Autonomous object modeling based on affordances for spatial organization of behavior

Simon L. Gay
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: simon.gay@liris.cnrs.fr

Olivier L. Georgeon
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: olivier.georgeon@liris.cnrs.fr

Christian Wolf
Université de Lyon, CNRS
INSA-LYON, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: christian.wolf@liris.cnrs.fr

***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 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. Experiments with a simulated agent and a robot show that they learn to navigate in their environment, taking into account multiple surrounding objects, reaching or avoiding objects according to the valence of the interactions that they afford.**

I. 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 environment-agnostic [6]. We base our work on a design principle introduced by Georgeon and Aha, called Radical Interactionism (RI) [3], 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. RI intends to account for cognitive theories that suggest that perception and action are inseparable (i.e. O'Regan [10], Piaget [13]). Specifically, interactions are used to model Piaget's notion of *sensorimotor scheme*.

In this approach, the agent is given a predefined set of uninterpreted interactions associated with predefined valences, and seeks to enact interactions with positive valence and to avoid interactions with negative valences. This motivation principle is called *interaction motivation* [4], and is related to the problem of intrinsic motivation [11]. The agent perceives its environment by identifying affordances proposed by the environment rather than recognizing objects on the basis of predefined features. This approach addresses the knowledge grounding problem [8] by letting knowledge of objects arise from experience, and introduces no disconnection between the agent's experience and the representation of objects.

Our previous implementations[1] have shown that an agent equipped with a sequential RI algorithm was able to autonomously capture and exploit hierarchical sequential regularities afforded by the environment. However these agents were unable to organize their behaviors in space and did not recognize the object permanence [2], ceasing to pursue objects when they escape from the agent's sensory system.

To overcome these limitations, we propose a mechanism that constructs, maintains and exploits a short-term spatial representation of the environment based on interactions. The agent then learns to extract relevant information from this structure to organize its behavior in space. This mechanism is based on a variation of RI design, which adds a structure called Spatial Memory and a set of heuristics to focus on the problem of interpreting and exploiting the memory content rather than constructing the structure of this memory. The utilization of a Spatial Memory is inspired from biology. Indeed, most natural organisms have brain structures that maintain some geometrical correspondence with the animal's local surrounding environment [1]. We do not intend to develop a path planning mechanism, nor a mapping algorithm, but rather a mechanism inspired by simple structures such as tectum of vertebrates, that allows an agent to recognize and localize possibilities of interactions in space, and generates behaviors that satisfy its intrinsic motivation. We tested our mechanisms in a simple environment to observe the emergence of such knowledge in controlled conditions, and analyze this knowledge and its utilization through the agent's behavior.

II. FORMALIZATION OF RADICAL INTERACTIONISM

A RI algorithm begins with a set I of primitive interactions. Each primitive interaction ι is attributed a *valence* v_ι that defines the agent's behavioral preferences: the agent *likes* to enact interactions with positive valence and *dislikes* to enact interactions with negative valence. The principle of the RI

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

mechanism is that the agent selects, at step t, an *intended interaction* ι_i^t, and is informed, at the end of the decision cycle t, of the interaction ι_e^t that was actually enacted. The *enaction* is a success if $\iota_i^t = \iota_e^t$, and a failure otherwise. A RI agent learns to anticipate the results of its interactions, and tries to enact interactions with high valences.

Georgeon, Marshall and Manzotti [5] proposed a Spatial RI design (SRI) as a variation of RI in which the agent is aware of the position of the enacted interaction in egocentric reference, and of the displacement in space provided by the enacted interaction. We define a set *P* of positions *p*, in an egocentric reference, in which an enacted interaction can be located. We note τ_ι the geometrical transformation associated with a primitive interaction ι. The intuition for p is that the agent has sensory information available that helps it situate enacted interactions in space, such as optic flow or interaural time delay. The intuition for τ_ι is that the agent has sensory information available that helps it keep track of its own displacements in space, such as vestibular system of vertebrates. We call *act* a a couple (ι, p) and A the set of possible acts, $A \subseteq I \times P$. The valence v_a of an act a is the valence v_ι of the interaction ι that composes it. SRI adds a structure called *Spatial Memory*, which maintains a correspondence with the environment space structure in egocentric reference. The Spatial Memory consists of a finite topographic structure discretized in a set of positions p of P', with P ⊂ P'. Enacted acts are placed on this structure according to their position p. At each step t, previously enacted acts are updated according to the geometrical transformation $\tau_{\iota_e^t}$ associated with the last enacted interactions ι_e^t. Note that an updated act can be an element of $A' \subseteq I \times P'$. Acts are associated with a recentness value o_a, and removed after a certain number of steps.

