Autonomous spatial organization of behavior through categorization of objects based on affordances

Paper 227

ABSTRACT

We present an architecture for self-motivated agents to organize their behaviors in space according to possibilities of interactions afforded by initially unknown objects. The long-term goal is to design agents that construct their own knowledge of objects through experience, rather than exploiting pre-coded knowledge. Self-motivation is defined here as a tendency to experiment and to respond to behavioral opportunities afforded by the environment. Some interactions have predefined valences that specify additional inborn behavioral preferences. Over time, the agent learns the relation between its perception of objects and the interactions that they afford, in the form of data structures, called signatures of interaction, which encode the minimal spatial configurations that afford an interaction. The agent keeps track of enacted interactions in a topological spatial memory, to recognize and localize subsequent possibilities of interaction (through their signatures) afforded by surrounding objects. The distance that defines the topology is the length of the sequence of interactions to move from a point to an other. Experiments with a simulated agent and a robot show that they learn to take into account the position of multiple objects and to reach or avoid objects according to the valence of the interactions that they afford.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning,; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*.

General Terms

Algorithms, Experimentation

Keywords

Autonomous learning, Constructivism learning, Spatial Awareness, Body schema, Affordances, Self-motivation.

1. INTRODUCTION

In this paper, we address the problem of the construction, interpretation and exploitation of a short term representation of the surrounding environment by an artificial agent that initially ignores the elements that compose its environment. Such an agent can be defined as an *environment-agnostic agent* [7]. In order to respect this constraint, the learning mechanism of the agent must be independent from environment states. We base our work on a design principle in which perception and action are kept embedded within data structures called *interaction*, rather than being separated, as it is the case in traditional modeling approaches. This design principle is intended to account for cognitive theories that suggest that perception and action are inseparable (i.e. O'Regan [13], Piaget [16]). Specifically, interactions are used to model Piaget's notion of *sensorimotor scheme*. This design principle is formalized as a special case of Partially Observable Decision Process (POMDP) [1] called Enactive Markov Decision Process (EMDP) [6]. In an EMDP, the agent is given a predefined set of interactions associated with *satisfaction values*, and seeks to enact interactions with predefined positive values and to avoid interactions with predefined negative values. The learning mechanism is not based on rewards that depend on environment states, and thus respects the principle of environment-agnosticism. This motivation principle is called *interaction motivation* and is related to the problem of intrinsic motivation [14]. Because an EMDP agent can only obtain information about its environment through the active enaction of interactions, it implements a form of active perception.

Our implementations[1] have shown that an agent equipped with a sequential EMDP algorithm was able to autonomously capture and exploit hierarchical sequential regularities afforded by the environment. But such agents are unable to organize their behaviors in space and did not recognize the object permanence[5]. These implementations have shown the importance for an agent to maintain a representation of its surrounding environment to overcome these problems. A model of short range spatial representation based on sequences of interaction was proposed by Gay and Georgeon [4]. This model, called Spatio-sequential System, allows an agent to define elements of its environment based on sequences of interaction, and complete its perception by recognizing and updating these sequences. However, this model was limited to the close surrounding environment and was not sufficient to navigate in an open environment, which suggests that an additional mechanism must be implemented to take far objects into consideration.

We propose a structure that constructs, maintains and ex-

Appears in: *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), Lomuscio, Scerri, Bazzan, Huhns (eds.), May, 5–9, 2014, Paris, France.*

[1]http://e-ernest.blogspot.fr/2012/03/small-loop-challenge.html

ploits a representation of the environment using interactions enacted by the agent, with a long-range scale. This structure is inspired from biology. Indeed, most natural organisms have brain structures that maintain some geometrical correspondences with the animal's local surrounding environment [2]. We do not intend to develop a path planing mechanism (such as mechanisms inspired by hippocampus structures [11,12]), nor a mapping mechanism, but a mechanism inspired by simple structures such as the tectum of vertebrates, that allows an agent to recognize and localize elements of its surrounding environment. The structure we propose generates an egocentric representation of the environment based on recognized possibilities of interactions which are localized by the sequence of interaction needed to reach them. More broadly, these studies investigate the hypothesis that the geometric structure of space does not need to be presupposed by a cognitive system. This hypothesis was formulated by theoreticians such as Poincaré [17], for who objects are localized in space according to the movement needed to reach them.

Section 2 summarizes the EMDP formalism. Section 3 presents the spatial representation mechanism. The implementation on artificial agents is described in section 4, and the experiments we provided to test our mechanism are described in section 5. Finally, the paper discusses our results and draws up recommendations for future works.

2. FORMALIZATION OF THE EMDP

Formally, the EMDP described in [6] is defined as a tuple (Σ, I, q, ν, s) where Σ is the finite set of states σ of the environment, I the initial set of primitive interactions ι offered by the coupling between the agent and the environment (a primitive interaction represents an atomic sensorimotor scheme), q the distribution of probability of transition from a state σ_t to a state σ_{t+1} after attempting to enact an interaction ι at step t, ν the distribution of probability to result in an enacted interaction e_t after an attempt to enact an intended interaction i_t , and s the satisfaction value function. This function differs from the reward function of standard reinforcement learning problems in that it is not a function of an environmental state σ, but of the enacted interaction e_t. The principle of an EMDP is that the agent tries to enact, at the beginning of a step t, an intended interaction i_t, and is informed by the environment about the primitive interaction that was effectively enacted, e_t. If the intended interaction was effectively enacted, the enaction of this interaction is a success, and a failure otherwise.

