



HAL
open science

Bootstrapping perception with action: a case for sensory prediction

Jean Godon, Sylvain Argentieri, Bruno Gas

► **To cite this version:**

Jean Godon, Sylvain Argentieri, Bruno Gas. Bootstrapping perception with action: a case for sensory prediction. 2020. hal-02596671v1

HAL Id: hal-02596671

<https://hal.science/hal-02596671v1>

Preprint submitted on 15 May 2020 (v1), last revised 31 Oct 2020 (v2)

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.

Bootstrapping perception with action: a case for sensory prediction

Jean Godon, Sylvain Argentiari and Bruno Gas

Abstract—Recent developments have argued that perception can emerge from sensation when (and to an extent *only* when) actions are considered. This problem of Developmental Robotics is studied by drawing from the Sensorimotor Contingencies (SMC) theory. Specifically, an extended formalism for describing sensorimotor interaction is proposed, allowing to account for motor and sensory variations. Said formalism is then used to study the properties of sensory prediction in a naive agent, with two main results. First, spatial contingencies make such a prediction internally computable *via* a particular class of actions. Second, sensory prediction is shown to capture the structure of motor capabilities of the agent. Both those claims are further illustrated in a simulation. This provides a generalized approach to bootstrap the development of perception for naive agents, which can tackle a larger class of problems.

Index Terms—Sensory prediction, sensorimotor contingencies, interactive perception, bootstrapping, developmental robotics.

I. INTRODUCTION

HOW can an artificial agent discover and build cognitive capabilities by itself so it can face realistic and unpredictable environments? This key capability is one central question when trying to build autonomous robots. One solution consists in taking into account the physical properties of the interaction of the agent with such environments, e.g. by relying on a priori models, either of the agent’s body, its sensors, or of the agent world. Such conventional approaches are by definition relying on well designed models of the agent interaction for the assumed task it has to solve, like the navigation in a clustered environment, or the recognition of a sensory pattern. But such models are notoriously difficult to obtain [1], by definition incomplete [2], and certainly not generalizable to interactions varying in unknown spatial and temporal scales. Importantly, perceptive capabilities are of tremendous importance when dealing with such problems as they represent the interface between the low level sensations generated by the agent sensors and its higher level cognitive layers. The ability of the agent to act accordingly to its environment depends on it correctly learning its perception. But while this ability has been for long considered as a simple, *a priori* interpretation of the agent sensor’s outputs, the role of action in this process is now hard to discard. Multiple cues argue for a more central role of action from the agent in the building of its perception capabilities, to a point where talking about perception without action might be considered incorrect [3].

The field of interactive perception [4] emerged recently as a set of work dealing with this relation between action and perception. It can be seen as a topic regrouping low level considerations like embodied sensorimotor approaches to perception as well as higher ones like autonomous affordances discovering [5] or control strategies. This paper belongs to the former approaches, mostly along the Sensorimotor Contingencies Theory (SMC) [6] stating that perception is the mastery of invariant structures or regularities in the sensorimotor flow discovered during the interaction of the agent with its environment. We already proposed different contributions in this field, successively dealing with peripersonal space characterization [7], self-contact and body representation [8], and the emergence of a topological representation of sensors poses [9], everytime from the sensorimotor flow only. All these contributions tried to demonstrate how internal (mathematical) structures in the agent motor configuration (namely, motor quotient spaces and their corresponding topologies), discovered thanks to sensory invariance, are related to external, mostly spatial, properties of the agent interaction with its environment. However these works can only be applied to acquire local spatial knowledge and skills, and cannot be used with agents able to move freely in their environment. The approach proposed here generalizes these concepts by introducing a variational approach, i.e. variations of the agent motor configurations, as a way to leverage spatial regularities of the interaction to learn sensory prediction.

Sensory prediction is a topic that has been extensively covered in neuroscience and cognitive psychology. Many physiological experiments ([10], and later [11], [12]) described how it might be implemented via motor *effeference copies* coupled with forward sensory models and how it supported perceptual processes such as gaze stability. On the other hand, *ideomotor theory* [13] goes as far as stating actions are in fact represented in the mind as their perceptual consequences. This has naturally lead to sensory prediction being used in robotics, especially in cognitive robotics and grounded cognition [14]. Predictive Coding [15] constitutes one way to tackle this question, with the idea that the brain is constantly building and updating a predictive mental model of its sensorimotor experience. The ability to predict the sensory consequence of an action has been used in motor control [16]; it can be leveraged as a way of learning efficient representations of motor sequence [17], and it was also shown to lead to a sensorimotor notion of object [18] on top of which higher cognitive constructs can be built. Other works dealt instead with the matter of learning this ability: neural networks have been shown to be able to learn sensory predictions in a variety

of experiments [19], [20]. It should be noted that both of these contributions learned to relate *changes* in sensory states and in motor configurations.

Interestingly, sensory prediction has already been studied in the context of SMC theory [21], with the idea that the regularities of the latter could assist in the former. Our work follows the same approach, showing how *space-like* regularity properties in the experience of the agent allow for sensory prediction to be learned. This is done in the context of very simple models which are shown to be sufficient to pick up on the phenomenon and therefore illustrate precisely which structure is actually captured by the aforementioned neural networks. Moreover it shows how both sides of sensory prediction –learning it and using it to structure action– can be performed by a naive agent, therefore providing a new perspective for bootstrapping in the context of SMC theory.

This paper is organized as follows. To begin with, we introduce in Section II all the notations and concepts needed to exploit the proposed variational approach. On this basis, we leverage these elements in Section III to enunciate properties of the subsequent sensorimotor interaction. In particular, we demonstrate the equivalence between the structure of sensory predictions and actions. Then, some simulations are proposed in Section IV to assess the mathematical formalism, through a careful evaluation of each step of the proposed framework. Finally, a conclusion ends the paper.

II. DEFINING A FORMALISM FOR SENSORIMOTOR INTERACTION

This first section aims at expanding several previous results in *Interactive Perception* as obtained for example in [4]. These have made use of several classical objects such as the *pose* (or *working*) *space* and the *forward* (either geometrical or sensory) *maps*, at times rearranging their definitions or making them more precise to allow for formal proofs to be derived. Such work is followed upon in this contribution, with a somewhat significant overhaul of the formal definitions. This section is thus devoted to the definitions of the terms we will use to describe a sensorimotor problem, showing during the exposition how they appear in a simple classical example and how they differ from previous theoretical formulations. We then leverage these definitions to propose and prove new perceptive bootstrapping algorithms in the following section.

A. Motor actions

As a first step, this subsection is devoted to the introduction of all the notions and definitions of the *motor* side of the proposed sensorimotor framework. After highlighting the limitations of the previous approaches, we show how to reparameterize the sensorimotor interaction by introducing motor actions. Their definition and properties are then carefully discussed.

1) *A look back to previous formalisms*: Let us consider in all the following an agent endowed with motor and sensing capabilities. The *internal* motor configuration \mathbf{m} and the sensory configuration \mathbf{s} of this agent lie respectively into some sets \mathcal{M} and \mathcal{S} . Both of them define the *internal* agent

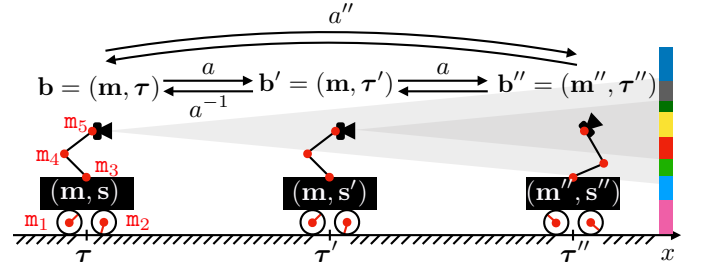


Figure 1: Illustration of the motor actions effects. The internal agent configuration is entirely described by its sensorimotor flow (\mathbf{m}, \mathbf{s}) , while its external configuration is given by (\mathbf{b}, \mathbf{s}) , partially unknown to the agent. An action a acts on a configuration \mathbf{b} so that the agent motor & sensory configurations and its position in space is changed.

configuration (\mathbf{m}, \mathbf{s}) , i.e. the sensorimotor flow the agent has access to. There is a clear dependency between the sensory and motor configurations that can be captured by the sensorimotor maps $s : \mathcal{M} \times \mathcal{E} \rightarrow \mathcal{S}$, such that $s(\mathbf{m}, \epsilon) = \mathbf{s}$, where $\epsilon \in \mathcal{E}$ represents the state of the environment. As said in the introduction, other contributions already exploited this kind of parameterization [22], [7], [8]. In all these contributions, only “grounded” agents are considered, since a single internal motor configuration $\mathbf{m} \in \mathcal{M}$ is only mapped to a single sensory configuration $\mathbf{s} \in \mathcal{S}$ for a fixed environment configuration ϵ .

To illustrate this point, Figure 1 represents a 2D-agent able to translate itself only along one dimension x . This agent is able to move inside an environment made of colored walls thanks to 5 rotating joints whose states m_i , $i = 1, \dots, 5$, are captured in its motor configuration $\mathbf{m} = (m_i)_i$. To begin, let us consider the case where m_1 and m_2 are fixed, so that the agent is only able to move its arm supporting a camera-like sensor generating a sensation \mathbf{s} , i.e. \mathbf{m} is restricted to (m_3, m_4, m_5) only. In such a scenario, one has a grounded agent for which each motor configuration \mathbf{m} can be mapped to one corresponding sensor pose, which is itself mapped to a sensation \mathbf{s} . This simple statement allows to build structures in \mathcal{M} by exploiting only the sensorimotor flow (\mathbf{m}, \mathbf{s}) , structures that can be leveraged to build an internal representation of the agent body, or even of its peripersonal space. In other words, \mathbf{m} contains all of the information, although sometimes with some redundancy, about the coupling between the agent and its environment. However, what would happen if the same agent was able to perform translations in its environment? This case is illustrated in Figure 1 when all motor states m_i are used. Indeed, one can imagine a case where the agent moves in its environment along the x axis from (external) position τ (with internal configuration \mathbf{m}) to τ' (same \mathbf{m}). In this case, the sensor samples two different parts of the color wall so that its generated sensations \mathbf{s} and \mathbf{s}' from these two different positions are different. Then two identical internal configurations \mathbf{m} give two different sensations: there is no more mapping between \mathbf{m} and \mathbf{s} , and all the mathematical developments performed in previous works can no longer apply. Therefore, it seems necessary to generalize these formalisms to cope with agents able to move freely in their environment. In this paper, one proposes a variational formulation of motor actions to deal with this issue. It is introduced in the next subsections.