We propose that several acts can be simultaneously enacted as a consequence of enacting an intended interaction. In this model, the agent experiments a set of enacted acts $\{e_k\}_t \subseteq A$ at each step t. Another difference with RI is that the agent tries to enact an act rather than an interaction. We note i_t an *intended act*, e_t an *enacted act* and E_t the set of enacted acts. We call this architecture Parallel SRI (figure 1).

The fact that more than one act can result from enaction of a unique intended act means that certain acts are consequences of an other act. We define such an act a *secondary act*, and call an *associated act* the act on which it depends. An act that does not depend on an other act is called a *primary act*. The type of act is defined as follows : if the set $\forall t \in N^+, (\cap_t \{E_t / \iota \in E_t\}) - \{\iota\}$ is empty, the act ι is primary, else it is secondary. We define an alternative act a' of a as $a' \in A / \exists t, a = i_t, a \notin E_t, a' \in E_t$ and Ψ_a the set of alternative acts of a. We define that an act a is completed as a success at step t if $a \in E_t$, and as a failure if $a \notin E_t \wedge \exists a' \in \Psi_a / a' \in E_t$. A secondary act can succed or fail only if its associated act is completed as success.

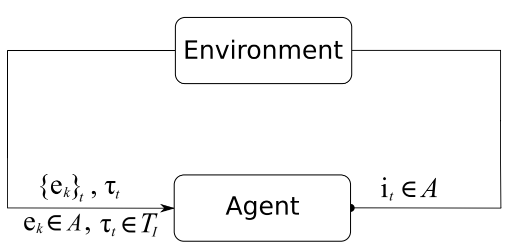

Fig. 1. Diagram of the Parallel Spatial Radical Interactionism model (adapted from the SRI model [5]). At time *t*, the agent tries to enact an intended act i_t, and receives a set of enacted acts $\{e_k\}_t$, called *enacted set* E_t, and the transformation τ_t of the environment, relative to the agent.

III. Spatial Memory System

The spatial memory, once completed with enacted acts, provides an uninterpreted representation of the surrounding environment of the agent. We propose a mechanism, which we called the *Spatial Memory System* (SMS), that helps an agent to define and recognize objects with which it can interact, and to organize its behavior according to its surrounding environment. This section formalizes the concepts used to implement the SMS: signatures of acts, object recognition and decision system.

A. Signatures of Acts

This mechanism is based on the assumption that the result of enacting a certain act depends on a limited spatial context of elements in the environment. We expect such contexts to define the “objects” with which the agent can interact. This definition of objects relates to the concept of *affordances* proposed by J.J. Gibson [7]. An *object* is thus defined as a specific spatial configuration of elements in the environment that *affords* an act, and does not require *a priori* knowledge.

In the RI approach, an agent perceives its environment by “experiencing” it through its interactions. Thus, an enacted act characterizes the existence of an element or a property in the environment. The aim of this mechanism is to define, for each act $a \in A$, a set of acts for which the enaction can characterize the presence of the object that affords a, and thus, to determine when a can be successfully enacted.

The idea of defining objects by learning to recognize affordances they provide is abundant in literature [9], [14]. The mechanism proposed by Uğur et al. [15] is perhaps the most closely related to our mechanism: they propose a model in which artificial agents learn to recognize affordances that allow them to move according to a set of trajectories, based on visual features. Our mechanism differs first because affordance “images” are defined with acts rather than perceptions, which allows implicit relations between acts to be discovered. Then, the agent uses emergent object models to recognize distal objects and spatially organize its behavior according to their positions.

The mechanism is formalized as a function c, called a *certitude function*, which gives, for a given act a and context defined by E_t, the certitude that a can be successfully enacted at step t+1, with an absolute certitude of success if $c(a, E_t) = 1$, and of failure if $c(a, E_t) = -1$. The function is learned and reinforced at each step, based on results of enacted acts.