We extend the EMDP mechanism by adding an additional set F of primitive observations. A primitive observation is an additional perception generated by the enaction of an interaction, has no a priori signification for the agent, and must be associated with the enacted interaction that produce it. Indeed, the information given by a primitive observation can depend on the movement of the agent. We call Dynamic Feature (DF) ϕ a primitive observation associated with an interaction and $\Phi = \mathrm{I} \times \mathrm{F}$ the set of possible DF. An enacted interaction can return an arbitrary number of DF. We note E_t the enacted ensemble, containing the enacted interaction and the produced DFs:

$$E_t = (e_t, \phi_{1,t}, ..., \phi_{n,t}) \quad (1)$$

The satisfaction value of an enacted ensemble E_t is the sum of satisfaction values of interactions and DFs that compose it. We note $\Psi = \mathrm{I} \cup \Phi$ the set of interactions completed with the set of DFs. We refer an interaction ψ as an element of Ψ, a primitive interactionι as an element of I and a DF ϕ as an element of Φ. Figure 1 illustrates this formalism.

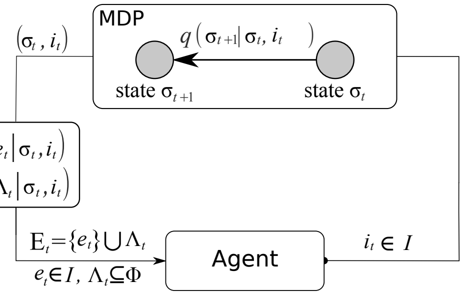

Figure 1: Diagram of the extended Enactive Markov Decision Process (adapted from EMDP [6]).

3. THE SPATIAL MEMORY SYSTEM

We propose an architecture that allows an agent to take into consideration elements of the environment that can afford interactions. We developed a mechanism, we called Spatial Memory System (SMS), that helps an agent to characterize its current situation by maintaining a representation of its surrounding space based on interactions. This section formalizes the concepts used to implement the SMS: signatures of interactions, extra-personal space memory and decision system. We provide details of our implementation

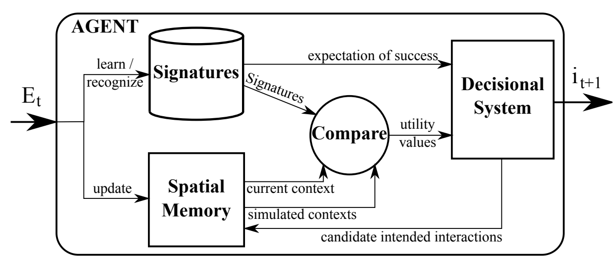

Figure 2: Diagram of the Spatial Memory System (SMS). At step t, the agent receives the set of previously enacted interactions E_t. The agent uses E_t to learn and recognize signatures of interaction and to update its spatial memory. The signatures then define the expectation of success of primitive interactions in the current context defined by E_t. The Decisional System uses these predictions to define a set of candidates intended interactions $i_{c,t+1}$. , and uses the spatial memory to simulate each candidate interactions. The signatures recognize and localize possibilities of interactions of each interaction in the current and predicted spatial contexts defined by the spatial memory. The positions of possibilities of interactions in the current context and the simulation of a candidate intended interactions are compared to define an utility value to this candidate. The decisional system then selects the intended interaction of the next step i_{t+1} according to its satisfaction value and utility value.

when it can help clarify the description of these mechanisms. Figure 2 gives the structure of the SMS.

3.1 Signatures of Interactions

Our mechanism is based on the assumption that the result of the enaction of an interaction depends on a limited set of elements of the surrounding environment. These elements, we call *objects*, can thus be characterized by the set of interactions that can be enacted in their presence. Overall, by representing objects by the possibilities of interaction that they afford, we investigate theories of cognition that suggest that knowledge of the world arises from interaction (e.g. Gibson [8], and Hume [9]).

We propose to organize the perception of the agent, defined by the enacted ensemble E_t, under the form of a vector that gives the result (i.e. success, failure or not used) of each interaction of Ψ. We define the Interactional Context C_t as a vector $[\epsilon_{1,t}, \ldots ,\epsilon_{m,t}]$, with m=Card($\Psi$). $\epsilon_{k,t} = 1$ if ψ_k is enacted (i.e. $\psi_k \in E_t$), $\epsilon_{k,t} = -1$ if ψ_k failed, and $\epsilon_{k,t} = 0$ if ψ_k was not enacted nor intended. Note that a DF cannot fail as it cannot be intended.

