



HAL
open science

Implementing Spatial Awareness in an Environment-Agnostic Agent

Simon Gay, Olivier Georgeon, Jong Kim

► **To cite this version:**

Simon Gay, Olivier Georgeon, Jong Kim. Implementing Spatial Awareness in an Environment-Agnostic Agent. Conference on Behavior Representation in Modeling and Simulation BRIMS2012, Mar 2012, Amelia Island, Floride, United States. pp.62-69. hal-01352980

HAL Id: hal-01352980

<https://hal.science/hal-01352980>

Submitted on 22 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Implementing Spatial Awareness in an Environment-Agnostic Agent

Simon Gay

Olivier Georgeon

Université de Lyon, CNRS

Université Lyon 1, LIRIS, UMR5205, F-69622, France

simon.gay@alumni.enseiht.fr, olivier.georgeon@liris.cnrs.fr

Jong Wook Kim

University of Central Florida

Orlando, FL 32816

Jong.Kim@ucf.edu

Keywords:

autonomous learning, cognitive development, spatial awareness, object persistence.

ABSTRACT: *We designed an autonomous agent that discovers, learns, and exploits basic spatial regularities of interaction with its environment. To do so, we propose implementing a persistence memory system that records bundles of “possibilities of interaction” afforded by objects in the environment, coupled with a local space memory system that represents the agent’s surrounding local space (inspired by the vertebrate’s tectum). An experiment in a simple simulated environment demonstrates how the agent performs multimodal integration of sensory stimuli, and allocates the origin of such stimuli to “phenomena” located in the external spatial environment. Such mechanisms open the way to implementing agents with minimal preconception of their environment, and to modeling intrinsic motivation in autonomous agents.*

1. Introduction

We address the question of implementing agents with minimal initial preconception of their environment. We define such agents as environmentally agnostic. An environmentally agnostic agent has to autonomously learn to extract relevant information about the environment, and simultaneously organize such information in exploitable knowledge (Georgeon & Ritter, 2011). Environment-agnostic agents are useful to facilitate the development of agent-based models and simulations by reducing the amount of knowledge that must be encoded in the agent. More broadly, studying such agents opens the way to modeling the emergence of new behaviors in autonomous agents.

In previous studies, we started to address this question by implementing an agent that learned hierarchical sequences of behaviors in a bottom-up fashion. To do so, we developed a novel algorithm that we called the intrinsically motivated schema mechanism (Georgeon, Ritter, & Haynes, 2009; Georgeon & Ritter, 2011). With this algorithm, the agent was able to autonomously capture and exploit hierarchical sequential regularities afforded by the environment. This mechanism implemented intrinsic motivation in that the agent’s behavior was driven by predefined low-level behavioral proclivities that gave rise to higher-level behavior. This approach stands in contrast from goal or task-directed navigation algorithms (e.g., Batalin, Sukhatme, & Hattig,

2004; Frommberger, 2008). It also differs from classical reinforcement learning techniques (e.g., Sutton & Barto, 1998) in that it addresses the question of developmental learning (i.e., fast learning during the agent’s development) (e.g., Lungarella, Metta, Pfeifer, & Sandini, 2003) rather than learning over many trials (often thousands in classical reinforcement learning). In particular, our agent received no predefined reward when a final goal was achieved and we did not implement backward propagation of a reward value.

A subsequent study (Georgeon, Cordier, & Cohen, 2011) showed that an agent equipped with such a sequential learning mechanism was able to acquire basic navigation skills in an open space environment. This study, however, also showed the limits of this purely sequential approach when applied to spatial regularity learning. For example, the agent was unable to notice that two different sequences of movement may lead to the same point in space. Moreover, the agent was unable to discover the persistence of objects. The agent stopped pursuing a target of interest when the target was lost by the sensors (hidden or out of span). To overcome these kinds of limits and to move on toward higher-level learning, we now address the question of the autonomous discovery of spatial regularities. We refer to this issue as implementing mechanisms of spatial awareness in an environmentally agnostic agent.

Our mechanism, a spatial awareness mechanism in an

environment-agnostic agent, is biologically inspired. Most natural organisms have some brain structures whose activation maintains some geometrical correspondence with the animal’s local surrounding environment. We refer to the mushroom body in the case of insects, and the tectum in the case of vertebrates, also called the colliculus in the case of mammals (e.g., Cotterill, 2001).