2) *Foregoing grounding : reparameterizing the sensorimotor interaction*: From previous arguments, the internal motor configuration \mathbf{m} can not be mapped unambiguously to sensations without additional considerations. If one still insists on having a functional relation between motor data and sensations, one then needs to enrich the initial motor set. In this paper, one proposes to introduce some superset \mathcal{B} of \mathcal{M} as initial parameter space. This new set \mathcal{B} can be thought of as the set of all *absolute* configurations \mathbf{b} made of pairs (\mathbf{m}, τ) where \mathbf{m} is the internal motor configuration and τ represents an absolute measure of the pose of the agent in its ambient space (which would most commonly be position and orientation in 3D space). It is important to understand that the agent itself has no knowledge of the current absolute configuration \mathbf{b} of its interaction with its environment, retaining the same hypotheses about a priori structure. However we may then consider the sensorimotor map as a function $s : \mathcal{B} \times \mathcal{E} \rightarrow \mathcal{S}$ instead of $s : \mathcal{M} \times \mathcal{E} \rightarrow \mathcal{S}$ to account for possible displacements in the environment. Defining such a new set \mathcal{B} allows then to introduce the notion of *external* agent configuration as the tuple (\mathbf{b}, \mathbf{s}) . As such, two different points of view must be stressed out: (i) the external point of view (i.e. coming from the designer of the system) will allow to characterize some properties of the agent interaction with its environment (through modelization, hypotheses, etc.), and (ii) the internal point of view which represents which data and concepts are available to the agent for its operations. This specific point is discussed in §III-A.

Coming back to Figure 1, the agent moves to three successive absolute configurations \mathbf{b} , \mathbf{b}' and \mathbf{b}'' . All of them are now different, which was not the case of the internal motor configurations: introducing $\mathbf{b} \in \mathcal{B}$ apparently solves the issue raised at the end of §II-A1. Let us now explain how the agent actually reaches some given absolute configuration \mathbf{b} .

3) *Going variational : introducing motor actions*: As explained previously, the agent has no direct access to the configuration data \mathbf{b} : it cannot know where it *is* in \mathcal{B} . Instead we suppose it starts with some (very limited) knowledge of how it *moves* in this set, i.e. it is capable of performing some moves in \mathcal{B} and of comparing any two moves for equality. To this end, we propose to introduce some new set \mathcal{A} behaving in the following manner: an element $a \in \mathcal{A}$ can be applied to any absolute configuration $\mathbf{b} \in \mathcal{B}$ to give a new configuration $\mathbf{b}' = a\mathbf{b} = a(\mathbf{b})$. Therefore, a can be seen as a function $\mathcal{B} \rightarrow \mathcal{B}$. We will usually denote $\mathbf{b} \xrightarrow{a} \mathbf{b}'$ this situation, and call a a *motor action*. Now as we intend to represent the way in which the agent can move in its environment, one can take for granted the existence of a special action $e \in \mathcal{A}$ that verifies $\forall \mathbf{b} \in \mathcal{B}, \mathbf{b} \xrightarrow{e} \mathbf{b}$: the agent may decide to stay still. Moreover, considering it is able to do any moves a and a' , it may then chain them in one single move $a'' = a'a \in \mathcal{A}$ which satisfies

$$\forall \mathbf{b} \in \mathcal{B}, \quad \begin{array}{ccc} \mathbf{b} & \xrightarrow{a''} & \mathbf{b}'' \\ & \searrow^a \quad \nearrow^{a'} & \\ & \mathbf{b}' & \end{array} \quad (1)$$

so that \mathcal{A} naturally carries the structure of a *monoid*. In the following, we will further restrict ourselves to the case where individual actions are *reversible*, that is for any action a there exists an action a^{-1} such that

$$\forall \mathbf{b} \in \mathcal{B}, \quad \mathbf{b} \xrightleftharpoons[a^{-1}]{a} \mathbf{b}', \quad (2)$$

making \mathcal{A} into a *group*. Seeing as how actions can be thought of as mappings $\mathcal{B} \rightarrow \mathcal{B}$, a necessary (and sufficient) condition is for all mappings in \mathcal{A} to be bijective.

Figure 1 illustrates these notions, with the agent moving from external configuration \mathbf{b} to \mathbf{b}' through an action a . This action, applied at $\mathbf{b} = (\mathbf{m}, \tau)$, happens to produce a translation of the agent so that its internal motor configuration finishes at the same \mathbf{m} . Note that the agent would be able to return back to its initial absolute configuration by applying the inverse action a^{-1} of a . Moreover, since a is a function defined on the whole \mathcal{B} set, the same action can be applied at $\mathbf{b}' = (\mathbf{m}, \tau')$ to reach a third configuration $\mathbf{b}'' = (\mathbf{m}'', \tau'')$. This time, the same action a has conducted to a global displacement of the agent in the environment, combined with a change in its internal motor configuration. Indeed, while it represents cases which are mostly avoided for practical reasons, it is not required for a to only depend on \mathbf{m} in the general case: the outcome of the same action a may depend on the position τ of the agent in the environment. Finally, the agent would have been able to move from \mathbf{b} to \mathbf{b}'' by applying the action $a'' = a^2$, as per Equation (1).

With these structure assumptions, for a given subset of *motor primitives* $\mathcal{A}' \subset \mathcal{A}$ available to the agent, we can search for the set of composed moves the agent can actually reach by iteration of its known ones. We shall say an action $a \in \mathcal{A}$ *decomposes over* $\mathcal{A}' = \{a_i\}_{i \in I}$ if it can be written in the form

$$a = a_{i_n} \dots a_{i_1} = \prod_{1 \leq k \leq n} a_{i_k}, \quad i_k \in I \quad (3)$$

The interest of these decompositions appears because the effect of composed moves on motor configurations boils down to the effects of its components as per the following diagram:

$$\forall \mathbf{b} \in \mathcal{B}, \quad \begin{array}{ccc} \mathbf{b} & \xrightarrow{a} & \mathbf{b}' \\ & \searrow^{a_{i_1}} & \nearrow^{a_{i_n}} \\ & \mathbf{b}_1 \xrightarrow{a_{i_2}} \dots \xrightarrow{a_{i_{n-1}}} \mathbf{b}_{n-1} & \end{array} \quad (4)$$

In the example in Figure 1, it may well be that the agent can move to any configuration \mathbf{b} i.e. that its action set is $\mathcal{A} = \mathbb{R}^5$ (for the 5 possible angular increments of its 5 joints). But it may also be restricted to a limited set of moves, for example if it only can send discrete commands to its joints. For instance, if each actuator is a stepper motor, then its action set turns into $\mathcal{A} = \mathbb{Z}^5$. In this case, a would be written as the tuple $(\Delta q_i)_i, i = 1, \dots, 5$, where $\Delta q_i \in \mathbb{Z}$ is the i^{th} motor increment expressed in step increments. Consequently, any action a would decompose over $\mathcal{A}' = \{a_i\}_i$ where action a_i corresponds to adding one step to the i^{th} actuator. In this specific case, while \mathcal{A} is infinite, it is sufficient for the agent to know the 5 motor primitives a_i to generate any action $a \in \mathcal{A}$.

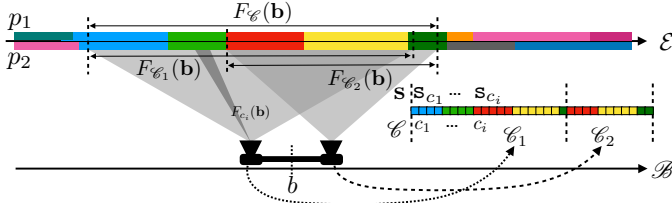


Figure 2: Illustration of the receptive fields for a sensor made of two rigidly linked cameras at configuration \mathbf{b} . Each pixel c_i of either camera produces a sensory value s_{c_i} in the overall sensory array \mathbf{s} explained only by a small subset of space $F_{c_i}(\mathbf{b})$. The same applies for both cameras, thus explaining how a sensation for the agent can be explained by the perception of a subset of space.

B. Grounding sensations in space

The previous subsection was devoted to the introduction of actions on the motor side of the proposed sensorimotor framework. This subsection accordingly deals with the *sensory* side of it, and more particularly with its relation to a persistent “space” which was entirely absent from previous considerations. After a more precise definition of the meaning of “environment configuration”, the link between local perception and spatial considerations is formalized. This will constitute the root of the theoretical developments proposed in the next section.

1) *Decoupling environment and its underlying layer*: In previous works, a traditional way for parameterizing the environment was to introduce the environment configuration ϵ . The meaning of such a variable was often fuzzy, almost without any formal semantics linking it to the sensorimotor experience of the agent. In this paper, and maybe more in line with classical “physical” formalisms, it is proposed to stress the difference between the ambient geometrical *space*—in which sensorimotor experience occurs—and the *environment* itself—that is the state of “things” lying in this space—. The former takes the form of some set \mathcal{E} endowed with a spatial structure as encoded by a group $\mathcal{G}(\mathcal{E})$ of *admissible transformations*. These spatial transformations are mappings $\mathcal{E} \rightarrow \mathcal{E}$ preserving some “geometry” of \mathcal{E} . The most common illustration is the usual affine geometry of \mathbb{R}^3 given by the group $\text{SE}_3(\mathbb{R}) = \text{SO}_3(\mathbb{R}) \rtimes \mathbb{R}^3$ of its rigid transformations, made of 3D rotations, translations and their compositions. On this basis, one chooses to define a “state of the environment” as a valuation that maps each point of \mathcal{E} to its corresponding physical properties such as temperature, color, luminance, etc. These states are therefore best represented as functions $p : \mathcal{E} \rightarrow \mathcal{P}$ where \mathcal{P} is a set describing the different physical properties the agent can observe. Consequently, $p(\mathbf{e})$ represents the observable physical properties at point $\mathbf{e} \in \mathcal{E}$. We will henceforth denote \mathcal{P} the set of *environment states*, i.e. a set of such functions p .

Figure 2 illustrates these considerations. In this simple case, the geometrical space \mathcal{E} is monodimensional, represented as an axis where each point \mathbf{e} is assigned a color through a function p_1 or p_2 . Interestingly, one can now distinguish points in \mathcal{E} for which different environment states locally coincide, as far as the agent is able to observe this coincidence. This trivial—but important—statement is discussed in the following.

2) *Local perception and receptive fields*: Now that we have formally defined what is “out there” from an external point of view, let us now focus on the sensory capabilities of the agent. Recall that we considered in §II-A2 the sensorimotor map as a function $s : \mathcal{B} \times \mathcal{E} \rightarrow \mathcal{S}$. From previous considerations, this map can now be generalized as a function $s_c : \mathcal{B} \times \mathcal{P} \rightarrow \mathcal{S}_c$ to account for the environment physical state p . The new notation s_c also outlines that the sensory map is explicitly written for a sensory element c (or *sensel*, i.e. one pixel for a camera, the cochlea cell coding for one sound frequency, etc.). Thus, the sensorimotor map $s_{\mathcal{C}}$ for the entire sensory apparatus is made of the aggregate of all sensels along $s_{\mathcal{C}} : \mathcal{B} \times \mathcal{P} \rightarrow \mathcal{S} = \prod_{c \in \mathcal{C}} \mathcal{S}_c$ with \mathcal{C} the set of all sensels¹. An illustration of these points is proposed in Figure 2 for an agent endowed with two cameras. In this case, the sensels c_i —each depicted as elements in a color array— represent the pixels of either camera. Separate sensors in the apparatus thus appear as sub-arrays in \mathcal{C} : the first (resp. second) camera is figured by \mathcal{C}_1 (resp. \mathcal{C}_2). Note that this decomposition of \mathcal{C} as $\mathcal{C}_1 \cup \mathcal{C}_2$ directly comes from our external understanding of the agent structure (i.e. with one camera corresponding to one set of sensels, i.e. one sensor). One could have selected others sub-arrays to form a distinct set of (virtual) sensors not necessarily corresponding to their (physical) implementation on the agent.