We define the certitude function c_ψ as a function, learned by the agent, that predicts the result of an interaction $\psi \in \Psi$ in an interactional context C, with an absolute certitude of success if $c_\psi(C) = 1$ and an absolute certitude of fail if $c_\psi(C) = -1$. In order to generalize the certitude function, we define the general certitude function c defined by $c(S_\psi, C) = c_\psi(C)$, with S_ψ a set of parameters that characterizes the interaction ψ, we call Signature of ψ. We expect the signature of an interaction to represent the "object" that allows this interaction to be successfully enacted.

We propose to implement the certitude function with an artificial neuron, using the interactional context C as input and the certitude values as output. The signature is defined by the set of weights w_k of the synapses, and the bias b of the neuron. Thus, the signature S_ψ of an interaction ψ is defined with a vector $[w_{\psi,1},\ldots, w_{\psi,m}, b_\psi]$, with m = Card($\Psi$). The certitude function can be defined with a weighted sum (2). We use a sigmoïd function as activation function to restrain the output value to the [-1;1] interval.

$$y = \sum_{k\in[1;m]} (\epsilon_{k,t} \cdot w_{\psi,k}) + b_\psi \tag{2}$$

$$c(S_\psi, C_t) = \frac{1}{1+e^{-y}} \times 2 - 1$$

A signature S_ψ is reinforced each time the primitive interaction ψ is completed as a success or a fail, using the delta rule (or Least Mean Square method). In the case of a DF, the signature is also reinforced if the associated primitive interaction is enacted. This means that we take the absence of a DF into consideration if the corresponding primitive interaction is enacted. The bias b is equivalent to a weight w_{m+1} related to an input e_{m+1} for which the value is 1 for each step t. The signature is updated as follow:

$$w^t_{\psi,k} = w^{t-1}_{\psi,k} + \alpha \times \epsilon_{k,t-1} \times (\epsilon_{k,t} - c(S_\psi, C_{t-1})) \tag{3}$$

$\forall k \in$ [1; m+1], α the learning rate with $\alpha \in$ [0;1]

Note that the set of weights of a signature of an interaction gives an average pattern of the interactional context that affords this interaction. The signatures thus define the "objects" that afford interactions in the point of view of the agent. The definition of an object thus relies on the interactions predicted as a success in presence of this object and does not imply any internal modelization.

3.2 The Space Memory

We propose a structure that stores, tracks and predicts the results of enacting interactions, we call Spatial Memory, noted M. This mechanism generates a representation of the surrounding environment based on enacted interactions. The memory is composed of a set of nodes that stores and propagates results of previously enacted interactions. This memory thus relates to neural maps [10], but differs first by the fact that movements are encoded in term of interactions rather than actions, and then because the memory stores interactions rather than perceptions. We define a node N as a structure defined by:

- a value $V_{N,t} \in [0; 1]$, that characterizes the certitude of presence of a possibility of interaction.

- a list of interaction Ψ_N, eventually empty.

- a list P_N of previous node $p_{N,\iota}$. Each previous node is associated with a different primitive interaction ι.

The spatial memory is based on the following principle: when an interaction is enacted, the object that affords this interaction is present in a certain (unknown) place of the environment. We thus associate a node to this interaction to store the presence of this object. If the agent enacts a new primitive interaction, the object moves to an other place in the egocentric reference. To track this object, we chain a new node in which the information of presence of the object is stored. The position of this node in space is defined according to the previous node and the last enacted primitive interaction. By adding nodes at each step, it is possible to track the object and define its position with a sequence of node.

We assume that the agent knows that certain nodes are related to a same place of the environment. It is then possible to merge nodes that share a same place if they are related to a same interaction. Note that the absolute position of nodes is not needed. The formed networks define the topology of space based on interactions. The nodes can also be merged if they are afforded by a same (or at least similar) object. It is then possible to define a distance between two interactions relying on a same object with the length of a sequence of interaction defined by the shortest chain of nodes that connects them. Note that two interactions can be associated with a same node if they rely on a same object at a same place in space. We then propose a simplification on the interactional context: if a group of interactions is connected to a same node, the group of interaction can be replaced by the associated node. This simplification allows to reduce the size of the interactional context, and thus, the size of signatures. We use this simplification in our implementations.

The network of nodes defines an occupancy grid [3] in which probability values characterize the certitude of presence of a possibility of interaction rather than a sensor stimulus. The values of nodes of the spatial memory are updated using the algorithm described in Table 1. We note β the vanishing coefficient, $\beta \in$ [0;1]. This coefficient reduce the certitude value of objects at each step, as their positions, defined by nodes, become unreliable.

The propagation of past enacted interactions allows the memory to track objects that escape from the sensory sys-

Table 1: Update algorithm of the spatial memory. We note β the vanishing coefficient, with $\beta \in$ [0;1]

For each node $N \in M$, -update node N with its previous nodes: if $\exists p_{n,\iota} \in P_N / \iota = e_t$, then $V_N^t = Vp_{N,\iota}^{t-1} \times \beta$ else $V_N^t = 0$ -update node N with its associated interactions: if $\exists \psi \in \Psi_N / \epsilon_{\psi,t} > 0$ then $V_N^t = \epsilon_{\psi,t}$

tem of the agent. The spatial memory is also able to predict the environment that may be observed after an intended interaction. This prediction can be simulated by replacing the enacted interaction e_t by a candidate intended interaction $i_{c,t+1}$ in the update algorithm.

