simulation of humans," *Digital Human Symposium 2009*, pp. 8–13, 2009.

[9] D. Thalmann and S. Musse, *Crowd Simulation*. Springer London, 2008.

[10] M. Kapadia and N. I. Badler, "Navigation and steering for autonomous virtual humans," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 4, no. 3, pp. 263–272, 2013. [Online]. Available: http://dx.doi.org/10.1002/wcs.1223

[11] N. Pelechano, A. Jan, and N. I. Badler, *Virtual Crowds: Methods, Simulation, and Control*. Morgan & Claypool Publishers, 2008.

[12] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, pp. 601–644, 1983. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.55.601

[13] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensionalcellular automaton," *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3-4, pp. 507–525, June 2001. [Online]. Available: http://dx.doi.org/10.1016/S0378-4371(01)00141-8

[14] A. Kirchner, A. Namazi, K. Nishinari, and A. Schadschneider, "Role of conflicts in the floor field cellular automaton model for pedestrian dynamics," in *Proc. 2nd International Conference on Pedestrians and Evacuation Dynamics*, 2003, pp. 51–62.

[15] H. Klupfel, "A cellular automaton model for crowd movement and egress simulation," PhD thesis, University of Duisburg-Essen, 2003.

[16] O. B. Bayazit, J. M. Lien, and N. M. Amato, “Roadmap-based flocking for complex environments,” in *Proc. Pacific Conference on Computer Graphics and Applications*, 2002, pp. 104–113.

[17] J. M. Lien, S. Rodriguez, J. P. Malric, and N. M. Amato, “Shepherding behaviors with multiple shepherds,” in *Proc. IEEE Robotics and Automatation (ICRA)*, 2005, pp. 3402–3407.

[18] S. Teller, “Visibility computations in densely occluded polyhedral environments,” PhD thesis, University of California, 1992.

[19] N. Pelechano and N. I. Badler, “Modeling crowd and trained leader behavior during building evacuation,” *IEEE Computer Graphics and Applications*, pp. 80–86, 2006.

[20] A. Lerner, Y. Chrysanthou, and D. Cohen-Or, “Efficient cells-and-portals partitioning,” *Computer Animation & Virtual Worlds*, pp. 21–40, 2006.

[21] A. Treuille, S. Cooper, and C. Popovic, “Continuum crowds,” in *Proc. ACM Transactions on Graphics (SIGGRAPH 2006)*, 2006, pp. 1160–1168.

[22] C. W. Warren, “Global path planning using artificial potential fields,” in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, vol. 1, 1989, pp. 316–321.

[23] C. W. Warren, “Multiple robot path coordination using artificial potential fields,” in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, vol. 1, 1990, pp. 500–505.

[24] R. D. Beer, “On the dynamics of small continuous-time recurrent neural networks,” *Adaptive Behavior, SAGE Publications*, vol. 3, no. 4, pp. 469–509, 1995.

[25] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. Wiley, 1949.

[26] Y. Nogueira, “Integração mente e ambiente para a geração de comportamentos emergentes em personagens virtuais autônomos através da evolução de redes neurais artificiais,” *PhD Thesis*, Federal University of Ceará, Fortaleza, CE, 2014.

[27] A., Soltoggio, J. Bullinaria, C. Mattiussi, P. Drr and D. Floreano, “Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward- based Scenarios,”*Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, 2008, pp. 569–576. MIT Press.

[28] C. Mattiussi and D. Floreano, “Analog genetic encoding for the evolution of circuits and networks,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 5, pp. 596–607, Oct 2007.

[29] F. Mondada, E. Franzi, and A. Guignard, “The development of khepera,” in *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshopn*, 1999, pp. 7–14.