Let us now focus on sensors $\mathcal{C}' \subset \mathcal{C}$ that satisfy the following locality principle: for any given absolute configuration $\mathbf{b} \in \mathcal{B}$ their output only depend on a certain *subspace* $F_{\mathcal{C}'}(\mathbf{b}) \subset \mathcal{E}$ as per the following property

$$\begin{aligned} \forall p_1, p_2 \in \mathcal{P}, \forall \mathbf{b} \in \mathcal{B}, \\ p_1|_{F_{\mathcal{C}'}(\mathbf{b})} = p_2|_{F_{\mathcal{C}'}(\mathbf{b})} \Rightarrow s_{\mathcal{C}'}(\mathbf{b}, p_1) = s_{\mathcal{C}'}(\mathbf{b}, p_2). \end{aligned} \quad (5)$$

In this equation, $F_{\mathcal{C}'}$ is a map intending to mirror the well studied concept of *receptive fields* of the sensor \mathcal{C}' , that is the region of space determining the sensory output of \mathcal{C}' . Figure 2 represents some of the receptive fields for the two cameras agent. The first one, $F_{c_i}(\mathbf{b})$, is the receptive field of a single sensel/pixel $c_i \in \mathcal{C}$. The receptive fields $F_{\mathcal{C}_1}(\mathbf{b})$ and $F_{\mathcal{C}_2}(\mathbf{b})$ of each camera can be obtained as the union of the receptive field $F_{c_j}(\mathbf{b})$ of their respective pixels. In the same vein, the overall receptive field of the agent $F_{\mathcal{C}}(\mathbf{b})$ is also given by $F_{\mathcal{C}_1}(\mathbf{b}) \cup F_{\mathcal{C}_2}(\mathbf{b})$. From the same figure, it is clear that even if $p_1 \neq p_2$ (since there are areas of different colors on the \mathcal{E} axis), the sensation captured by the agent is the same since the aforementioned differences are restricted to areas of space unseen to the agent.

It is important to notice that this is the formal step where the notion of receptive field formalizes an implicit relation between the sensations of the agent and spatial information. This constitutes one fundamental property sufficient to leverage spatial knowledge from the agent interaction with its environment. The application of these theoretical elements is proposed in the next section.

¹In all the following, the map $s_{\mathcal{C}}$ will be shortened to s when there is no ambiguity, consistently with the initial definition of the sensorimotor map of the agent recalled in §II-A1.

III. A ZERO-TH LAYER OF SENSORIMOTOR CONTINGENCIES: SPATIAL REGULARITIES THROUGH VARIATIONS

In this section, we proceed by describing how the formal elements from Section II can be arranged to enunciate some interesting properties of the sensorimotor interaction. First, to keep in line with considerations of minimalist bootstrapping, the model of knowledge of the agent is discussed. Then the interactions between the objects defined earlier are outlined, with some preliminary conditions, definitions and results made formal. Finally, a novel learning process is proposed for what we argue to be the class of *most basic* agents, allowing for the internal representation of their initial motor structure. The corresponding results are of two distinct but equally important natures: some, taking the viewpoint of an external observer, assert that certain particular objects of interest (such as a *sensory prediction function*) exist; others guarantee these objects to be *computable* in the boundaries set by our model of knowledge. This is made in an effort to keep *a priori* knowledge to a minimum, and these proofs are generally of a constructive nature. They are both formulated for general agents as defined by the previous section, before being assessed on a particular toy case in the next section as proof of concept.

A. How can it be so “naive”?

In the authors previous works [9], sensorimotor interaction occurred as a sequence of (generally discrete) steps where at each point, the agent can access both its proprioception $\mathbf{m} \in \mathcal{M}$ (seen as an array of current joint configuration states) and its corresponding exteroceptive array $\mathbf{s} = s(\mathbf{m}, \epsilon)$. These sensory arrays are then compared for equality *as total vectors*, that is the agent may not access the vectors component by component. This knowledge is then used for example to compute set-theoretic motor kernels which are shown to be a structural invariant of the sensorimotor interaction. By contrast, in this paper slight modifications are applied. Indeed, from the external point of view we now have \mathcal{B} as a functional analogue to the previous \mathcal{M} , that is the set of “parameter” data that entirely determines the state of the interaction between agent and environment. However, as the definition of \mathcal{B} refers to some explicitly *external* data (i.e. the τ in $\mathbf{b} = (\mathbf{m}, \tau)$), we cannot assume its knowledge from the point of view of the agent. We could however elect, on the same basis previous contributions used, to assume internal knowledge of the \mathbf{m} part of $\mathbf{b} = (\mathbf{m}, \tau)$. Note that this would *not* however guarantee us to achieve all previously obtained results since the class of represented agents and interactions grew far wider: this is precisely what motivated the reparameterization of the problem from \mathcal{M} to \mathcal{B} in the first place. Instead, we even dispose of the distinguished knowledge of “proprioception” *per se* altogether. By this we mean that although the agent *may* have access to feedback from its actuators, these values should be treated as part of a unified sensory array. We posit that the property for certain components to be *proprioceptive* can be inferred from otherwise discovered statistics of the interaction. This however effectively makes us lose a great deal

of knowledge since internal motor states \mathbf{m} used to represent the point where the agent stood in its relation to space.

As for remedies, it is instead where a variational approach is preferred: while configuration data represented by $\mathbf{b} \in \mathcal{B}$ still exists as an *external* object, the agent may only choose a *motor action* $a \in \mathcal{A}$ which, applied to \mathbf{b} , yields the following configuration $\mathbf{b}' = a\mathbf{b} = a(\mathbf{b})$. The agent is therefore given the capacity to compare any two elements of \mathcal{A} for equality, so that it may tell whether at any two steps of its sensorimotor experience it performed two identical or distinct actions. Moreover, much deeper change in knowledge occurs at the level of sensory readings: in the following we not only ask that the agent be able to compare its entire sensory output $\mathbf{s} = (s_c)_{c \in \mathcal{C}}$ for equality as a vector, but that it also can check for equality two values of any given sensel. That is, for every sensel $c \in \mathcal{C}$, for every values s_c, s'_c this sensel may output, the agent may test whether $s_c == s'_c$. In this contribution, it will be further assumed that the values output by distinct sensels are themselves *a priori* comparable for equality. While it is a common property in many classical applications, this constitutes a current limitation which will be addressed in a future ongoing work.

B. Sensorimotor binding: a marketplace for spatial information

The formalism introduced in Section II allows for space to appear as a variable in the sensorimotor equations, which we will use in this section to prove that under some reasonable assumptions we can talk about the spatial information content of a sensory signal. This in turn is used to form the basis of a *sensory prediction* the agent can use to try and infer the sensory consequences of its motor actions, mirroring the psychological construct of forward sensory model which is at the heart of ideomotor theories. This is the core idea we will further develop in the simulations of Section IV to see how a naive agent can derive such a prediction function from its sensorimotor flow.

Recall that for any given sensel $c \in \mathcal{C}$ and environment state $p \in \mathcal{P}$, we introduced F_c the receptive field of sensel c as the function which given agent configuration $\mathbf{b} \in \mathcal{B}$ yields $\mathcal{E}' = F_c(\mathbf{b}) \subset \mathcal{E}$ the minimal region of space which entirely determines the output of s_c . Therefore we can write

$$\forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, s_c(\mathbf{b}, p) = f_c(p|_{F_c(\mathbf{b})}), \quad (6)$$

where f_c is a “sensitivity” function (or filter) which converts the physical properties of environment sampled into a sensory output, both selecting to which property the sensor reacts and how. Equation (6) describes the sensorimotor dynamics by dissociating the spatial dependency (which is given by F_c) and the sensitivity one (as seen with f_c), so that we the observer can now speak of sensels that *look at the same region of space*. Let us then consider a particular condition, in which sensels c_i and c_j verify for motor action a the relation

$$\forall \mathbf{b} \in \mathcal{B}, F_{c_i}(a\mathbf{b}) = F_{c_j}(\mathbf{b}). \quad (7)$$

This describes the case where, for a given action a , sensel c_i samples *after* a the same point sensel c_j was sampling before

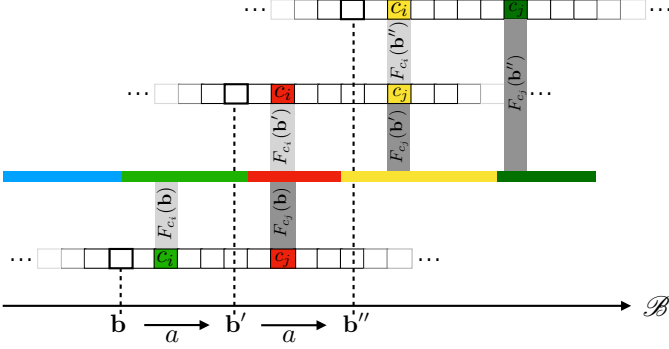


Figure 3: Illustration of how the underlying 1D space induces transitions between cross-sensel outputs. In this case, under action a , sensel c_i takes the place of sensel c_j : the output of c_j after a (red) is the same as the output of c_j before a (red). The same applies when performing a a second time: the yellow color is transferred from c_j to c_i .

the agent began to move. To have this relation apparent to the agent, we also require that the output of these particular sensels be comparable, as already discussed in §III-A. In the strictest sense, this can be by requiring that their sensitivity functions f_{c_i} and f_{c_j} are equal. It follows that

$$\begin{aligned} \forall \mathbf{b} \in \mathcal{B}, \quad \forall p \in \mathcal{P}, \\ s_{c_i}(\mathbf{ab}, p) = f_{c_i}(p|_{F_{c_i}(\mathbf{ab})}) \\ = f_{c_j}(p|_{F_{c_j}(\mathbf{b})}) = s_{c_j}(\mathbf{b}, p). \end{aligned} \quad (8)$$

Equations (7) and (8) are both illustrated in Figure 3, where a 1D (infinite) pixel array is placed in front of a 1D colored line along which the sensor can translate itself thanks to actions a . Eq. (7) is captured by the fact that both receptive fields $F_{c_j}(\mathbf{b})$ and $F_{c_i}(\mathbf{b}')$, drawn as two rectangular shapes, project on the same area on the environment. Then, Eq. (8) explains how it causes both sensels to actually generate the same sensory (red) output. It is clear that the spatial relation being forwarded to sensory transitions depends on the sensels actually outputting the same (red) value. This might be argued to be a restrictive assumption. Nevertheless, being able to deal with different sensitivity functions is a sizeable development to which an ongoing contribution shall be devoted.