3.3 Selection Mechanism

In order to select the next intended interaction i_{t+1}, we propose a selection mechanism based on two decision systems. The first system implements a form of curiosity that leads the agent to learn and reinforce the signatures of its interactions. The second mechanism allows the agent to select interactions in order to maximize the satisfaction value at short and medium term. There are, however, no separated learning and exploitation periods: the two mechanism are used according to the reliability of signatures of interaction. The agent thus keeps the ability to adapts its behavior to environmental changes.

The first mechanism is based on the following rule: we note $c_{\iota,t} = c(S_\iota, C_t)$ the certitude of successfully enact a primitive interaction ι in the interactional context C_t. If the condition (4) has a solutionι_{min}, then the decision mechanism selects this interaction. This mechanism implements a form of curiosity that leads the agent to try interactions for the sake of reinforcing unreliable signatures, regardless of satisfaction values of interactions.

$$\exists \iota_m in / |c_{\iota_{min},t}| = \underset{\iota}{min} |c_{\iota,t}| \wedge |c_{\iota_{min},t}| < \tau \tag{4}$$

with $\tau \in [0;1]$ the learning threashold

Otherwise, the agent uses the second mechanism. This mechanism exploits the predictions given by the spatial memory to determine interactions that can bring possibilities of interactions closer or take them away from the agent. The mechanism then adds weights to interactions according to the movements of possibilities of interactions they produce in the space memory. This selection mechanism is not a path planning mechanism. Indeed, the system takes all of the possibilities of interactions into consideration rather than a unique target, and only computes the next intended interaction.

In order to define the movements of possibilities of interactions, we first define a function υ that computes a value that characterizes the number of possibilities of a given interaction, and their distances to the spatial condition defined by the signature of this interaction. We then define the distance function d in the space memory. The distance is given by the minimal number of interactions needed to move from a node to an other one. We define the function υ as follow:

$$\upsilon(S_\psi, M) = \sum_{N \in M} \sum_{k \in [1;m]} w_{\psi,k} \times V_N \times f(d(N, N_k)) \tag{5}$$

Where f : $\mathbf{R}^+ \rightarrow$]0;1] a strictly positive and strictly decreasing function with f(0)=1 and $\lim_{x \to \infty} f(x) = 0$. This function is used to reduce the influence of distant elements. We note N_k the node connected to the k^{th} interaction of the set Ψ. In our implementations, we use the function:

$$f : x \rightarrow e^{-\gamma \times x} \tag{6}$$

with γ a coefficient that characterizes the influence of objects depending on their distances. As the spatial memory can predict the environment resulting from enaction of a candidate intended interaction, we can measure the average difference of distance of possibilities of interaction before and after the enaction of this interaction. We note M_t the spatial context at step t and $M_{i,t+1}$ the predicted context that may be observed after enacting a candidate intended interaction $i_{c,t+1}$. The global movement of possibilities of interactions that affords an interaction ψ produced by the enaction of $i_{c,t+1}$ is defined with:

$$\Delta\upsilon_\iota(\psi) = \upsilon(S_\psi, M_{i_c,t+1}) - \upsilon(S_\psi, M_t) \tag{7}$$

We then add a utility value to candidate interactions i_c based on the satisfaction value of ψ and the movement of possibilities of interactions that afford it. We define the global satisfaction value of a candidate interaction i_c as:

$$s'_i = s_i + \rho \times \sum_{k \in [1;m]} (\Delta\upsilon_i(\psi) \times s_\psi) \tag{8}$$

with $\rho \in \mathbf{R}^+$ influence coefficient of the SMS

The selection mechanism then selects the interaction i for which the global satisfaction value is the highest among interactions considered as possible (i.e. predicted as a success with a positive certitude).

In our implementation, we propose the following algorithmic simplification: we define the position of each node of the space memory in space. Note that the agent does not have access to these positions. This simplification allows to approximate the distance function with the algebraic distance function rather than computing the minimal sequence that connect two nodes, and thus, strongly reduces the computational cost of the mechanism.

4. IMPLEMENTATION ON ARTIFICIAL AGENTS

We implemented and tested our mechanism on artificial agents evolving in a 2-dimensions continuous and static environment. These agents have a predefined list of seven possible interactions, with predefined satisfaction values (in parenthesis):

- move forward of one step (2)
- bump in a wall (-5)
- eat a prey (50)
- turn right of 90° (-3)
- turn left of 90° (-3)
- turn right of 45° (-3)
- turn left of 45° (-3)

We add a set of visual DF Φ provided by a visual system. This system is composed of a 1-dimension retina, with a visual field of 180°, that can detect colors among (red, green, blue) and measure the optic flow. We consider that a position in the surrounding space can be characterized with the optic flow and the movement provided by an enacted interaction. A DF can then be defined as an interaction that characterizes the presence of a certain color on a certain place in space after enacting a certain primitive interaction. The set Φ is defined such as the DFs are uniformly distributed in space. Note that the positions of DFs are not explicitly defined. In order to respect the assumption that a position is measured with the optic flow, we do not define DF associated with the bump interaction, as it does not produce movement. The DFs have a predefined satisfaction value of zero.