In this study, we advocate implementing two initial mechanisms: the *persistence memory* and the *local space memory*. The persistence memory is a long-term memory that memorizes associations of interactions and stimuli based on their co-occurrence. We name such associations by the term *bundle*. This term refers to pragmatic epistemology (e.g., Hume, 1739) that postulates that the knowledge of objects is constructed through usage rather than given a priori. In this framework, Hume proposed the bundle theory of objects. This theory posits that objects consist only of the collection of their properties observed through interaction. Accordingly, we expect our agent’s bundles to represent objects in the environment in the form of possibilities of interaction. The second mechanism, the local space memory, is inspired by the tectum in the vertebrate’s brain, and consists in an internal geometrical counterpart of the surrounding environment. These two mechanisms constitute the agent’s *spatial system*. The spatial system has two objectives: it allows the agent to perform a spatially-organized multimodal integration of sensory stimuli, and it makes the agent able to project the consequences of its actions in an egocentric referential, possibly beyond the range of the agent’s perception.

We propose a design methodology that begins by indulging some hard-coded preconceptions to get the spatial system running. In this first step, we setup and demonstrate the coupling between the intrinsically motivated sequential system and the spatial system. The second step consists of progressively removing the preconceptions from the spatial system in order to move toward an agent as much agnostic as possible. Following this approach, we organized the paper in two parts. The first part (Section 2) presents the initial experiment made with the hard-coded spatial system. This experiment illustrates how the agent works. From the lessons learned in this initial experiment, we list the infringements of the principle of agnosticism that need to be addressed. The second part (Sections 3 and 4) reports our algorithms that start addressing these infringements. Finally, the paper discusses our results and draws recommendations for future work.

2. Initial experiment

We implemented an autonomous agent in the environment shown in Figure 1. The agent is represented as a shark. So

far, the environment is static: the agent is the only thing that moves. The agent has four primitive *possibilities of action*: (a) move forward (approximately one grid unit), (b) turn approximately $\pi/4$ to the left, (c) turn approximately $\pi/4$ to the right, and (d) eat fish. The agent moves freely in a continuous space and gets uneven from the grid. The agent has a visual system of 12 pixels covering a total span of π radian. Each pixel reports the dominant color seen in its $\pi/12$ corresponding span. The filled cells and the surrounding perimeter represent walls where the agent would bump if it tries to move through them. The agent cannot see through objects (wall, alga or fish). The agent also has a “9-pixels” tactile system (a 3x3 matrix) that detects adjacent objects or objects below the agent (alga or fish). Fish and alga feel *soft*. Walls feel *hard*.

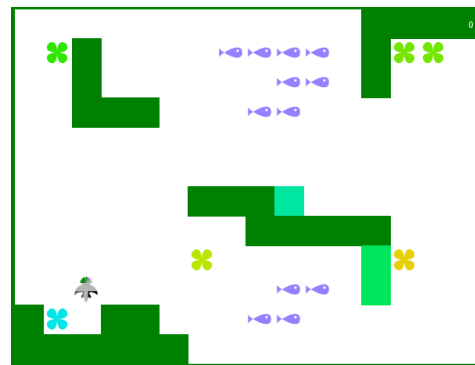


Figure 1: The agent in the environment.

The agent’s behavior is generated by the intrinsically motivated sequence learning mechanism described in our previous work (Georgeon & Ritter, 2011). We implemented the local space memory with a radius of 2 grid units. When a co-occurrence of two or more stimuli is detected, the bundle formed by these stimuli is constructed in persistence memory and a pointer is placed in the local space memory to follow the relative displacement of this bundle when the agent moves.

2.1 Analysis of an example run

A representative run can be seen online (Georgeon, 2011). The first two hundred steps of this run are represented in Figure 2, using a technique of activity trace representation developed in a previous study (Georgeon, Mille, Bellet, Mathern & Ritter, 2011). The various tapes show the sensory and internal state of the agent at each step, as described next. A step is a cycle of interaction between the agent and the environment.