To conclude, a key point here is that a phenomenon entirely defined from the external point of view through receptive fields is accessible from the internal one by the constraints it imposes on the sensels outputs values during exploration. Equation (8) therefore shows how space, insofar as it is common to all sensels and actions, makes this informational content accessible to a naive agent.

C. A motor and sensory account of spatial conservation

1) *Conservation through permutation : conservative actions*: The result obtained in the previous subsection exhibits an important property making internally available some external information. But given that the actual motor exploration follows the algebraic structure of actions \mathcal{A} , it still remains to be shown how these two structures are consistent. This can be made apparent by introducing *conservative actions* as those a

of \mathcal{A} verifying

$$\forall c \in \mathcal{C}, \exists c' \in \mathcal{C} \text{ such that } \forall \mathbf{b} \in \mathcal{B}, F_c(\mathbf{ab}) = F_{c'}(\mathbf{b}), \quad (9)$$

generalizing somewhat Equation (7). Then, conservative actions are those for which *all* sensels of the agent exchange the places they sample: there is conservation of the (spatial) information available. In the following, “ $\forall \mathbf{b} \in \mathcal{B}, F_c(\mathbf{ab}) = F_{c'}(\mathbf{b})$ ” will be shortened to the more legible $c \xrightarrow{a} c'$, and c (resp. c') is said to be the *predecessor* (resp. *successor*) of c' (resp. c) by a . It is proven in Appendix V-A that for conservative actions a , the relation \xrightarrow{a} can be made into a successor function

$$\begin{aligned} \sigma_a: \mathcal{C} &\rightarrow \mathcal{C} \\ c &\mapsto c' \end{aligned} \quad (10)$$

where $c' = \sigma_a(c)$ is a sensel verifying $c \xrightarrow{a} c'$. Therefore, conservative actions can equivalently be thought of as *permutation of sensels*. Importantly, conservative actions provide a natural framework for exploiting Equations (7) and (8) during motor exploration. Indeed, it is proven in Appendix V-B that conservative actions form a *subgroup* $\mathcal{A}_{\mathcal{C}} \subset \mathcal{A}$ for its composition operation. That is to say, chaining conservative actions yield other actions which are necessarily conservative, and the inverses of conservative actions are themselves conservative. This also defines a new group action of conservative actions on sensels via $ac = \sigma_a(c) = c'$ where $c \xrightarrow{a} c'$.

At this stage, it has been shown how the spatial property of permutation of the receptive fields relates to the intrinsic motor structure of the agent. However, this does not suffice to make this structure accessible to its knowledge as the dependency of the sensorimotor process on the spatial variable is implicit. We must therefore go through one final step to relate the available informational content (i.e. sensory reading) to the motor structure.

2) *From permutation to prediction : making it into sensory territory !*: Let us consider the agent at any point (\mathbf{b}, \mathbf{s}) of its sensorimotor experience. Its sensory output is $\mathbf{s} = s_{\mathcal{C}}(\mathbf{b}, p) = (s_c)_{c \in \mathcal{C}}$, and for any action a it may perform this sensory output should shift to $\mathbf{s}' = \mathbf{s}(\mathbf{b}' = \mathbf{ab}, p)$ provided the environment state stays constant throughout the action. But following our previous discussions, were a to be conservative we would have

$$\begin{aligned} \mathbf{s}' &= (s_c(\mathbf{ab}, p))_{c \in \mathcal{C}} = (f_c(p|_{F_c(\mathbf{ab})}))_{c \in \mathcal{C}}, \\ &= (f_c(p|_{F_{ac}(\mathbf{b})}))_{c \in \mathcal{C}} = (s_{ac}(\mathbf{b}, p))_{c \in \mathcal{C}}, \\ &= (s_{\sigma_a(c)})_{c \in \mathcal{C}}, \end{aligned} \quad (11)$$

so that performing motor action a only results in a permutation of the components of the sensory output. This permutation is exactly σ_a , and therefore is a constant of the agent which does not depend on the actual current configuration (\mathbf{b}, p) . Equation (11) shows that any conservative action $a \in \mathcal{A}_{\mathcal{C}}$ corresponds to a *sensory function*

$$\begin{aligned} P_a: \mathcal{S} &\rightarrow \mathcal{S} \\ (s_c)_{c \in \mathcal{C}} &\mapsto (s_{ac})_{c \in \mathcal{C}} \end{aligned} \quad (12)$$

which verifies the property

$$\forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, s_{\mathcal{C}}(\mathbf{ab}, p) = P_a(s_{\mathcal{C}}(\mathbf{b}, p)). \quad (13)$$

Per this property, P_a is a function which given *any* starting sensory reading of the agent can determine the sensation it would experience after performing action a (provided the environment state stays constant during a). It must be reiterated that a crucial part is that this function operates on *sensory* data, which is precisely the only data available to the agent.

D. A conservative policy to regulate the predictive market?

From there, let us now consider

$$P: \mathcal{A}_{\mathcal{E}} \rightarrow \text{Bij}(\mathcal{S}) \quad (14)$$

$$a \mapsto P_a$$

with $\text{Bij}(\mathcal{S})$ the set of all bijections from \mathcal{S} onto itself, i.e. P maps abstract motor actions to their sensory prediction functions. As proven in Appendix V-C, it establishes a group isomorphism between conservative actions $a \in \mathcal{A}_{\mathcal{E}}$ and their associated sensory prediction maps $P_a \in P(\mathcal{A}_{\mathcal{E}})$, so that

$$\mathcal{A}_{\mathcal{E}} \cong P(\mathcal{A}_{\mathcal{E}}). \quad (15)$$

While Equation (15) written as is might easily pass as benign, it is actually a very powerful result and the centerpiece of our argument. It indeed specifies how structure computable by the naive agent—as captured by its group of sensory predictions $P(\mathcal{A}_{\mathcal{E}})$ —relates to the *a priori* structure of its motor capabilities present in \mathcal{A} . Using the terminology introduced in §III-A, it shows how some *external* information describing the interaction between agent and environment can emerge as a hierarchical construct from the *internal* point of view. In turn, it is the enunciation—and the proof—of this result that motivate developing the formalism as in Section II, going as far back as absolute configurations $\mathbf{b} \in \mathcal{B}$ and ambient space \mathcal{E} . Equation (15) will be leveraged as part of the simulations in the following.

IV. SIMULATING A 2D VERSION OF OUR TOY MODEL

Up until this point, the discussion has been kept to a purely theoretical level. The following section is now devoted to a simulated experiment illustrating the new proposed formalism. To this end it starts with a description of the experimental setup, highlighting how it manifests in the proposed formalism of §III. Then, we describe what tasks the agent is given and how they relate to the theoretical results we put forth in the previous section. Finally, we review the observable results of these tasks to inspect how our earlier theoretical claims appear in practical cases.

A. Description of the experimental setup

In the following, we will consider the 2D generalization of the illustrating case used in the previous sections. That is, the studied agent body is now made of a planar, rectangular camera sat atop omnidirectional wheels, see Figure 4. These allow for translations along both x and y coordinates, as well as rotations in the plane. The pixels of the camera are sensitive to the luminance of the ambient stimulus, which for our experimental purposes is a fixed grayscale image placed above the moving camera. Describing the problem in the terms

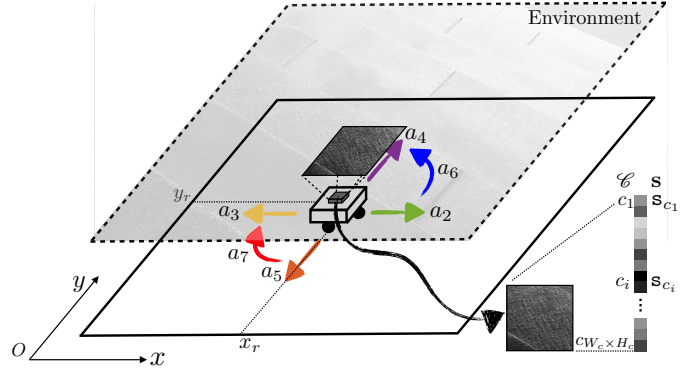


Figure 4: Experimental setup used in simulation to assess the proposed formalism. A holonomic agent is placed in a 2D environment which ceiling is made of a fixed grayscale image. The agent can move in this environment by applying 7 different actions a_k , including the “rest” one a_1 . A 10×10 camera pointed towards the ceiling is placed on the top of the agent and generates a sensory array $\mathbf{s} = (s_{c_i})_i$.

of the developed formalism gives:

- the ambient space \mathcal{E} is the plane \mathbb{R}^2 ;
- the set of physical properties of space \mathcal{P} is $[0; 255]$ the set of luminance values. Therefore, a state of the environment $p \in \mathcal{P}$ is a function which takes points (x, y) of the ambient plane and map them to luminances as given by the data of the acquired image;
- the configuration space \mathcal{B} is $\mathbb{R}^2 \times S_1 \cong \mathbb{R}^2 \times]-\pi; \pi]$ to account for both position (x, y) and orientation θ of the robot on the plane;
- the sensory output of the agent is an array $\mathbf{s} \in [0; 255]^{W_c \times H_c}$, with W_c (resp. H_c) the number of sensels/pixels in one row (resp. one column) of the camera. In the simulation, the image dimension is set to $W_c = H_c = 10$. Each of the components s_{c_i} of \mathbf{s} are the sensory output of pixel c_i , given by the luminance of the spatial location in the environment it is currently looking at. Importantly, the position of each pixel in \mathbf{s} is chosen arbitrarily.

Let us define a set A of *seven* basic actions a_k , $k = 1, \dots, 7$:

- 1) one identity action a_1 , mapping any current absolute configuration to itself;
- 2) four elementary translations a_2, a_3, a_4, a_5 , one for each direction of the basis axes on the plane. These are defined relative to the *current orientation of the agent*, which can end up distinct from external systems of axes when the agent rotates;
- 3) two elementary rotations a_6, a_7 , to account for both clockwise and counter-clockwise turns.

These actions are depicted in Figure 4 with colored arrows. Note that the color convention used in this figure is the same used in the forthcoming figures for coherence.

Relative to the prior discussion about properties of motor actions, these are not strictly *conservative* as per the definition (9): indeed, consider dF the elementary “forward” translation. While inner pixels of the camera will certainly exchange receptive fields, those in the front row will necessarily observe new areas of space after the agent has moved forward. Therefore none of these front row pixels has any successor

for dF , which precludes it from being strictly conservative. The same phenomenon of border unpredictability occurs for all translations, each with their respective side failing to verify the conservation property. We nevertheless proceed with the formalism on the basis that actions are at worst, informally speaking, “quasi” conservative. This is based on the quick analysis that, for a N -by- N square camera, this defect only occurs in N pixels which remains an order of magnitude fewer than the N^2 total.