The environment was designed to afford spatial regularities the agent can discover through its interactions. We defined three types of objects, characterized by a color that makes them recognizable using visual DFs:

- The walls (green), that afford the interaction *bump*. The agent cannot move trough a wall.

- The preys (blue), that afford the interaction *eat*. We use the term prey rather than target as these elements are not targets the agent has to reach, but elements that afford a positive interaction, and for which the agent has to define as a target by itself.

- The alga (red). These elements do not afford any interaction. We expect the agent to learn to ignore them.

Note that all of these objects are opaque. The agent cannot perceive an object hidden by an other one.

We first implemented our mechanism on a simulated agent. Both the environment, the agent and the SMS are implemented in Java. Figure 3 gives an example of environment. The environment is composed of a grid for which each cell can contains one of the three object type described previously. The content of these cells can be edited during the experimental run. When the agent reach a prey, this prey

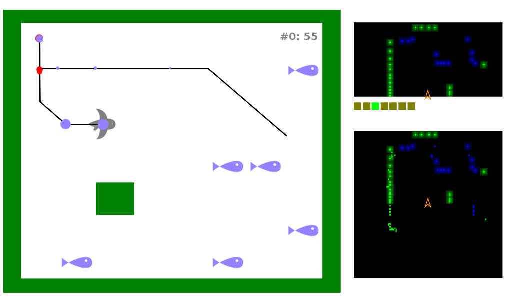

Figure 3: Left: environment of the simulated agent. Preys are represented with blue fishes. The trace shows the last 30 steps. Eat interactions and bump interactions are represented respectively by blue and red dots that shrink at each step. Top right: the interactional context. The inputs related to visual DF are displayed under the form of the image returned by the visual system to improve readability. The seven remaining inputs related to primitive interactions are represented with a row of seven squares. A green square means a success, and a red square, a fail of the corresponding interaction. Bottom right: the spatial Memory. The nodes are organized to match their position in space. The position and orientation of the agent in the spatial memory is given by the orange arrow. Note that the agent ignores its position on the spatial memory.

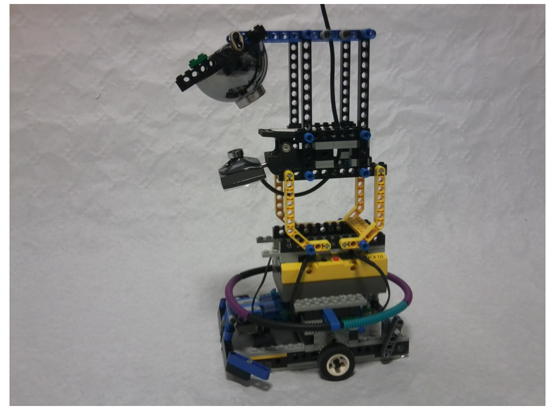

Figure 4: The robot used in our experiments. The design is defined according to the set of interactions of the interactional system. The robot is equipped with a large bumper in front to detect bump interaction, a light sensor beneath to detect eat interaction, and an elevated panoramic camera to provide visual Dynamic Features (DF).

disappears and an other one is randomly placed in an empty cell. The agent is independent from this grid and can move freely in the environment. The agent is represented as a gray shark and preys as blue fishes. We display the trace of the 30 last steps. Bump interactions are displayed with red dots and eat interactions with blue dots. These dots shrink after each step and disappear after 30 steps.

We then implemented the mechanism on a robot. We found that most robots available in the market were not suitable to implement the interactionism approach of an EMDP design. Indeed, they do not fulfill an ecological balance between sensors and actuators. The term of ecological balance was proposed by Pfeifer [15] to refer to the fact that possibilities offered by sensors and possibilities offered by actuators must be well balanced to support a sensorimotor approach. We then design our own robots based on Lego Mindstorm

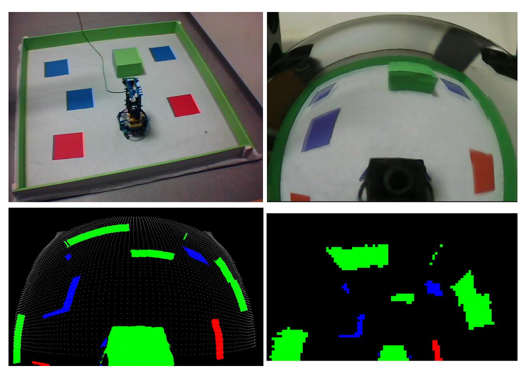

Figure 5: Top left: the robot in its environment. Top right: the environment perceived by the camera. The elements of the environment are recognized by their color and hidden objects are removed (bottom left). The image is then projected (bottom right). The final image is used to define the Dynamic Features (DF).