We implemented the certitude function with an artificial neuron. To this end, the context E_t is coded into a binary vector $[\epsilon_{1,t}; ...; \epsilon_{n,t}]$ of dimension $n = card(A)$, where $\epsilon_{k,t} = 1$ if $a_k \in E_t$ (for all $k \in [1; n]$) and 0 otherwise. Each act $a \in A$ is attributed a set W_a of n weights $w_{a,k}$ and a bias $w_{a,n+1}$. The certitude function is defined with a linear function of inputs, passed through an activation function:

$$c(a, E_t) = g\left((\sum_{k \in [1;n]} \epsilon_{k,t} \cdot w_{a,k}) + w_{a,n+1} \right) \qquad (1)$$

$$g(x) = \frac{2}{1 + e^{-x}} - 1$$

A set W_a is reinforced each time the act a is completed as a success or a failure, using the delta rule (or Least Mean Square method) (2). The bias is related to an input ϵ_{n+1} for which the value is 1 for each step t. We note $r_{a,t} = 1$ if a was successfully enacted at step t and $r_{a,t} = -1$ if a failed.

$$w^t_{a,k} \longleftarrow w^{t-1}_{a,k} + \alpha \times \epsilon_{k,t-1} \times (r_{a,t} - c(a, E_{t-1})) \qquad (2)$$

$\forall k \in$ [1; n+1], α the learning rate with $\alpha \in$ [0;1]. We choose this method for its simplicity and robustness, but also because it allows us to observe how the agent "constructs" objects by analyzing the set of weights of acts. Indeed, a set W_a gives an average pattern of contexts that allow an act a to be enacted. We call a set of weights the *Signature* of an act a, as it characterizes the object that affords this act.

B. Selection Mechanism

We propose a selection mechanism based on two decision systems to select the next intended act i_{t+1}. The *exploration* system implements a form of curiosity that leads the agent to try acts for the sake of learning signatures. The *exploitation* mechanism allows the agent to select acts to maximize valence in the short and medium term. There are, however, no separate learning and exploitation periods: the two mechanisms are used according to the reliability of signatures.

The exploration mechanism allows the agent to test and reinforce signatures when the certitude of an act is low (in absolute value). This mechanism is based on the following rule: at each step t, we note a_{min} the act for which $|c(a_{min}, E_t)|$ is minimum. If $|c(a_{min}, E_t)| < \lambda$, with $\lambda \in [0; 1]$ the *learning threshold*, then the mechanism selects a_{min}. Otherwise, the agent uses the exploitation mechanism. A secondary act can only be tested when its associated act is predicted as a success with a high certitude, as its result depends on the success of enacting its associated act. Note that the more accurate the signatures, the less this mechanism will be used.

The exploitation mechanism considers the relative movement of objects in egocentric reference generated by the enaction of an act. The mechanism then adds a positive utility value to acts which allows to move closer to attractive objects (i.e. defined by an act with a positive valence) and to move away from repulsive objects (and negative utility values in opposite cases). Closest objects have more influence as the agent is more likely to interact with them in the short term.

Recognition and localization of distal objects is based on the following principle: we note $\mathcal{T}(M, \tau)$ an image of the Spatial Memory M when the transformation $\tau \in T$ is applied. We note $E(\mathcal{T}(M, \tau))$ the list of acts stored in $\mathcal{T}(M, \tau)$, limited to the acts of $A = I \times P$. For each geometrical transformation τ, a distal object that affords an act a is considered as present with a certitude of $c(a, E(\mathcal{T}(M, \tau)))$. We call a distal object an *instance* of the object that affords a, localized at τ. We note $d(\tau)$ the distance of the instance. We define the *global proximity value* ψ that consists of a weighted sum of distance of instances of objects that afford a given act, with a higher weight for the closest instances.

$$\psi_(a, M) = \int_{\tau \in T} c(a, E(\mathcal{T}(M, \tau))) \times f(d_{(\tau)}) \qquad (3)$$

Where $f : \mathbf{R}^+ \to]0; 1]$ is a function that characterizes object influence according to their relative distance. In our implementations, we use the function $f : x \to e^{-\gamma \times x}$ where γ is a coefficient that characterizes the decreasing of object influence depending on their distance.