In Figure 2, Tape 1 represents the agent’s tactile system (light gray: touching nothing, intermediary gray: touching *soft*, black: touching *hard*). The touching in front of the

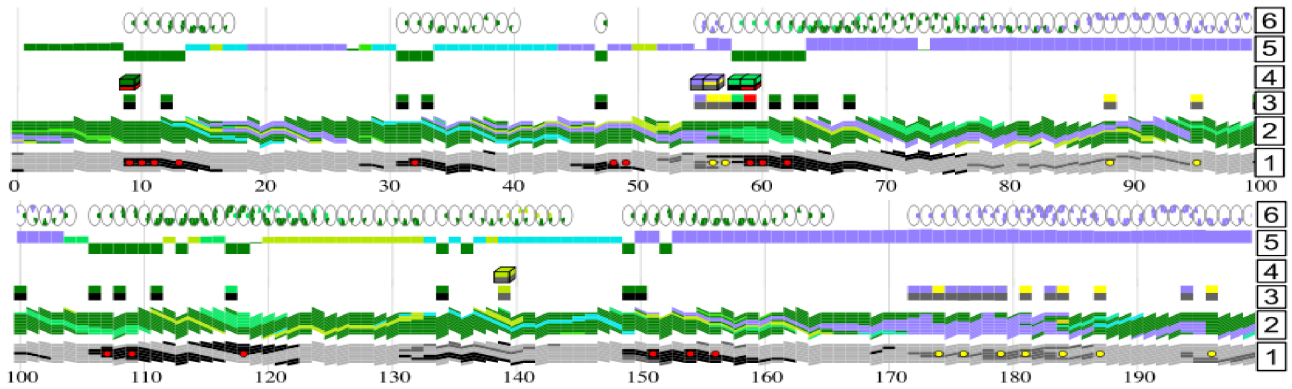


Figure 2: Example activity trace (upper part: steps 0-100, lower part: steps 100-200).

agent is represented in the center of the tape. The touching to the agent's left side in the upper part, and to the right side in the lower part. The touchings below and behind the agent are not represented. Additionally, red circles represent bumping walls, and yellow circles represent eating fish. The agent experiments bumping on dark green walls on steps 9 through 13, and then significantly reduces bumping on such walls. Similarly, the agent learns bumping on light green walls on steps 59 and 60.

Tape 2 represents the agent's visual perception. The twelve visual pixels are represented vertically as rectangles when the agent moves forward, and as trapezoids when the agent turns. This tape shows colored object traversing the visual field as the agent moves and turns.

Tape 3 represents the co-occurrences of interactions from different sensory modalities. For example, on step 55, the gray square associated with the blue square represents the co-occurrence of touching and seeing a fish. On step 56, the gray square associated with the yellow square represents the co-occurrence of touching a fish and eating a fish. Co-occurrences trigger the construction of bundles, or either confirm or infirm existing bundles.

Tape 4 represents the construction of bundles from co-occurrences. For example, on step 9, the agent constructs a bundle made of the association of touching, seeing, and bumping a wall (green, black, and red cube). On step 55, the agent constructs the bundle of seeing and touching a fish (gray and blue cube). On step 56, the interaction of eating is added to this bundle (gray, blue, and yellow cube).

Tape 5 shows a bar-graph whose color represents the focus of the agent's current *attention*, and whose value represents the associated attractiveness (positive or negative). For example, during steps 1 through 8, the agent focuses on the dark green color just because it is the

preeminent color in its visual field. This color has a mild attractiveness because the agent has not yet learned how to interact with it. On step 9, the agent associates this color with bumping, which makes this color become repulsive. Conversely, on step 56, the blue color becomes highly attractive when it gets associated with eating a fish. At any point in time, the most attractive or most repulsive bundle in the surrounding space wins the agent's attention. The agent has a proclivity to move toward the object of current attention if it has a positive attractiveness, or to turn away from the object of current attention if it has a negative attractiveness.

Tape 6 represents the agent's local space memory, that is, the memory of bundles surrounding the agent. The agent's surrounding space is represented as an ellipse, with the front of the agent being on the right. For example, on step 9, the green area in the ellipse represents the agent being *aware* of the wall in front of it. On steps 188 and 189, the trace shows that the agent does not see nor touch any fish, but it is still aware of a fish on its rear-right side (blue area in the ellipse). This awareness causes the agent to turn back towards that fish and eat it on step 196.

In summary, this experiment illustrates how we envision implementing spatial awareness in an intrinsically motivated agent. The agent was able to gradually learn the associations of possibilities of interaction afforded by persistent objects in the environment. Such associations were memorized internally in the form of bundles. Bundles have a value (attractiveness) related to the possibilities of interaction that they afford. The agent maintains a memory of the position of bundles in local space memory. This memory, in turn, impacts the agent's sequential behavior.