Representing the sensory configuration as numerical arrays makes the permutation of sensels into N_c -by- N_c sparse matrices, where $N_c = W_c \times H_c$ is the number of sensels. Indeed, starting with any permutation $\phi : \llbracket 1, N \rrbracket \rightarrow \llbracket 1, N \rrbracket$ we can define a matrix $M_\phi \in M_{N,N}(\mathbb{R})$ by

$$M_{\phi,i,j} = \begin{cases} 1 & \text{iff } j = \phi(i), \\ 0 & \text{else.} \end{cases} \quad (16)$$

It can then be checked that for any array $\mathbf{s} = (s_i)$, the array $\mathbf{s}_\phi = (s_{\phi(i)})$ obtained by permutating the components of \mathbf{s} by ϕ verifies $\mathbf{s}_\phi = M_\phi \mathbf{s}$.

B. Description of the tasks

The proposed simulation can be decomposed as a sequence of 2 related, successive, tasks. First, these are briefly described in a global manner so as to go through the flow of the experiment. Then, each task is described in greater detail with respect to its implementation details. It is in this second part that relevant proofs ensuring both completion and correctness of the endeavor are provided. In this setup, the robot is given a set $\mathcal{A}_{\text{init}}$ of $n_{\mathcal{A}}$ unknown actions drawn in A . Although A was designed for convenience from an external point of view, $\mathcal{A}_{\text{init}}$ may not accurately reflect it. Indeed, for random draws there is a high likelihood of missing actions when $n_{\mathcal{A}}$ is small, of duplicate actions when it is large. However, as discussed previously these notions do not yet make sense to the agent, which can only “run” actions drawn. Importantly, in the following, the considerations will be restricted to the case where $\mathcal{A}_{\text{init}} = A$. This is a possibly strong assumption about the initial fitness of readily available commands to the “objective” capabilities of the agent. The influence of this choice and the effect of less optimally designed starting command shall be postponed to later works.

The first part of the experiment is one of *motor babbling*. During it, the agent effectively runs its available actions $a_k \in \mathcal{A}_{\text{init}}$ multiple times and tries to figure out whether they are *conservative* by computing their associated sensel permutation map. This is realized as a sequential process : at timestep n , the agent (randomly) chooses and runs an action $a_k = a[n] \in \mathcal{A}_{\text{init}}$, and the absolute configuration $\mathbf{b}[n] = (x[n], y[n], \theta[n])$ is accordingly changed to $\mathbf{b}[n+1] = a_k \mathbf{b}[n]$. Corresponding sensory array $\mathbf{s}[n+1] = (s_i[n+1])_i$ is then used to proceed in the computation of the (candidate) permutation matrix M_{a_k} of a_k , with the details of the update rule discussed in the following subsection.

Once this first task is complete, the agent computes all products of (quasi) permutation matrices to make the resulting set of matrices. As per Equation (4), this set is precisely the

one of all matrices that decompose over the $M_{a_k}, a_k \in \mathcal{A}_{\text{init}}$. This is formally analogous to the compressed representation of motor sequences observed in [17]. Following our argument about the groups of prediction functions and motor actions being isomorphic, this graph is taken as the *global* understanding of its motor capabilities the agent has acquired. Here “global” denotes that new structure, absent from the first empirical phase which was limited to $\mathcal{A}_{\text{init}}$, emerged from the computation of products.

1) *Learning the prediction through sensorimotor interaction*: The first task performed by the agent is computing, where possible, the permutation matrix associated to each of its available motor actions. This is done according to the following procedure: at the beginning of the sensorimotor experience, to each starting action $a_k \in \mathcal{A}_{\text{init}}$ associate a $N_c \times N_c$ matrix M_{a_k} where N_c is the number of sensels. This matrix is initialized so that all of its coefficients are 1. Then, at the end of timestep n where it performed action a_k (that is $a[n] = a_k$), the agent uses its sensory output arrays both previous ($\mathbf{s}[n]$) and current ($\mathbf{s}[n+1]$) as per the update rule :

$$(M_{a_k}[n+1])_{i,j} \leftarrow \begin{cases} 1 & \text{iff } s_j[n+1] = s_i[n] \text{ and } (M_{a_k}[n])_{i,j} = 1 \\ 0 & \text{else.} \end{cases} \quad (17)$$

Let us first observe that in this rule the only possible change in coefficients is going from 1 to 0 : whenever a coefficient $(M_{a_k}[n])_{i,j}$ is already 0, the condition of the first case automatically fails so that its value stays at 0. Therefore, the rough dynamics of the update is that while all coefficients start at 1, some are eventually switched to 0 upon exploration until matrices converge to a final (possibly null) form.

In Appendix V-D an argument is provided that for any conservative motor action this algorithm makes the empirical matrix M_{a_k} converge to the associated permutation matrix $M_{\sigma_{a_k}}$. It also proves that for non conservative actions, under the same richness hypothesis the associated empirical matrix will converge to the null matrix. This fact allows the robot to naively distinguish between conservative and non conservative actions, should he be given the capability to perform both on startup.

2) “*And then the simulation began to simulate...*” : *inferring motor structure from learned interaction*: In the second phase of the experiment, the agent uses the prediction functions it discovered for elementary conservative moves to infer how combinations of these moves relate to each other. Indeed, it was proved in the previous part that for any conservative actions a and a' with associated permutation matrices M_{σ_a} and $M_{\sigma_{a'}}$, it is true that

$$M_{\sigma_{a'}} M_{\sigma_a} = M_{\sigma_{a'a}} \quad (18)$$

so that the agent can internally infer the sensory prediction of *any* combination of the moves it empirically learned. This is used in a Dijkstra-like process to build a graph of prediction matrices, which runs as follows, see Algorithm 1: starting from a prediction matrix M_0 corresponding to any origin action a_0 , each of the known matrices $M_{a_k}, a_k \in \mathcal{A}_{\text{init}}$ are applied to yield both a set of new neighboring “end

Algorithm 1 Dijkstra-like algorithm for live construction of action group graph.

Input

- A The set of all matrices learned in exp.1
- D A bound on length of matrix combinations used
- O A reference matrix around which to explore

Output

- G A local view of the combinatorial graph of matrix products around O, using edges in A

Add O to collection U

O.depth \leftarrow 0

Add node O to G

while U is not empty **do**

 K \leftarrow node in U

for all M_α in A **do**

 P $\leftarrow M_\alpha K$

 P.depth \leftarrow K.depth + 1

if P.depth \leq D **then**

 B \leftarrow False

for all node C in G **do** \triangleright Test previously explored points for equality
 if predictions for P and C match **then**

 B \leftarrow True

 Set edge M_α : K \rightarrow P in G

end if

end for \triangleright for all node C in G

if B is False **then**

 Add P to U

 Add node P to G

 Set edge M_α : K \rightarrow P in G

end if

end if \triangleright if P.depth \leq D

end for \triangleright for all α in A

 Remove K from U

end while \triangleright while U is not empty

points” $N_{M_0} := \{M_{a_k} M_0, a_k \in \mathcal{A}_{\text{init}}\}$ and for each pair $M_0 - M_{a_k} M_0$ a directed edge M_{a_k} . This is then recursively applied to all newly discovered end points, while those that were previously visited (as the prediction matrices *can* be compared for equality) are discarded. However, the resulting graph would in most cases be infinite, therefore a stopping rule must be chosen. In our case, we chose to explore up to a given depth parameter in graph edge distance.

C. Results

This subsection is devoted to the evaluation in simulation of the previous points, divided in two successive experiments. The first one illustrates how the agent can build the permutation matrices associated to each of its conservative actions; a discussion about the convergence and the statistics of this experiment is then proposed. The second one exploits the permutation matrices just obtained to structure its own actions through a graph of their combinations; a discussion about its fidelity as a representation of the action group \mathcal{A} is proposed to conclude the subsection.

1) Experiment 1: discovering the permutations:

a) *Building the permutation matrices:* To begin with, the simulated robot in Figure 4 is placed at a random 2D position inside the image to be explored. The available action set is defined as $\mathcal{A}_{\text{init}} = A$ so that $n_{\mathcal{A}} = 7$. Then, at each time step n , a random action $a_k = a[n] \in \mathcal{A}_{\text{init}}$ is run, and the associated permutation matrix M_{a_k} is updated according to (17). After this update, the agent is able to evaluate if these matrices have finished converging and therefore can decide when to stop the exploration. An entropy-like internal criterion is proposed to quantify this convergence, along

$$C(M) = 1 - \frac{1}{N_c \log_2(N_c)} \sum_{i=1}^{N_c} H_i,$$

$$\text{where } H_i = - \sum_{j=1}^{N_c} \frac{M_{i,j}}{\mu_i} \log_2 \left(\frac{M_{i,j}}{\mu_i} \right), \quad (19)$$

$$\text{and } \mu_i = \frac{1}{\max(1, \sum_{j=1}^{N_c} M_{i,j})}.$$

In this criterion, H_i is the entropy of the post-action output of sensel c_i as a random variable of the pre-action outputs of all sensels c_j . Therefore it measures which degree of surprise remains in the determination of which (if any) sensel is successor to c_i . Finally, this makes C into an average measure of certainty in the discovery of successor sensel pairs, going in nondecreasing trajectories from 0 at initialization to 1 at permutation matrices. Consequently when it obtains the updated matrices $M_{a_k}[n+1]$, $a_k \in \mathcal{A}_{\text{init}}$ the agent computes all $C_k[n+1] = C(M_{a_k}[n+1])$ to assess the state of its discovery, stopping its exploration when all the C_k have reached 1.