The selection mechanism measures the variation of distance of objects by measuring the variation in global proximity values produced by acts. The mechanism first selects a set of candidate acts i_k with a positive certitude of success. It then generates, for each candidate i_k, an image of the spatial memory by applying the transformation τ_{i_k}, noted $\mathcal{T}(M, \tau_{i_k})$. The variation produced by an intended act i_k is defined as the sum of variation of global proximity of each object, weighted by the valence of the acts it affords:

$$\Delta\psi_{i_k} = \sum_{a \in A} (\psi(a, \mathcal{T}(M, \tau_{i_k})) - \psi(a, M)) \times v_a \qquad (4)$$

The mechanism then selects, among candidates i_k, the next intended act i_{t+1} defined by:

$$i_{t+1} = \max_{i_k} (v_{i_k} + \beta \times \Delta\psi_{i_k}) \qquad (5)$$

where $\beta \in \mathbf{R}^+$ is the influence coefficient of the SMS.

IV. Implementation on Artificial Agents

We implemented and tested our mechanism on agents evolving in a 2-dimensional continuous and static environment. These agents have a predefined list of seven primary interactions, listed below (valences are in parenthesis):

- ▷ move forward by one step (2)
- ▶ bump in a wall (-5)
- ▶ eat a prey (50)
- ◡ turn right by $90°$ (-3)
- ◠ turn left by $90°$ (-3)
- ◸ turn right by $45°$ (-3)
- ◺ turn left by $45°$ (-3)

These interactions cannot be located, and thus, are not placed in the spatial memory. The enaction results are, however, used as inputs by the certitude function.

We add a set of visual secondary interactions provided by a visual system with a visual field of 180°, which can detect colors among {red, green, blue}, and measure distances. We assume that distance is measured through optic flow, while the agent is interacting. Visual interactions consist in seeing a red, green or blue moving element while enacting a primary interaction (except for *bump*, as it does not produce relative movement), for a total of 18 secondary interactions. These interactions have a predefined valence of zero. The visual field is discretized as a grid of 10×20 positions p of P. This system defines $18 \times 10 \times 20 + 7 = 3607$ possible acts.

The Spatial Memory is defined as a finite 2-dimensional surface that contains P, centered on the agent (which ignores its position on the Spatial Memory). The Spatial Memory is discretized as a grid to define positions of P'.

The environment is designed to afford spatial regularities that the agent can discover through its interactions. We defined three types of elements characterized by a color that makes them recognizable with visual interactions:

- walls (green), that afford *bump*.

- preys (blue), that afford *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.

- alga (red), that afford *move forward*, as well as empty spaces. We expect the agent to consider alga as equivalent to empty spaces. Note that all these elements are opaque.

We first implemented our mechanism on a simulated agent. Both the environment, agent and SMS are implemented in Java. The environment's contents can be edited during the experimental run. The agent can move freely and continuously in the environment. When the agent reaches a prey, this prey disappears and another one is randomly added in an empty place. The agent is represented as a gray shark, and preys as blue fishes. We display the trace of the last 30 steps. Figure 2 gives an example of environment.

We then implemented the mechanism on a robot (figure 3). We design our robot based on Lego Mindstorms, which offers a flexibility that allows to define new designs that fulfill an *ecological balance* between sensors and actuators, and to implement the interactionism approach of RI. The term ecological balance was proposed by Pfeifer [12] to refer to

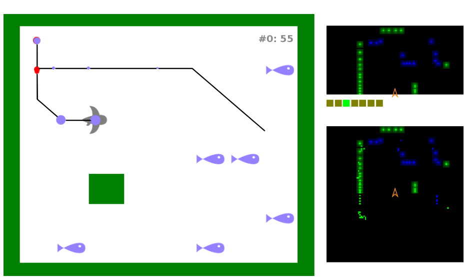

Fig. 2. Left: environment of the simulated agent. The trace shows the last 30 steps. Top right: simplified context. Inputs related to visual acts are displayed according to their positions in space. Inputs related to primary acts are represented with a row of seven squares. A green square means the corresponding act succeeded (here, *eat* is enacted). Bottom right: the Spatial Memory. The position and orientation of the agent in the Spatial Memory is given by an orange arrow. Note that the agent ignores its position.