2.2 Infringements of the principle of agnosticism

When implementing this agent, we had to hard code some presuppositions about the coupling between the agent and the environment. We hard coded what co-occurrences

were possibly interesting:

a.1 Bumping while touching something with the frontal tactile sensor.

a.2 Eating while touching something with the tactile sensor below the agent.

a.3 Touching something with the frontal tactile sensor while seeing a specific color within the two central visual pixels.

a.4 Eating while seeing a specific color within the two central visual pixels.

We hard coded the agent's knowledge of its basic geometry:

b.1 The position of the tactile sensors in the egocentric reference.

b.2 The maximum radius of the local space memory (2 grid units).

We hard coded the consequences of the agent's actions in the local space memory:

c.1 The move forward action generates a translation of one grid unit in the local space memory.

c.2 The turn action generates a rotation of $\pi/4$ in the local space memory.

To move towards agnostic agents, such hard coded presuppositions should be replaced with autonomous learning mechanisms. To address this concern, we implemented the autonomous learning algorithms described in the next section.

3. Learning correspondence between sensors, actuators, and space

By implementing an algorithm that autonomously learns the correspondence between the values returned by sensors, the agent's actions, and the surrounding local space, we aim at developing a general space-aware system as independent as possible from its sensory and motor configuration. To do so, we first address presupposition b.1 by implementing an algorithm to learn the correspondence between sensors and local space (Section 3.1). Then, we address presuppositions c.1 and c.2 by implementing an algorithm to learn the geometrical transformations that apply to the local space memory depending on each possible action (Section 3.2).

3.1 Correspondence between sensors and local space

We called this algorithm the *sensor mapping algorithm* that learns the correspondence between the values returned by the sensors and the agent's surrounding local space. This algorithm relates to existing algorithms that

allow robots to exploit uninterpreted sensors (e.g., Pierce & Kuipers, 1997). Its specificity, however, is that it constructs a representation of how the sensors cover the surrounding space. We call this representation the *sensory space structure*.

The sensor mapping algorithm requires that it exists a correspondence between the values returned by sensors and individual points in the surrounding local space. Specifically, it is intended to work with sensors that return rough information on the distance of the detected object. Examples of such sensory systems are a stereoscopic visual system that returns approximate distance and color for each pixel, a sonar system that returns distance and echoic property, and a whiskers system (vibrissa) that returns approximate distance and tactile property. We formalize such a requirement as follows:

Each sensory *modality* consists of a set of directional *probes* (e.g., a single whisker, or a "light cone" generating a "pixel" in a visual system). The positions and directions of probes are initially unknown. The probes may not be straight but they must be fixed with regard to the agent (if not, the algorithm must be run for each configuration of the probes). Each probe returns two numerical values: A (abscissa) and S (stimulus). The value A reflects the position of the first object detected along the probe (the object's abscissa along the probe). The only condition on this abscissa is of being a monotonic function of the distance of the object from the agent. This condition needs to consider the fact that objects may mask other objects behind them. The metrics of the abscissa is, however, unknown and may not be linear. These metrics may not be consistent across probes and modalities. The value S reflects a physical property of the detected object (e.g., the color for vision, the tactile feeling for touch), or absence of object (e.g., touch nothing).

This set of assumptions indicates that each tuple [*probe*, *abscissa*] corresponds to a single *Point of detection* (P_d) in the agent's surrounding space. Each P_d in the environment is represented by a *Point of sensation* (P_s) in the sensory space structure. A point of sensation is said active if the value A returned by the corresponding *probe* is greater than the point's *abscissa*. The sensor mapping algorithm gradually adjusts the positions of P_s s in the sensory space structure to reflect the actual positions of P_d s in the environment, starting from any arbitrary configuration (random or implementing an inborn assumption). It relies upon the assumption that the distance between two points is proportional to the average delay between changes of activity or S value of the corresponding probe at each of these two points. The P_s s are placed to optimize the consistency between the delays in the changes of values and P_s 's distance in the sensory space structure.

The pseudo code of this algorithm is summarized in Table 1. This version of the algorithm only considers the abscissa value of sensors. A point of sensation is denoted by p_i , and a probe of the sensory system S is denoted by P . The set of point is denoted by E and the subset of point that belong to the probe P is denoted by E_p , and the output value of the probe P is denoted by A_p .