After convergence, the resulting matrices for all the 7 actions shown in Figure 4 are depicted in Figure 5. In this figure, a 0 (resp. 1) is represented in black (resp. white), and action a_1 corresponds to the “rest” action. Since the agent has no knowledge of its sensor geometry, the position of its sensels (i.e. pixels) inside the sensory array \mathbf{s} (i.e. the flattened image) is randomly chosen. In this case, the resulting permutation matrices for each action is depicted in Figure 5 (top), demonstrating the fact that those matrices are not easy to understand from an external point of view. If one now selects a more natural ordering of the pixels inside \mathbf{s} , like a line by line arrangement, one then gets the permutation matrices in Figure 5 (middle). With such an arrangement, an external observer is now able to get a clearer intuition about the effects of each action on the pixels permutations. Nevertheless, these two different sets of matrices are purely equivalent from an internal point of view. This can be illustrated by mapping the permutation on the overall sensor to better catch how the agent has been able to discover the underlying spatial transfer between sensels. This is done by plotting the sensel pairs along which values are transferred as proposed in Figure 5 (bottom). In this figure, the 10×10 pixel grid of the simulated camera is represented together with arrows connecting each sensel to its successor. While such a representation requires external knowledge in the sensor geometry, the arrows are entirely determined by the internal permutation matrices from either of the two sets presented. It is thus a convenient external way

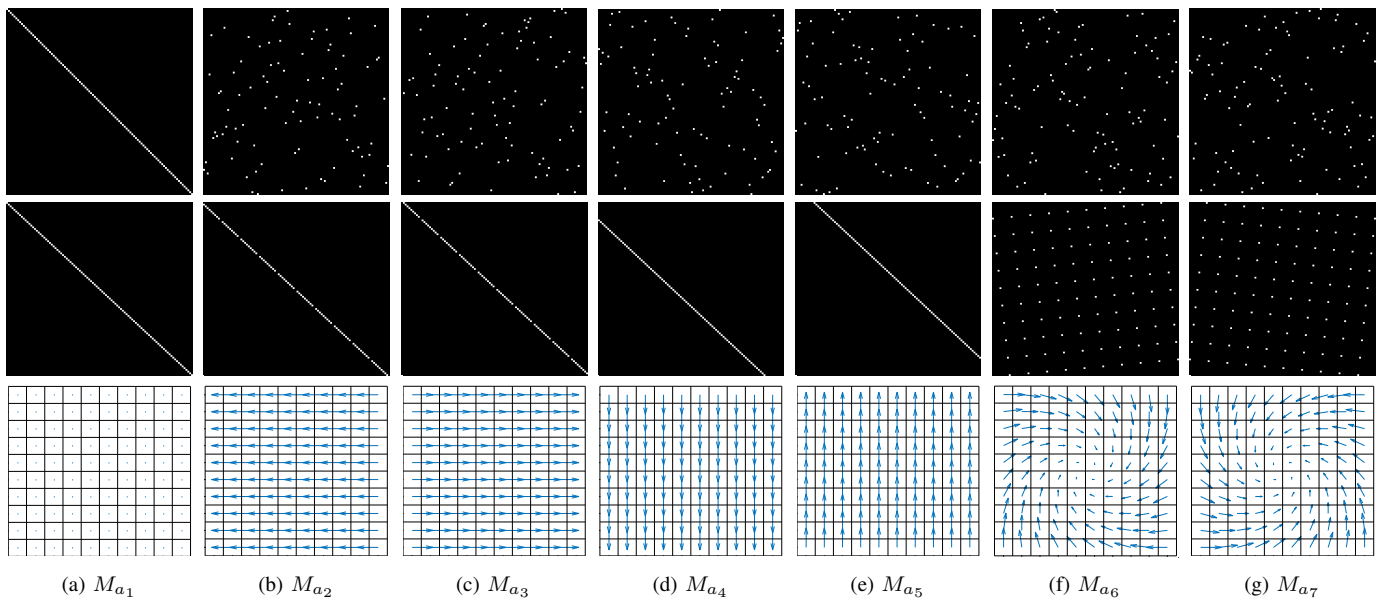


Figure 5: Representation of the 7 binary $10^2 \times 10^2$ permutation matrices M_{a_k} corresponding to the 7 actions a_k possibly generated by the agent, where a 0 (resp. a 1) is represented as black (resp. white). **(first row)** Matrices obtained for a random organization of the sensels outputs s_i inside the sensory array s . **(second row)** Matrices obtained for a well chosen sensels arrangement, where each pixel values are stored line by line in s . **(third row)** Interpretation of the permutation matrices (either from the first or second row) directly on the physical 10×10 pixel array: if a 1 is present at line i and column j of a matrix M_{a_k} , then an arrow joining pixel i to j is plotted. Note that the arrow length has been resized for actions 6 and 7 (i.e. rotations) to enhance readability.

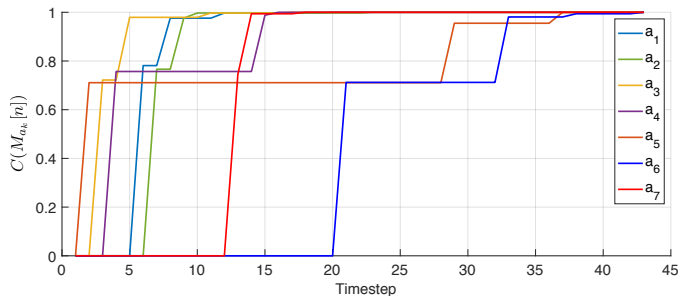


Figure 6: Representation of the criterion $C_k = C(M_{a_k})$ for the 7 actions a_k . Each jump in this figure corresponds to a reevaluation of the criterion happening at a timestep when the corresponding action has been drawn in the set in $\mathcal{A}_{\text{init}}$. As expected, the criterion starts from 0 to reach 1, indicating that all possible permutations have been found.

to display that each matrix has actually captured the pixel shift induced by each action. For instance, with such a visualization, it is now very clear that (a_2, a_3) , (a_4, a_5) or even (a_6, a_7) are all found to be pairs of inverse actions; this specific capability will actually be exploited in §IV-C2 to structure the agent set of actions.

b) A discussion about the dynamics of convergence: It is clear from Figure 5 that at some point the agent captured the permutation to the best of its capabilities. One therefore proposes to study the dynamics of the convergence of the approach w.r.t. the experimental time step n . First, the internal criterion $C_k = C(M_{a_k})$ defined in Equation (19) is evaluated at each n and each a_k , resulting in the plot in Figure 6. One can then confirm that the C_k increase from 0 (all elements in the matrices are initialized at 1) to 1 (all successor pairs have been discovered). It also appears that for each particular action a_k , the associated criterion increases in sparse jumps because its matrix M_{a_k} is only actually updated at the random time steps

when a_k is drawn. Figure 6 also illustrates the fact that the amplitude of these jumps decreases over the experiment. For the starting conditions of this experiment, a detailed analysis shows that about 7 realizations of each action are necessary to fully discover the target permutation matrices. But it also appears that most of the initial 1s in the matrices are wiped out very early, with a criterion value $C_k[n] \approx 0.7$ after only one execution of the corresponding action a_k . However, one still questions whether the differences in the dynamic of all actions is a random occurrence of this particular exploration, or there is an intrinsic variance in difficulty in learning between actions.

c) A statistical analysis about richness of the environment: The answer to the previous question can be obtained by performing an empirical survey (i) by averaging over random explorations for given starting conditions, and (ii) by varying these starting conditions and comparing the resulting performances. With such a study, (i) will allow to quantify the influence of the randomness in exploration, while (ii) lets us assess how the properties of the environment influence the discovery of permutations. For this experiment, the environment is made of the image shown in Figure 7a, where the starting points of each exploration is depicted as a grid of points on it. At each of these points, 1000 random explorations are conducted, each of them consisting in a random run of actions a_k as in §IV-C1a, resulting in 1000 sets of 7 C_k curves as in Figure 6. For each random exploration l and each action a_k , the number of jumps J_{l,a_k} in the C_k curve obtained is taken as a measure of difficulty in learning the permutation. The average $J = \frac{1}{L} \sum_l \sum_k J_{l,a_k}$ of J_{l,a_k} over all actions a_k and explorations l at a given starting position is depicted as the color of the grid in Figure 7a, with $L = 1000$ (runs) $\times 7$ (actions). Green points correspond to a low number of jumps

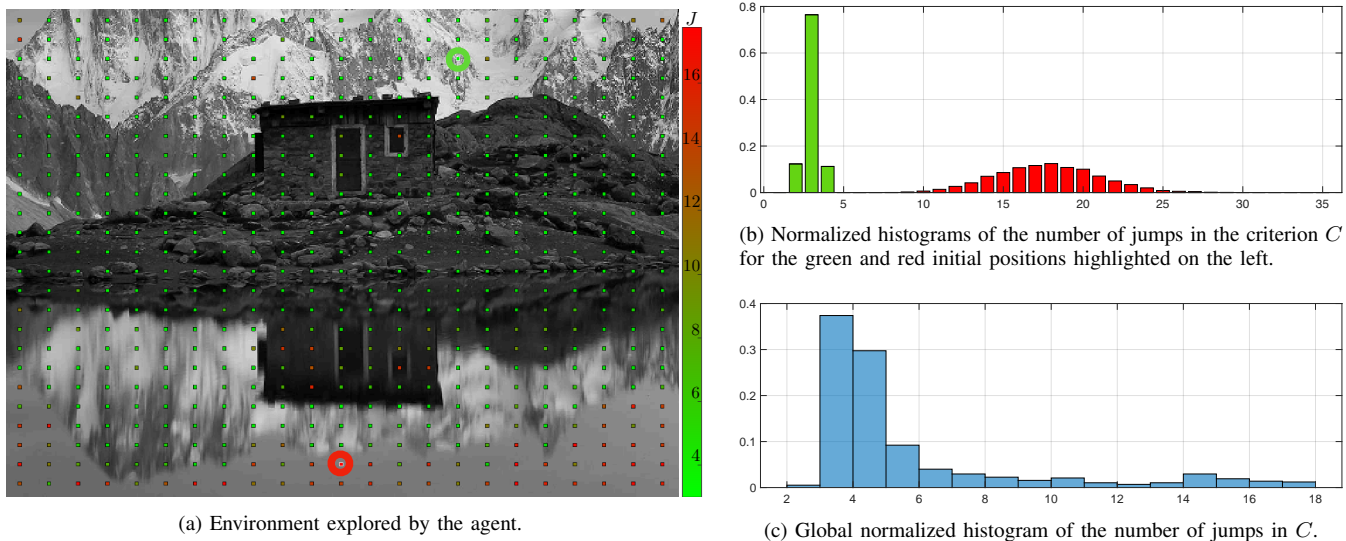


Figure 7: Statistical analysis of the permutation matrices building process. **(left)** Environment explored by the agent. Each point in this environment corresponds to a starting position around which the agent draws actions to build the permutation matrices M_{a_k} . Counting the mean number of jumps in the criterion curves C_k for each realization of the exploration around a given starting point and each action leads to the value J representing the difficulty to exploit for sensory prediction. A high (resp. low) J value in red (resp. in green) corresponds to areas in the environment harder (resp. easier) to exploit for sensory prediction. **(right)** Normalized histograms of the number of jumps in the criterion curves C_k averaged across actions. **(top right)** Focus on the histograms obtained around two different starting conditions corresponding to low (resp. high) J value highlighted by a green (resp. red) circle in the environment. **(bottom right)** Overall normalized histogram for all actions and all starting positions in the environment, showing that most of the permutation matrices are correctly obtained after a low number of action run.

J , while red ones are representing higher values. One can observe that the points are overwhelmingly green, and that the red ones are restricted to precise areas in the picture. These correspond to areas with locally low contrast, such as the sky (in the top left corner) or its reflection (in the bottom). The extremal conditions corresponding to the two green and red highlighted points are further compared. For each of them, the distribution of the J_{l,a_k} is plotted as an histogram in Figure 7b. Clearly, green points correspond to areas in the environment where the permutation matrices can be discovered in at most 5 executions of actions. On the contrary, at red points the agent must wait for about 17 on average, and up to 35, executions before it has obtained the same results. This illustrates how the richness of the environment might influence the agent ability to capture the structure of its sensory prediction. On a more global scale, Figure 7c shows the distribution of the J_{l,a_k} for all random explorations, indiscriminately of the starting position. This corroborates the observation that most positions in the image are green, i.e. lead to easy convergence. It appears that for a randomly selected starting position, there is more than 66% of chance of permutation matrices being discovered in less than 4 executions of their corresponding actions.