(Figure 4), that offer a flexibility that allows to define new designs that maintain a balance between actuator and sensors. Our robot is designed according to the proposed set of interactions: the bump interaction is detected using a frontal contact sensor with a bumper that has the same width than the robot. The eat interaction is detected using a light sensor beneath to detect when the robot moves over a colored patch on the floor. The turn interactions do not need sensors. The visual DFs are defined by an elevated panoramic camera that provide a 180° visual field. As the environment is flat, the distance of an object can be determined by its position on the camera image. Note that to respect the opacity of objects, we remove the hidden objects on the final image (Figure 5). The robot is remotely controlled by the same mechanism than for the simulated agent. We implemented an interface in Not Quite C[2] to interpret intended interaction commands from the SMS and return the enacted interaction once the interaction is completed.

5. EXPERIMENTS AND OBSERVATIONS

We propose a set of experiments to test the spatial memory and decision mechanisms through agent's behavior. The first experiment consists in observing the signature learning, and thus, the emergence of a relation between elements of the environment and interactions. The next experiments are provided to test the possibilities offered by the SMS.

5.1 Signature Learning

We let the agent evolves in its environment and observe the evolution of signatures. After 2000 to 3000 steps, the agent begins to navigate efficiently in its environment, moving from a prey to an other one and avoiding walls. A temporary deactivation of the curiosity selection mechanism does not affect the behavior of the agent, which means that the agent has successfully learned to interact with objects of its environment.

The signatures of the simulated system obtained after 3000 steps are shown in figure 6. Note that a signature is a vector. However, for a better readability, the weights are displayed as follow: weights that are related to a visual DF are organized to match their position in space. As we define three DFs per position (one per observable color), we propose to display the three values of DFs of a same position in RGB space. Thus, we define for each position a color defined by the values of the three corresponding DFs. Note that the color channels are attributed to match the color of DFs. The values of the signatures are normalized according to the highest weight value (in absolute values). Middle gray means the three normalized weights are 0, black means they are -1 and white means they are 1. The weights that are related to primitive interactions and the bias input are represented with a row of height squares for which the color is determined by the associated weigh: pure green for 1 and pure red for -1.

By analyzing signatures of interactions, we can observe that the interactions *move forward*, *bump* and *eat* are related to elements that are in front of the agent (figure 6): the *move forward* interaction is related to the absence of wall and prey (mid red blob), the *bump* interaction to the presence of a wall (green blob) and the *eat* interaction to the presence of a prey (blue blob). We can also observe that the *move forward*

[2] http://bricxcc.sourceforge.net/nqc/

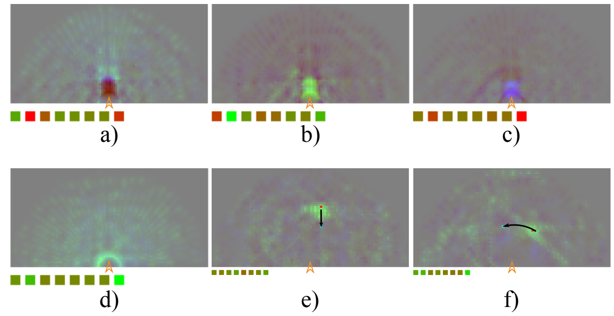

Figure 6: Signatures of Interactions. The top parts represent the weights of visual DFs organized in a topographic way to improve readability. The height squares represent the seven weights associated to the primitive interactions and the bias value. We display the signatures of interactions move forward (a), bump (b), eat (c) and turn right of 90° (d) of the simulated agent. The signatures of turn interactions are similar to the signature of turn right of 90°. e) and f): signatures of DFs corresponding to seeing a green element in the position given by the blue point, by enacting move forward (e) and turn right of 45° (f) interactions. Red points show the maximum value nodes. We observe that these DFs are related to the same element perceived before the enacted interaction (movement of interactions are displayed with arrows).

and *eat* interactions are related to the fail of the interaction *bump*, while the interaction *bump* is related to the success. The *turn interactions*, that cannot fail, are strongly related to the bias input. These signatures show that the agent has defined the objects related to its interactions and the location required to interact with them under the form of structures that gather interactions that allows to perceive them.

We also observe that the DFs are strongly related to other DFs that encode a same element of the environment before the corresponding primitive interaction was enacted (figure 4.e and 4.f). It appears that the links between nodes of the spatial memory may be defined according to these signatures. This observation seems to prove that the spatial memory can be learned rather than being hard-coded.

5.2 Navigation in the Environment

This experiment is designed to test the ability of an agent to navigate in its environment, taking into consideration multiples objects and their positions around the agent, including elements that are out of the agent sensory system. The experiment is conducted on a trained agent as follow: we let the agent discovers a new prey on the other side of the environment. Once the agent begins to move toward the prey, we add wall blocks to hide the prey from the visual system of the agent. We then analyze the behavior of the agent.