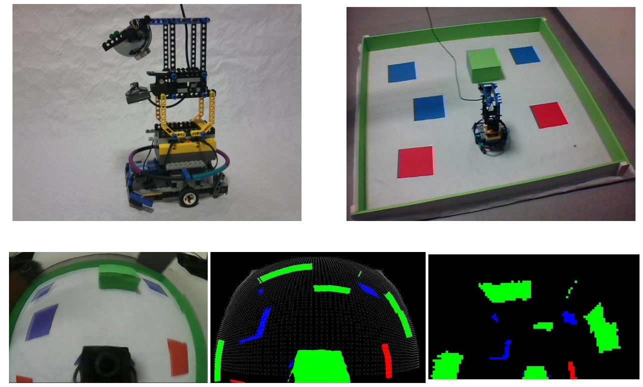

Fig. 3. Top: the robot used in our experiments. The design is defined according to the set of interactions of its interactional system. The environment is similar to the simulated system. Bottom: the visual system. Left: the image provided by the panoramic camera. The image is filtered and hidden elements are removed (middle). The image is then projected (right). The final image is used to define enacted visual acts. Note that the agent can see a part of its "body" (the camera).

the fact that possibilities offered by sensors and by actuators must be well balanced to support a sensorimotor approach. Our robot is thus designed according to the proposed set of interactions: *bump* is detected using a large frontal contact sensor. *Eat* is detected using a light sensor underneath to detect whenever the robot moves over a colored patch on the floor. Visual interactions are defined by an elevated panoramic camera that provides a 180° visual field. As the environment is flat, the position of a visual act can be determined by its position on the camera image. The robot is remotely controlled by the same mechanism as for the simulated agent. The robot model is available online[2].

V. Experiments and Observations

We propose a set of experiments to test the SMS. The first experiment focuses on the signature learning mechanism on the simulated agent. The next experiment focuses on testing the possibilities offered by the SMS through emergent behaviors on the simulated agent and the robot.

A. *Signature Learning*

We let the agent evolve in its environment and observe the evolution of signatures. The signatures of the simulated system obtained after 25 000 simulation steps are shown in Figure 4. We display signature weights as follows: weights related to visual act inputs are organized according to the position of these acts. As there are 18 weights per position (one for each visual interaction), we represent the weights on different layers. Each layer displays weights related to three visual acts based on a same primary act, using the three color channels. The color channels match the color characterized by acts. The intensity shows the values of weights: an intensity of 0 means a weight value of -1, while an intensity of 1 means a value of 1. A gray color means the three weights have a value of 0. The weights related to primary act inputs are displayed as seven squares, and

[2]http://liris.cnrs.fr/simon.gay/index.php?page=eirl&lang=en

an eighth square displays the bias value. For these squares, white means a value of 1 and black a value of -1.

We observe that signatures of primary acts cease evolving after 4000 to 5000 simulation steps. We can observe on these signatures that the acts *move forward*, *bump* and *eat* (Figure 4.a, b and c) are related to elements that are in front of the agent, respectively the absence of wall and prey (dark red blobs), the presence of a wall (green blobs) and the presence of a prey (blue blobs). We also observe that *move forward* and *eat* are related to the *bump* act (second square) with a negative weight, while *bump* is related to itself with a positive weight: indeed, when the agent enacts *bump*, it stays in front of a wall. The turn acts, that cannot fail, are strongly related to the bias. These signatures show that the agent has identified contexts that allows to characterize the presence of objects that afford its acts: the properties of objects, such as their relative positions, sizes and colors, are clearly defined. An other interesting result is that we can observe a similar blob of the same color on each layer of a signature. The signatures thus gather acts for which enaction characterizes the presence of the same object.

After 8000 simulation steps, certain signatures of visual acts show an interesting structure: these acts are strongly related to a group of seeing acts located on the same position in memory (green blobs on Figure 4.d and e). The difference between the position p of a visual act and the position p' of acts designated by its signature corresponds to the transformation provided by this act (white arrows on figure 4.d and e). We believe that these "movements" may open a way on how the agent can learn the transformations provided by enacting acts and connections between positions of space.

B. Navigation in the Environment

This experiment is designed to test the ability of an agent to navigate in its environment, taking into consideration multiple objects.We propose a simplified version of the Spatial Memory System based on the observation that signatures can gather acts that evoke the same act. The simplification is based on the following intuition: when a visual act a is evoked as possible according to the current context, every act designated by the signature of a can be considered as enacted as they are related to the presence of the object that affords a. We thus propose to gather acts related to the same position and color and consider them as a unique act. For example, move forward and see green at position p_1 is equivalent to turn left 90° and see green at p_1. This simplification divides by six the number of visual inputs of signatures and reduces significantly the number of steps needed to obtain accurate signatures, as inputs are activated more frequently. The downside is that we cannot define the result of visual acts, as they can be enacted as a consequence of more than one primary act. However, visual acts do not influence the decision mechanism as their valences are zero. We increase the definition of the visual field to define a grid of 50×100 positions p, and observe more precisely the properties of objects defined by the agent.