2) *Experiment 2: structuring actions by combination:* From the previous experiment, we now have as many permutation matrices M_{a_k} as we have actions in $\mathcal{A}_{\text{init}}$. As outlined in §IV-B2, one can then use them to build a *graph of prediction matrices* by following Algorithm 1. Recall that in this graph, a node is a permutation matrix obtained as a combination of the M_{a_k} matrices, while there is a M_{a_k} edge from matrix M to M' iff $M' = M_{a_k}M$. Therefore, all edges in the graph correspond to the permutation matrices built during experiment 1. According to Equation (15), this graph is isomorphic to

the graph of corresponding actions, meaning all properties discovered of any combination of matrices holds true for the corresponding combination of actions. As an example, if one discovers that $M_{a_1} = M_{a_2}M_{a_3}$, then one also has $a_1 = a_2a_3$.

As a first step, let us consider only the actions corresponding to translations in the environment, i.e. a_6 and a_7 are discarded from $\mathcal{A}_{\text{init}}$. This *a priori* selection is only made to simplify the visualization of the graph at first. After applying Algorithm 1 to the matrices shown in Figure 5, one gets the directed graph in Figure 8a, where all the color conventions are consistent with experiment 1. This particular graph has been built for a maximum depth set to 3 and with M_{a_1} the permutation matrix of the “rest” action taken as the origin of the graph. Note that the depth of this graph has been maintained voluntarily low so as to help in the reading of the graph. Note also that the arbitrary choice of origin makes all of its neighbors themselves correspond to one of the M_{a_k} discovered in experiment 1 since they all occurred as $M_{a_k}M_{a_1} = M_{a_k}$ products, whereas all other nodes are indeed new matrices.

This graph mirrors many algebraic properties of the M_{a_k} as captured by the internal experience. Indeed one can first observe that the light blue arrow leads from any given node M to itself, which corresponds to M_{a_1} being the identity matrix I_{N_c} . Furthermore one can note that the graph obtained is, up to its borders, completely homogeneous; that is the neighborhoods of each interior nodes share the same geometry. This even extends to the color of edges matching, so that some of them form pairs. One can for example verify that whenever a yellow edge goes from node M to M' , there is a green edge from M' to M and no other one. This identifies the corresponding matrices to be inverses w.r.t. matrix multiplication since from any starting node, taking first the green (resp. yellow) edge

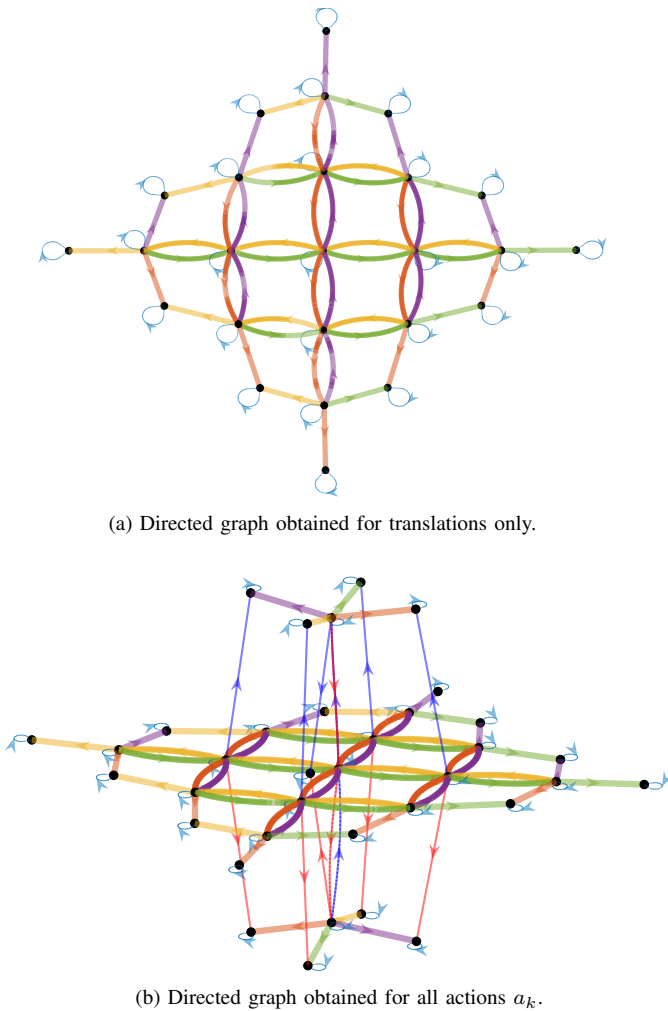


Figure 8: Directed graph of permutation matrices M_{a_k} —and thus also of corresponding actions a_k as per Equation (15)—obtained by combination of these matrices. Color conventions for edges match the color of each action in Figure 4.

then the yellow (resp. green) one forms a loop. The same can be said of the red and purple colors, which are found to correspond to another pair of inverse matrices. At last, the four central squares correspond to the commutativity of the M_{a_k} used: indeed one can see on the graph that taking the red edge first, then the green one always leads to the same node as green first, red second.

While those observations were discussed as properties of the permutation matrices, the actual result is their representing properties of the abstract motor actions a_k . And indeed one can check that the blue arrow corresponds to the identity action a_1 , that the inverse pairs (yellow, green) and (orange, purple) respectively correspond to (leftward, rightward) and (forward, backward) translations, and that the commutativity discussed is that of “forward then left” being the same as “left then forward”. While these facts seem obvious from an external point of view, they were not part of the initial knowledge of the agent discussed in §III-A. This only appears as a consequence of the agent capability to predict the sensory consequences of its own actions built during experiment 1.

This also applies to the graph shown in Figure 8b obtained

when considering all seven actions, i.e. the two rotations corresponding to actions a_6 and a_7 are now included in the analysis. This plot shows the same 2D graph of translations obtained before, but enriched with a third dimension supporting the change of orientation induced by rotations. Again, the depth of the graph is maintained low to keep things legible. The global structure of the graph can be described as a disjunction of 2D subgraphs corresponding to translations at fixed orientation. Each subgraph is therefore equivalent to each other up to a rotation as can be seen by the edges colors shifting between the plans. As an example, one can see that the same node in the graph can be reached by following either (yellow, dark blue) and (dark blue, purple), or (left, turn left) and (turn left, forward) in terms of actions seen from an external point of view. Figure 8b also shows that rotations are limited to the third vertical dimension in which they form cycles at constant position in the planar subgraph. The cycle simply mirrors the external observation that taking four $\pi/2$ rotations successively takes you back to the initial orientation. Importantly, this could constitute an internal signature of rotations as opposed to translations.

In the end of this second experiment, the agent has thus been able to discover a structure of its actual group of motor actions. The agent now has access to algebraic relations between its own actions which relate to its motor capabilities. This knowledge also allows it to generalize the sensory prediction it discovered in experiment 1 to all the combinations considered in the graph. Nevertheless, one have to keep in mind that all the actions considered in these experiments are not exactly conservative in the sense of Equation (9). Indeed, they fail at conserving spatial information on the border of the simulated camera. However, results show that conservativity holding true for all internal pixels allows for discovering the aforementioned properties. Because all calculations are made on actual current outputs of the pixels, one obvious consequence is that the agent has no way to predict what happens outside of its field of view, and so far it keeps no memory of it. Therefore, the results only hold for a very local movement w.r.t. the dimension of the agent. However, the discovered structure stays true whatever the initial position of the exploration. This is one crucial improvement in comparison to previous works, one brought about by the proposed variational formalism of actions.

V. CONCLUSION

This paper was devoted to the introduction of a new variational paradigm into the sensorimotor framework, extending the scope of such approaches to naive agents able to move freely in their environment. We demonstrated how these agents could exploit said approach not only to perform sensory prediction, but also to structure their own actions, with minimal prior knowledge. The proposed formalism has been assessed in simulation as a proof of concept, with a naive agent able (i) to build for each of its action some permutation matrices associated to its own sensory array, and (ii) to exploit them to structure its own set of actions. Thus, with the proposed contribution a naive agent endowed with

the ability to apply actions without any understanding of them can anticipate the sensory consequences of its movement in the environment, from which it can in turn discover the relations between actions and how to combine them. This is a major improvement w.r.t. existing sensorimotor approaches, leveraging the limitation of local understanding of the agent interaction with its environment.

Implementing a formal version of sensory prediction comes with many interesting perspectives, as it was shown to yield crucial properties both in the original cognitive psychology literature and in the previous robotic contexts. It should therefore be studied further how the results obtained here fare in reproducing these properties, including e.g. motor control, motor planning, isolating proprioception, differentiating self-induced changes from external ones or distinguishing objects.

Nevertheless, the applicability of the proposed paradigm to real agents or robots is still an opened question. First, it is clear that most of the actions an agent will be dealing with are not strictly conservative, but rather *quasi-conservative* like in the proposed simulations. While not extensively studied in this paper, some ongoing mathematical developments tend to show that their properties still allow to reach the same concepts of sensory prediction and action structuration. Then, the fact that the sensory prediction relies on exact sensory values shifts inside the un-noisy sensory array is not very realistic. Introducing stochastic matrices instead of permutation ones constitute a promising way to deal with such an issue, also pulling all these developments inside a probability territory [23] in which a lot of development still needs to be done. Finally, actions can also be noisy, and the question of their repeatability over time needs to be addressed so as to face realistic conditions. All these paths constitute future promising works in the field and will undoubtedly extend the scope of these approaches to naive adaptive and robust agents able to build by themselves their own understanding of their interaction with their environment.

APPENDIX

In the following proofs, we will assume

$$(\forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, s_{\mathcal{C}}(a'\mathbf{b}, p) = s_{\mathcal{C}}(a\mathbf{b}, p)) \Rightarrow a = a' \quad (20)$$

and

$$\forall c, c' \in \mathcal{C}, F_c = F_{c'} \Rightarrow c = c'. \quad (21)$$

These assumptions are of minimal importance for two reasons

- Eq. (21) only mandates that any two distinct sensels have different receptive fields *if only one time*, while Eq. (20) asks for actions to have no difference (as denoted by $=$) except that which can be assessed by the sensory capabilities $s_{\mathcal{C}}$. These conditions only fail to hold in very particular cases and can be found to be true in the presented examples.
- In any case where they indeed *fail* to hold, the exact same results can be found with suitable equivalence relations for actions (for Eq. (20)) and sensels (for Eq. (21)) at the cost of more loaded notations.

Therefore these conditions only serve as a way to streamline the presentation of the results with an at most negligible impact on generality.