We define the constant *timer* as the maximum delay for which the variations of two points are considered related to the same event. Constants n and ε define the precision of the system.

To each point p_i , we attribute a position (x_i, y_i) in the sensory space structure, an internal value a_i corresponding to the activity, a sensor value threshold v_i , corresponding to the point's *abscissa* value, a timer value T_i and two lists related to links between points. The first list, D_i , stores the estimated distances of each other point, the second list, C_i , stores the confidence (number of measured delays) of each link.

Table 1: The pseudo code of *sensory mapping*.

```

initialization:
for each probe P, define a subset of point of sensation.
For each point  $p_i$  of  $E$ , set a random position

repeat:
  for each probe P, for each point  $p_i$  of  $E_p$ , do
     $a_i(t-1) = a_i(t)$ 
    if  $A_p > v_i$  then  $a_i(t)=1$  else  $a_i(t)=0$ 
  end for

  for each point  $p_i$  of  $E$ , do
    if  $a_i(t) \neq a_i(t-1)$  then
       $T_i = \text{timer}$ 
      for each point  $p_j, j \neq i$ , do
        if  $T_i[j] > 0$  then
           $D_i[j] = (D_i[p_j] * C_i[p_j] + (\text{timer} - T_i)) / (C_i[p_j] + 1)$ 
          increment  $C_i[p_j]$ 
        else  $T_{p_i} = \max(0, T_i - 1)$ 
      end for

    repeat n time:
      for each couple of points  $(p_i, p_j), i \neq j$  do
         $\Delta = \varepsilon * (D_i[p_j] - d(p_i, p_j)) * (1 - 1/(C_i[p_j] + 1))$ 
        move point  $p_i$  according to  $\Delta * p_i p_j / \|p_i p_j\|$ 
        move point  $p_j$  according to  $-\Delta * p_i p_j / \|p_i p_j\|$ 
      end repeat
    end repeat
  end repeat
end repeat

```

After initializing the system by defining the set of point of sensation, the main loop is composed of three steps. The first step consists of reading sensors and determining the internal value of points. The second step estimates distances between points: if the internal value of a point has changed, we compare the timer value to other points. The average delay measured for each other point is considered as proportional with the distance from these points. Then, the third step updates the position of points to match the estimated distances.

Once the sensory space structure is learned, the agent can localize a place in the environment as the *origin* of the stimulus. Because the metrics of the sensory space structure does not rely on the metrics of the abscissa of the probes, the localization of the origin is consistent across modalities. Therefore, the sensor mapping algorithm supports a spatially-organized multimodal integration of stimuli. For example, the agent can determine that the origin of a specific tactile stimulus *soft* and the origin of a specific visual stimulus *green* are located at the same place in the environment. Allocating an origin to stimuli implies assuming that stimuli have a *cause* in the environment. Such cause can be called a *phenomenon*, typically defined as any observable occurrence. To external observers, these phenomena correspond to physical objects in the environment (e.g., walls, algae, fish). The agent, however, does not see these objects as we see them, nor does it allocate them the same utility as we do, which is why we refer to the objects as phenomena from the agent's viewpoint.

The agent can construct an *origin map* that represents the location of the phenomena in the environment. Such an origin map is, however, not enough to have an operational representation of the environment. Additionally, the agent needs to learn the relation between its motor actions and the origin map. This question is addressed next.

3.2 Correspondence between actions and local space

This algorithm is called the *motion mapping algorithm*, which learns the correspondence between the actions of the agent and the geometrical transformations in the agent's origin map. This algorithm addresses presuppositions c.1 and c.2 introduced in Section 2.2.

The motion mapping algorithm consists first of computing a vector field that describes the relative movements of the *origins* in the origin map when the agent moves. This vector field can be thought of as an "optic flow" (Figure 4) in the image (Figure 5c) made of the origin map. Because the resolution of this image may be low, we used an algorithm inspired by the *insect eye algorithm* developed by Franceschini, Pichon and Blane (1992). This algorithm estimates the movement by

measuring the time between variations of values in two adjacent points of the image. Note that the goal of the motion mapping algorithm is not to anticipate complex consequences of actions in the environment (such as the trajectory of objects in motion) nor to allow complex navigation and localization in space (e.g., Mataric, Meyer, & Wilson, 2009; Meyer, Guillot, Khamassi, Pirim, & Berthoz, 2005), but only to learn the relation between primitive actions and the local space.