In a majority of experimental run, we observe that the agent first turns in the side that allows it to move away the most from the wall. The agent then gets around the wall until it can see the prey again, and moves toward it. The agent is thus still attracted by the prey, that escaped from its sensory system. This behavior illustrates how the SMS works: the agent is strongly attracted by the prey, as it affords a high satisfaction value interaction, and moderately repulsed by the wall. While the agent approaches the wall, the influence of the wall becomes stronger than the prey, which makes the interaction *move forward* (that makes the wall closer) less interesting than turning (that allows to take the wall away). The agent then get around the wall. Once the agent bypasses the wall, *moving forward* allows to take the wall away, which makes the agent moving again toward the prey. Note that most of the time, the agent selects the side with the largest way that leads to the prey. This behavior is observed both with the simulated agent and with the robot (figures 7 and 8). We also observe that if the prey is too close to a wall, or if the way that leads to the prey is too narrow, the agent refuses to move toward the prey.

Figure 7 gives an example of experimental run on the simulated agent. This run is interesting as the agent first turns left to avoid the wall. It then makes an about-turn and then goes trough the right side of the wall to reach the prey. The fact that the agent first turns left shows that it considers that avoiding the wall, which is closer that the prey, is most important than reaching the prey that is on its right side. The agent then uses the right side way as the left one needs to move behind the wall before reaching the prey. A similar behavior is observed with the robot (Figure 8): the agent turns left to align itself toward the prey. Unfortunately, a wall appears and prevents the agent to move forward. The agent turns right to avoid the wall and uses the left side of the wall to reach the prey, as the right side is very narrow.

5.3 Invisible Preys Experiment

This experiment shows how the agent can organize its behavior to interact with elements it cannot perceive, based on surrounding clues. We modify some properties of the prey: they are now invisible, and they do not disappear when the agent reaches them. We organize the environment as shown in figure 9. We place three alga blocks, and add

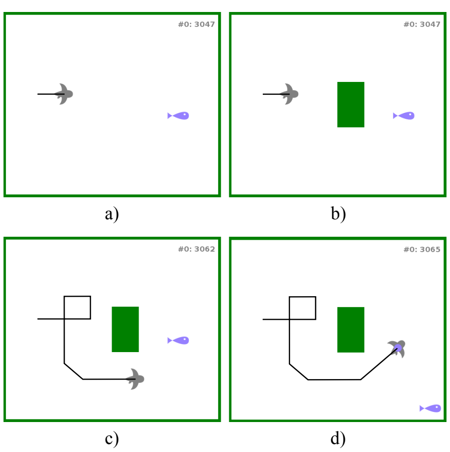

Figure 7: The hidden prey experiment with the simulated agent. a) the agent moves toward a prey. b) A wall is added between the agent and the prey. c) The agent first turns left to avoid the wall. Then, it selects the largest way (right). d) the agent finally captures the prey.

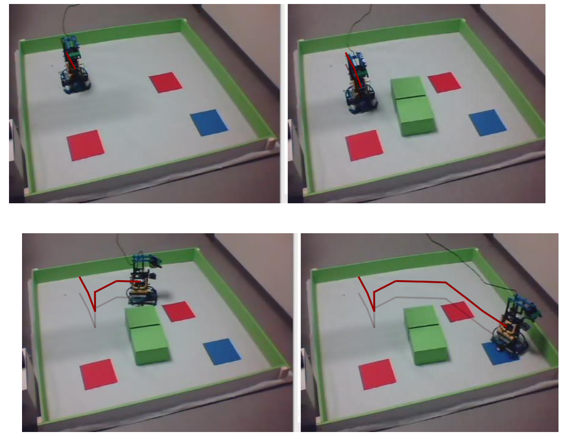

Figure 8: The hidden prey experiment with the robot. The behavior is the same as with the simulated agent. Note that the agent is not influenced by the alga (red square). The agent thus learned to neglect this element.

preys at a constant distance of the alga. Due to the size of the environment needed for this experiment, we do not test it with the robot.

We first used an untrained agent. The learning period is longer than for the normal environment: 5000 simulation steps are needed to obtain an efficient behavior that allows the agent to capture prey. We observe that the signature of the eat interaction contains the presence of an alga at a certain distance of the agent (red half circle on figure 9). This means that the agent has learn that the eat interaction is related to the presence of an alga at a certain distance from the agent. The alga is thus considered as a part of the object that afford the interaction eat. The agent then uses alga as landmarks and turns around them to capture preys (Figure 10). This observation shows that the agent can organize its behavior in space according to spatial clues.

We then used an agent previously trained with visible preys. We observe that the agent rotates, trying to find preys under the form of blue objects. We have to "guide" it toward a group of preys using additional wall blocks to make the agent interact with invisible preys. After 500 steps, the red half circle pattern appears on the signature of the eat interaction (Figure 9). These modifications made the predictions unreliable enough to make the agent using the curiosity decision mechanism. After 1500 steps, the agent turn around alga to capture preys. We can observe on the signature of eat interaction that the agent does not forget the visible prey, represented by the blue blob on Figure 10. We

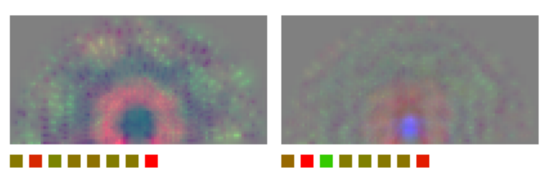

Figure 9: Signature of the eat interaction obtained with an untrained agent (left) and with an agent previously trained with visible preys (right).