We first let the agent move in its environment and learn signatures of acts. After 2000 to 3000 steps, signatures remain stable, and the agent begins to navigate efficiently in its environment, moving from one prey to another and avoiding walls. Deactivating the curiosity selection mechanism does

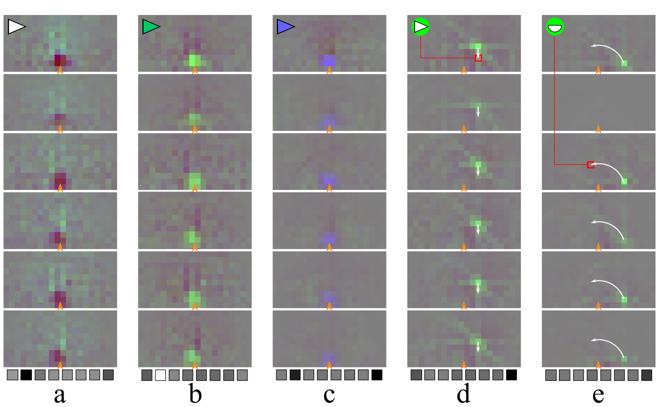

Fig. 4. Signatures of acts move forward (a), bump (b), eat (c), move forward and see green at position p_d (defined by a red square) (d) and turn right by 90° and see green at p_e (e). Weights related to visual acts are placed to match their position in the spatial memory, on 6 different layers. Weights related to primary acts are displayed with seven squares. The eighth square displays the bias value. Orange arrows give the position of the agent. The signatures gather acts related to the same object: we can observe the same blob of the same color on each layer. White arrows on d) and e) show the offset between position of acts and objects that afford them. Note that they correspond to movements of move forward (d) and turn right by 90° (e).

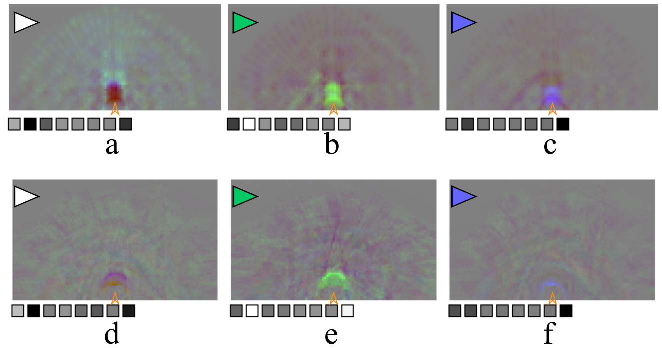

Fig. 5. Signatures of acts move forward (a,d), bump (b,e) and eat (c,f), learned by the simulated agent (top) and the robot (bottom), with the simplified context. Top part of signatures represents weights of visual acts. The height squares represent the seven weights associated to primary act and the bias value. We can observe on signatures defined with the robot that the agent excludes the visible part of its camera.

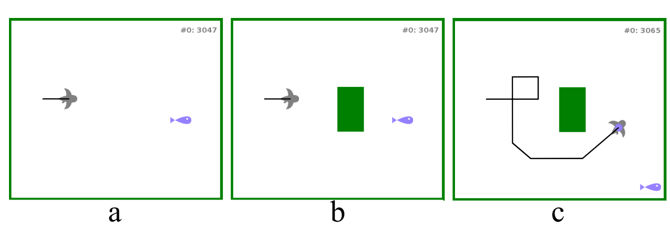

Fig. 6. The "hidden prey" experiment with the simulated agent. a) the agent moves toward a prey. b) A wall is added between the agent and the prey. c) The agent first turns left to avoid the wall. Then, it selects the largest way (right side of the wall) and finally captures the prey.

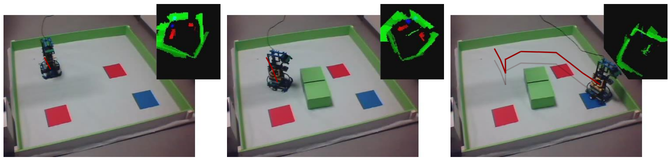

Fig. 7. The "hidden prey" experiment with the robot. The behavior is the same as with the simulated agent. The agent is not influenced by alga (red square): it has thus learned to ignore this element. Note that the mechanism works even with a rough precision from spatial memory.

not affect behavior, as it is mainly driven by the exploitation mechanism. Figure 5 shows the signatures obtained after 2000 steps with the simulated agent and the robot. Signatures are displayed in a similar way to the previous section, except that there are only three groups of weights displayed on a single layer. We can observe that the signatures are similar to the previous experiment. The differences between the signatures learned by the simulated agent and the robot show that the body affects the affordances, and thus the object models constructed by agents.