We assume that the agent cannot move its body parts but can only move as a whole block in a two-dimensional environment. With this assumption, the agent’s actions can be expressed as the sum of a translation and a rotation. Consequently, the resulting geometrical transformations in the local space memory consist of the sum of a translation and a rotation in the opposite direction. The algorithm computes the value of this translation and this rotation by measuring the average translation and rotation in the vector field.

Once the translation and rotation values are known, the agent can apply them to the origin map to follow up the relative positions of phenomena when the agent moves. This approach is related to map learning algorithms based on occupancy grids (Elfes, 1989). For example, if a wall is on the agent’s left side, and the agent makes a rotation step to right, then the agent knows that the wall moved behind, even though the agent cannot see the wall anymore. Moreover, the origin map is now related to the agent’s possibilities of actions. For example, the agent can estimate the distance of phenomena in terms of the actions needed to reach them.

3.3 Bundle construction

As noted in Section 3.1, origin maps are consistent across sensory modalities because they are based on delays during movements rather than on the metrics of sensors. Therefore, the agent can infer that different stimuli from different sensory modalities are “caused” by the same phenomenon when the origins of such stimuli overlap. The agent creates a bundle to represent the set of the different interactions afforded by this phenomenon. Once learned, the bundles are memorized in persistence memory, and can be subsequently recognized. For example, the agent creates the bundle of touching *hard* from the tactile map, and seeing *dark green*, from the visual map to represent the phenomenon “wall”. This mechanism eliminates presupposition a.3. The agent can then subsequently enrich the “wall bundle” by adding the interaction “bump”. Then, the sequence learning mechanism will cause the agent to avoid walls when it recognizes them, as reported in Section 2.

4. Second experiment

We implemented these algorithms in a similar agent as presented in Section 2, in the same environment as in Figure 1. We, however, modified the visual and the tactile system to provide more precise input to the algorithms. The visual system now has a resolution of 5° over a total span of 180° (36 “pixels”). Each pixel returns the color and the distance of the detected object. The tactile system is composed of 18 whiskers distributed all around the agent, with a resolution of 3 points of detection on each whisker at a distance of 0.5, 1, and 1.5 grid units from the center of the agent (Figure 3.a).

4.1 Sensor mapping

Figure 3 allows a comparison of the actual points of detection (Figure 3.a) with the learned points of sensation in the tactile sensory space structure (Figure 3.b). This result was obtained after 1000 steps starting from an initial condition where the individual points of sensation were placed randomly (independently from the whisker to which they belong). This result shows that the agent was able to approximately learn the configuration of its whiskers. The whiskers, however, appear “shrunked”. We believe that the precision on the whiskers’ length could be improved by considering the movement of the agent learned from the motion mapping algorithm. This would involve interweaving the sensor mapping algorithm with the motion mapping algorithm in the future developments.

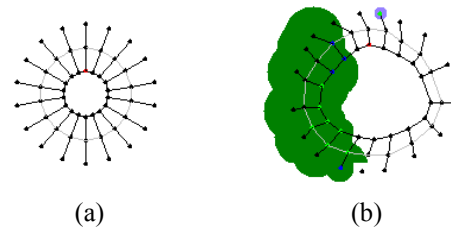


Figure 3: the real tactile system (a) and the sensory space structure (b) given after 1000 steps. Black radial lines represent whiskers and the black circle line represents the whiskers’ basis.

Figure 3b also provides a representation of an instance of the tactile origin map when the agent is sensing a “wall phenomenon” on its left side (dark green) and a “fish phenomenon” on its front (blue).

4.2 Motion mapping

Figure 4 reports examples of vector fields computed by the motion mapping algorithm applied to the visual system. To obtain these results, we, however, hard coded the visual sensory space structure rather than learning it

with the sensor mapping algorithm. As noted in Section 4.1, merging these two algorithms remains a challenge that we plan to address in future studies.