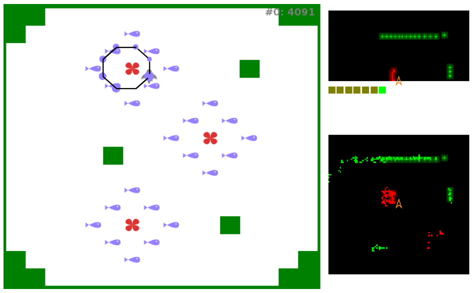

Figure 10: invisible prey experiment: the agent cannot see the preys. The agent learns to use alga (red) as landmarks to capture preys.

suppose that the learning period is shorter than with the untrained agent because of the signatures of other interaction that are still valid in this environment. This experiment with the trained agent shows that it is still able to adapts its definition of objects to take environmental changes into consideration.

6. CONCLUSIONS

We propose to implement a representation of the environment of an artificial agent for which elements and locations are both based on interactions. An agent equipped with this mechanism can generate its own definition of objects, by associating interactions that allow to interact with them, and then, to recognize, localize and track these objects in the surrounding environment without any ontological preconception about these objects. The spatial memory encodes space and distances in term of interactions, and allows the agent to navigate in its environment and moves toward objects without a path planing mechanisms.

The agent constructs its own perception and knowledge of its environment and becomes aware of the elements that compose it. The perception the agent has of its environment consists in interactions that can be predicted as success or fail, and possibilities of interactions that can move toward or away from the agent. This simple perception allows the agent to demonstrate complex spatial behavior taking into consideration multiple objects of the environment.

We used a hard-coded spatial memory with predefined links between nodes, which makes an infringement of the principle of environmental agnosticism. However, the signatures we obtained on DFs suggest that the relation between nodes can be learned. We can note that the algorithmic and computational simplification allowed by organizing the nodes of the spatial memory in a topographic arrangement may explain the existence of inborn cerebral structures that maintain a topolographic representation of space in the brain of vertebrates (such as tectum). Future works will investigate the emergence of the structure of space based on signatures of interactions. We also intend to implement our mechanisms in more complex systems, especially agents using continuous set of interaction.

7. ACKNOWLEDGMENTS

This work received the support the Agence Nationale de la Recherche (ANR) contract ANR-10-PDOC-007-01.

8. REFERENCES

[1] K. Astrom. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, January 1965.

[2] R. M. Cotterill. Cooperation of the basal ganglia, cerebellum, sensory cerebrum and hippocampus: possible implications for cognition, consciousness, intelligence and creativity. *Progress in Neurobiology*, 64(1):1–33 , May 2001.

[3] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57 , June 1989.

[4] S. Gay and O. L. Georgeon. Interaction Based Space Representation for Environment Agnostic Agents. *In proceedings of ALA2013, at AAMAS2013*, 8 pages, May 2013.

[5] S. Gay, O. L. Georgeon, and J. W. Kim. Implementing a spatial awareness in an environment-agnostic agent. *In proceedings of BRIMS2012*, pages 62–69, 2012.

[6] O. L. Georgeon, J. B. Marshall, and R. Manzotti. ECA: an enactivist cognitive architecture based on sensorimotor modeling. *Biologically Inspired Cognitive Architectures*, 6:46–57, September 2013.

[7] O. L. Georgeon and I. Sakellariou. Designing Environment-Agnostic Agents. *In proceedings of ALA2012, at AAMAS2012*, pages 25–32, June 2012.

[8] J. J. Gibson. The theory of affordances. In R. Shaw and J. Bransford (Eds.), Perceiving, acting, and knowing: Toward an ecological psychology (pp. 67-82). Hillsdale, NJ: Erlbaum, 1977.

[9] D. Humes. *Treatise of human nature* Oxford, David Fate Norton and Mary J. Norton, 1739, ed. 2000.

[10] M. G. Lagoudakis and A. S. Maida. Robot Navigation with a Polar Neural Map. *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference*, pages 965–, 1999.

[11] J.-A. Meyer et al. The Psikharpax Project: Towards Building an Artificial Rat. *Robotics and Autonomous Systems*, 50(4):211–223, 2005.

[12] J. Meyer and S. Wilson. Navigation with a rat brain: a neurobiologically-inspired model for robot spatial representation. *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 169–175, 1991.

[13] J. K. O'Regan *Why Red Doesn't Sound Like a Bell: Understanding the feel of consciousness.* Oxford, 2011.

[14] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

[15] R. Pfeifer. Building fungus eaters: Design principles of autonomous agents. *from animals to animats*, 4:3–12, 1996.

[16] J. Piaget. *The construction of reality in the child* New York: Basic Books, 1937/1954.

[17] H. Poincaré. *Une architecture du son - Cahiers 1/2* 1905, Ed. L'Harmattan, 2005, page 322.