We then remove all objects in the environment except for a prey. We let the agent discover the prey, then, once it begins to move toward the prey, we add wall blocks to hide the prey. We then analyze the behavior of the agent. In the majority of experimental runs, we observe that the agent first turns to the side that allows it to move furthest away from the wall. The agent then gets around the wall until it can see the prey again, and moves toward it. This behavior illustrates how the SMS works: the agent is strongly attracted by the prey, as it affords an act with a high valence, and is moderately repulsed by the wall. While the agent approaches the wall, the influence of the wall becomes stronger than the prey, which makes *move forward* less interesting than turning. The agent then gets around the wall, and moves toward the prey again. This behavior is observed both with the simulated agent and with the robot (figures 6 and 7).

VI. Conclusion

We implemented a representation of the environment of an artificial agent for which objects are defined according to a predefined set of interactions. An agent equipped with this mechanism can generate its own models of objects, by associating interactions that allow the agent to detect them, and then, recognize, localize and track these objects in the surrounding environment without any ontological preconception about these objects. The spatial memory allows the agent to navigate in its environment, moving toward objects defined as attractive (in the agent's point of view) and avoiding repulsive objects, without path planning mechanisms.

The agent constructs its own perception and knowledge of its environment and becomes "aware" of the elements that compose it. The information provided by the Spatial Memory System consists of interactions that are predicted as a successes or failures, and of interactions that can move recognized objects toward or away from the agent. This simple information allows the agent to demonstrate complex spatial behavior taking into consideration multiple objects in the environment.

This RI mechanism also shed some light on the scalability problem of the environment: regardless of the number of elements present in the environment, the number of objects learned by the agent is limited by the number of acts. Two different elements will be seen as the same *object* if they afford the same act, such as empty places and algas, in our environments, that both afford *move forward*. Size and shape also have no influence while the elements afford an act: long border walls and square walls both provide contexts that afford *bump* and are interpreted as negative elements.

We used a hard-coded and topographic spatial memory, which infringes the principle of environmental agnosticism. However, the signatures we obtained with visual acts suggest that the relation between spatial positions can be learned. Future works will investigate the emergence of the structure of space based on interaction signatures. We also intend to implement our mechanisms in more complex systems, and in particular agents using continuous sets of interaction.

Acknowledgment

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

References

[1] 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, 2001.

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

[3] O. L. Georgeon and D. Aha, *The Radical Interactionism Conceptual Commitment*, Journal of Artificial General Intelligenc, 4(2):31-36, 2013.

[4] O. L. Georgeon, J. B Marshall, and S. Gay, *Interactional motivation in artificial systems: between extrinsic and intrinsic motivation*. In proceedings of the Second International Conference on Development and Learning, and on Epigenetic Robotics (EPIROB), 2012.

[5] 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, 2013.

[6] O. L. Georgeon and I. Sakellariou, *Designing Environment-Agnostic Agents*, In proceedings of ALA2012, Adaptive Learning Agents workshop, at AAMAS 2012, pages 25–32, 2012.

[7] J. J. Gibson, *The theory of affordances*, in Perceiving, Acting, and Knowing: Toward an Ecological Psychology, R. Shaw and J. Bransford, Eds. Hillsdale, pages 67–82, 1977.

[8] S. Harnad, *The symbol grounding problem*, Physica D(42), 335–346,1990.

[9] T. E. Horton, A. Chakraborty, and R. Amant, *Affordances for robots: a brief survey*, Published in Avant, 3(2):preceding p71,2012.

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

[11] 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.

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

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

[14] E. Sahin et al., *To afford or not to afford: A new formalization of affordances towards affordance-based robot control*, Published in Adaptive Behavior, 15(4):447–472, 2007.

[15] E. Uğur, M. R. Doğar, M. Çakmak, and E. Şahin, *The learning and use of traversability affordance using range images on a mobile robot*, IEEE International Conference on Robotics and Automation, pages 1721–1726, 2007.