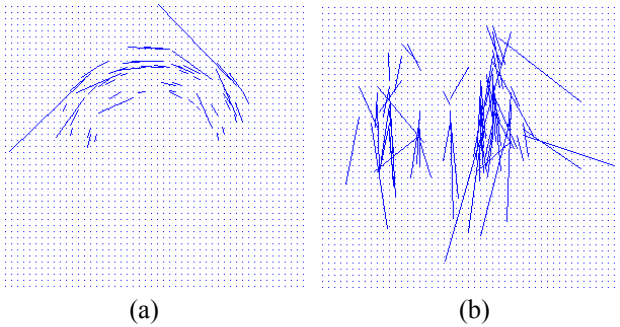


Figure 4: average movement flow given by the visual system, for a rotation (a) and a translation (b).

Figure 5b shows an instance of the origin map for the tactile system, using whiskers with 15 points of detection each. In this figure, colored areas represent the origin of tactile stimuli: touching soft (light gray), touching hard (dark gray), empty (white), and black areas indicate untouched areas.

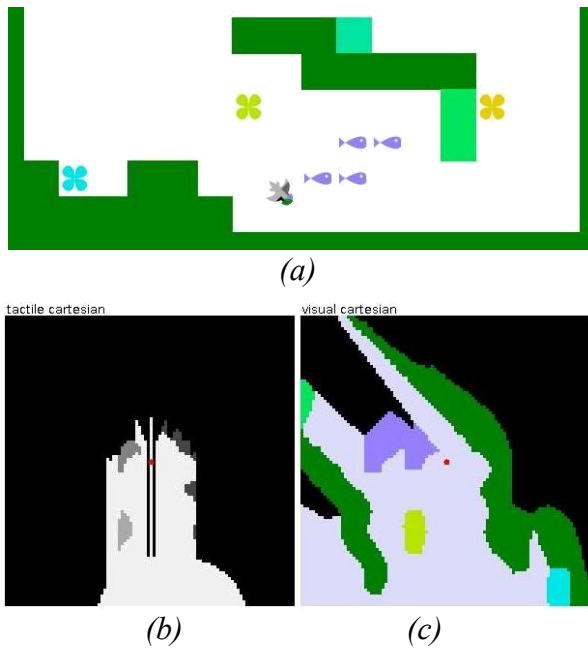


Figure 5: figure (a) shows the actual position of the agent in the environment. (b) map represents the tactile origin map, (c) the visual origin map. The red point shows the position of the agent in its own egocentric reference, the front of the agent is on the top.

Figure 5.c shows the instance of the origin map for the visual system in the same situation. In both of these

figures, the agent's location is represented by a red point and the agent's front is oriented upwards. More precisely, the agent keeps track of the probabilities of different phenomena at each location but the colors in the figures only represent the most likely phenomenon at each location. Figure 5.a represents the corresponding situation of the agent in the environment. Figure 5 shows that this mechanism provides the agent with a sense of persistence of phenomena: in the lower part of the visual map, the yellow and blue phenomena are still present in the visual origin map while being outside of the agent's visual span.

4.3 Bundle construction

As introduced in Section 3.3, bundles are constructed when different sensory stimuli have overlapping origins. Figure 6 illustrates this mechanism. Figure 6.a summarizes the bundles constructed in this instance by associating visual stimuli (x axis) with tactile stimuli (y axis): empty, hard, soft, and edible.

Figure 6.b shows the agent's local space memory in the same instance as presented in Figure 5. In Figure 6.b, bundles are represented by their colors but actually are multimodal representations of phenomena that are spatially localized in the agent's surrounding local space.

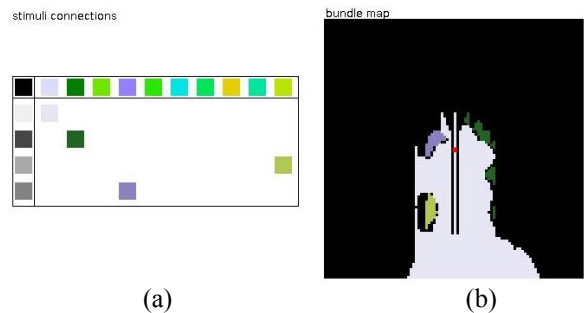


Figure 6: Bundle construction. (a) composition of bundles, y axis: empty, hard, soft, edible, x axis: color. (b) local space memory containing bundles represented by their color.

5. Discussion and Conclusion

We propose the implementation of a spatial system to enable an autonomous agent to keep track of objects in its environment. Such system improves the agent's ability to construct increasingly elaborated behaviors. This implementation is part of an ongoing study of how an intrinsically motivated agent becomes aware of the world in which it exists. We believe that the algorithms presented here shed some light on this question by illustrating the relations between the capacity of an agent to orient itself in space and its capacity to allocate a "cause" to its perceptions in the world (phenomena). In

particular, this work confirms the importance of time, delays, and sequences in a cognitive system, as many recent studies tend to show (e.g., Nicoletis, 2011). All our algorithms involve time: the sensor mapping algorithm constructs spatial dependencies from temporal dependencies, the motion mapping algorithm learns relations between actions and space, and the agent's decision process is based on sequence learning. We argue for a methodology relying on techniques of activity trace analysis to study temporal dependencies in a cognitive system.

More practically, this work opens the way to modeling agent's behavior without having to program specific behavioral rules and predefined sensors. This will facilitate agent modeling in the future, and will facilitate studies on the emergence of complex behaviors.

Our current implementation, however, still has limitations. One limitation is that the sensor mapping algorithm and the motion mapping algorithm remain to be merged together. The agent should simultaneously learn the consequences of its actions and the structure of its sensory system. We believe that the separation of these algorithms causes imprecision in the whole process that still prevented us from being able to set up a comprehensive experiment to demonstrate the overall improvement of the agent's behavior. Another limitation is that some hard-coded presuppositions still remain. Specifically, presuppositions a.1 and a.2 require creating bundles by associating tactile or visual stimulations with active interactions such as bumping or eating. Addressing this limitation requires taking vision and touch as active processes and merging the control of these processes with the intrinsically motivated sequence learning mechanism. We plan on addressing these questions in future work.

6. Acknowledgment

This work was supported by the Agence Nationale de la Recherche (ANR) contract ANR-10-PDOC-007-01.

7. References

- Batalin, M. A., Sukhatme, G. S., & Hattig, M. (2004). Mobile robot navigation using a sensor network. In *Proceedings of the Conference IEEE International Conference on Robotics and Automation*, 636-642.
- Cotterill, R. M. (2001). 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.
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22 (6), 46-57.
- Franceschini, N., Pichon, J. M., & Blanes C. (1992). From insect vision to robot vision. *Philosophical Transactions: Biological Sciences*, 337 (1281), 283-294.
- Frommberger, L. (2008). Learning to behave in space: a qualitative spatial representation for robot navigation with reinforcement learning. *International Journal on Artificial Intelligence Tools*, 17 (3), 465.
- Georgeon, O. (2011). Autonomous learning through experience. Retrieved from the web November 10th, 2011. <http://e-ernest.blogspot.com/2011/11/ernest-104.html>
- Georgeon, O.L., Cohen M., & Cordier, A. (2011). A model and simulation of early-stage vision as a developmental sensorimotor process. In *Proceedings of Artificial Intelligence, Applications and Innovations (AIAI 2011)*, 15-18.
- Georgeon, O. L., Mille, A., Bellet, T., Mathern, B., & Ritter, F. E. (2011). Supporting activity modelling from activity traces. *Expert Systems*, in press.
- Georgeon, O.L. & Ritter, F.E. (2011). An intrinsically-motivated schema mechanism to model and simulate emergent cognition. *Cognitive Systems Research*, in press.
- Georgeon, O.L., Ritter, F.E., & Haynes, S.R. (2009). Modeling bottom-up learning from activity in soar. In *Proceedings of the 18th Annual Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, 09-BRIMS-016, 65-72.
- Hume, D. (1739, ed. 2000). *Treatise of Human Nature*. Oxford: David Fate Norton & Mary J. Norton.
- Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15, 151-190.
- Mataric, M. J., Meyer, J., & Wilson, S. (2009). Navigating with a rat brain: a neurobiology-inspired model for robot spacial representation. In *Proceedings of the 1st Conference on From Animals to Animats*, 169-175.
- Meyer, J.A., Guillot, A., Khamassi, M., Pirim, P., & Berthoz, A. (2005). The Psikharpax project: towards building an artificial rat. *Robotics and Autonomous Systems*, 50 (4), 211-223.
- Nicoletis M. (2011). *Beyond boundaries: The new neuroscience of connecting brains with machines — And how it will change our lives*, New York: Times Books.
- Pierce, D., & Kuipers, B. (1997). Map learning with uninterpreted sensor and effectors. *Artificial Intelligence*, 92, 169-229.
- Sutton, R.S., & Barto, A.G. (1998). *Reinforcement learning: An introduction*, Cambridge, MA.