A. Equivalency between conservative and permutative

Here is provided a proof that conservative actions can be described as permutations of sensels, as discussed in section III-C1.

Proposition 1. *Let a be a conservative action $\in \mathcal{A}$, there exists a unique map*

$$\begin{aligned} \sigma_a: \mathcal{C} &\rightarrow \mathcal{C} \\ c &\mapsto c' \end{aligned} \quad (22)$$

such that

$$\sigma_a(c) = c' \Leftrightarrow c \xrightarrow{a} c' \quad (23)$$

Proof. Let $a \in \mathcal{A}$ conservative and $c \in \mathcal{C}$. By conservativity $\exists c' \in \mathcal{C}$ such that $c \xrightarrow{a} c'$. Let $c'' \in \mathcal{C}$ such that $c \xrightarrow{a} c''$, then

$$\forall \mathbf{b} \in \mathcal{B}, F_c(\mathbf{b}) = F_{c'}(a\mathbf{b}) \text{ and } F_c(\mathbf{b}) = F_{c''}(a\mathbf{b})$$

so that

$$\forall \mathbf{b} \in \mathcal{B}, F_{c'}(a\mathbf{b}) = F_{c''}(a\mathbf{b}).$$

But $a: \mathcal{B} \rightarrow \mathcal{B}$ must be surjective because it is bijective, so that *all* $\mathbf{b} \in \mathcal{B}$ can be written $a\mathbf{b}'$ for some $\mathbf{b}' \in \mathcal{B}$. Therefore $F_{c'} = F_{c''}$, from which $c' = c''$: successor sensels are necessarily unique. We therefore declare σ_a to be the map that takes each sensel $c \in \mathcal{C}$ to its *unique* successor sensel. \square

Proposition 2. *For any conservative action $a \in \mathcal{A}$, its successor map σ_a is bijective.*

Proof. Let a be a conservative action, and let $c, c' \in \mathcal{C}$ be sensels such that $\sigma_a(c) = \sigma_a(c')$. From this it follows that

$$\forall \mathbf{b} \in \mathcal{B}, F_c(\mathbf{b}) = F_{c'}(a\mathbf{b}) \text{ and } F_{c'}(\mathbf{b}) = F_{c''}(a\mathbf{b})$$

for some common successor $c'' \in \mathcal{C}$. But it entails in particular

$$\forall \mathbf{b} \in \mathcal{B}, F_c(\mathbf{b}) = F_{c'}(\mathbf{b})$$

that is $F_c = F_{c'}$, which further yields $c = c'$: σ_a is injective. From injectivity of σ_a , it follows that $|\sigma_a(\mathcal{C})| = |\mathcal{C}|$. But because \mathcal{C} is finite it in turns follows from this equality that $\sigma_a(\mathcal{C}) = \mathcal{C}$, i.e. σ_a is also surjective. \square

B. Conversing of conserving

We provide here the proof, as used in section III-C1 and following, that conservative actions are themselves a subgroup of \mathcal{A} for its succession operation.

Proposition 3. *Let $\mathcal{A}_{\mathcal{C}} \subset \mathcal{A}$ be the subset of all conservative actions. Then $\mathcal{A}_{\mathcal{C}}$ is in fact a subgroup of \mathcal{A} .*

Proof. $\mathcal{A}_{\mathcal{C}} \subset \mathcal{A}$ by its very definition, therefore we only need prove it is actually a group.

- $\forall c \in \mathcal{C}, c \xrightarrow{e} c$ with e the identity action: e is conservative.
- Let a and a' be conservative actions, and $c \in \mathcal{C}$: since $a \in \mathcal{A}_{\mathcal{C}}, \exists c' \in \mathcal{C}$ such that $c \xrightarrow{a} c'$. But since $a' \in \mathcal{A}_{\mathcal{C}}$ too, there also exists $c'' \in \mathcal{C}$ verifying $c' \xrightarrow{a'} c''$, so that finally

$$\forall c \in \mathcal{C}, \forall a, a' \in \mathcal{A}_{\mathcal{C}}, \exists c'' \text{ such that } c \xrightarrow{a'a} c''$$

that is $a'a$ is conservative itself.

- Let $a \in \mathcal{A}_{\mathcal{C}}$ and let σ_a be its successor map $\mathcal{C} \rightarrow \mathcal{C}$. $\forall c \in \mathcal{C}$ since σ_a is surjective (see proof in V-A) we have $c = \sigma_a(c')$ for some $c' \in \mathcal{C}$, or equivalently

$$\forall c \in \mathcal{C}, \exists c' \text{ such that } c' \xrightarrow{a} c.$$

Finally, since $c' \xrightarrow{a} c \Leftrightarrow c \xrightarrow{a^{-1}} c'$ it follows that a^{-1} is conservative too. \square

C. Conservation or prediction, it is all the same

Here is provided a proof that mapping conservative actions a to their respective *sensory prediction functions* P_a provides a group isomorphism, as per Equation (15). To this end, let us recall the essential property of these functions:

$$\forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, s_{\mathcal{C}}(\mathbf{a}\mathbf{b}, p) = P_a(s_{\mathcal{C}}(\mathbf{b}, p)). \quad (24)$$

From this we get:

Proposition 4. *The map*

$$\begin{aligned} P: \mathcal{A}_{\mathcal{C}} &\rightarrow \text{Bij}(\mathcal{S}) \\ a &\mapsto P_a \end{aligned} \quad (25)$$

is a group morphism. Moreover it is injective, so that it induces a group isomorphism $\mathcal{A}_{\mathcal{C}} \cong P(\mathcal{A}_{\mathcal{C}})$.

Proof. Let a, a' be two conservative actions, we have

$$\begin{aligned} \forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, \\ P_{a'^{-1}a}(s_{\mathcal{C}}(\mathbf{b}, p)) &= s_{\mathcal{C}}(a'^{-1}\mathbf{a}\mathbf{b}, p) \\ &= P_{a'^{-1}}(s(\mathbf{a}\mathbf{b}, p)) \\ &= P_{a'^{-1}}(P_a(s(\mathbf{b}, p))) \\ &= (P_{a'^{-1}} \circ P_a)(s(\mathbf{b}, p)) \end{aligned}$$

so that $P(a'^{-1}a) = P(a'^{-1}) \circ P(a) : P$ is a group morphism.

Now let $a, a' \in \mathcal{A}_{\mathcal{C}}$ such that $P_a = P_{a'}$. It follows from Equation (24) that

$$\forall \mathbf{b} \in \mathcal{B}, \forall p \in \mathcal{P}, s_{\mathcal{C}}(a'\mathbf{b}, p) = s_{\mathcal{C}}(\mathbf{a}\mathbf{b}, p)$$

so that under hypothesis (20) P is indeed injective. \square

D. “Poof!” goes the matrix: a proof

This part is devoted to the proof of the relevance of Equation (17) in §IV-B1, that is the convergence of matrices M_a towards the associated permutation matrices M_{σ_a} for all conservative actions a .

Lemma 1. *For any coefficient $m_{a_i,j}$ of M_{σ_a} , the associated sequence $(m_{a_i,j}[n])_n$ of values taken in $(M_a[n])_n$ during exploration is nonincreasing with values in $\{0, 1\}$.*

Proof. Let us consider an arbitrary timestep $n \in \mathbb{N}$ in the exploration. If a is not drawn at this timestep, then

$$m_{a_i,j}[n+1] = m_{a_i,j}[n] \leq m_{a_i,j}[n].$$

If it is instead chosen, assuming $m_{a_i,j}[n] \in \{0, 1\}$ then as per the update rule of M_a , either

- $m_{a_i,j}[n] = 0$ and then $m_{a_i,j}[n+1] = 0$ too,

- or $m_{a_i,j}[n] = 1$ and $m_{a_i,j}[n+1] \in \{0, 1\}$

so that the lemma follows by induction on n .

Lemma 2. *For any coefficient $m_{a_i,j} = 1$ in M_{σ_a} , the associated sequence $(m_{a_i,j}[n])_n$ is constant with value $m_{a_i,j}[0] = 1$.*

Proof. At any timestep n of the exploration, if a is not chosen then $m_{a_i,j}[n+1] = m_{a_i,j}[n]$.

If it is instead drawn, then by Equation (11) we know that

$$s_j[n+1] = s_i[n]$$

because $m_{a_i,j} = 1$ implies that $j = \sigma_a(i)$ as per the definition of M_{σ_a} . Then by the update rule of $M_a[n]$, $m_{a_i,j}[n+1] = 1$. The lemma then follows by induction on n .

We now proceed with the last part of our argument, that is showing that coefficients of the empirical matrices M_{a_k} which do not correspond to successor sensel pairs will actually be nulled during exploration. This specific part is provided in the specific case of the simulated experiment presented, allowing us to formulate the relevant equations in the vector geometry of $\mathcal{E} = \mathbb{R}^2$. The same idea could also be adapted for generalized spaces \mathcal{E} , \mathcal{B} and actions $\mathcal{A}_{\text{init}}$, more in line with the previous theoretical descriptions. However, such a development is out of the scope of this contribution.

Let us define

$$\forall \theta \in \mathbb{R}, R_{\theta} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

the matrix corresponding to the rotation in \mathbb{R}^2 of angle θ . As per the definitions provided for our particular example, we may assume the properties :

$$\forall c \in \mathcal{C}, \forall (x, y, \vec{\theta}) \in \mathcal{B}, F_c(x, y, \vec{\theta}) = \begin{pmatrix} x \\ y \end{pmatrix} + R_{\theta} F_c(0, 0, \vec{0})$$

and

$$\forall a \in \mathcal{A}_{\text{init}}, \exists \vec{u}_a = \begin{pmatrix} x_a \\ y_a \end{pmatrix} \in \mathbb{R}^2, \exists \theta_a \in \mathbb{R}$$

$$\text{such that } \forall \mathbf{b} = (x, y, \vec{\theta}) \in \mathcal{B}, \mathbf{a}\mathbf{b} = (x', y', \vec{\theta}')$$

$$\text{where } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + R_{\theta} \begin{pmatrix} x_a \\ y_a \end{pmatrix} \text{ and } \theta' = \theta + \theta_a.$$

Therefore we have

Lemma 3. *Let $a \in \mathcal{A}$, $c, c' \in \mathcal{C}$. There exists a unique vector $\vec{d}_{a,c,c'} \in \mathbb{R}^2$ such that*

$$\forall \mathbf{b} = (x, y, \vec{\theta}) \in \mathcal{B}, \overrightarrow{F_c(\mathbf{a}\mathbf{b})F_{c'}(\mathbf{b})} = R_{\theta} \vec{d}_{a,c,c'}.$$

Proof. Let $\mathbf{b} = (x, y, \vec{\theta}) \in \mathcal{B}$. We therefore have

$$1) F_{c'}(\mathbf{b}) = \begin{pmatrix} x \\ y \end{pmatrix} + R_{\theta} F_{c'}(0, 0, \vec{0}),$$

$$\begin{aligned} 2) F_c(\mathbf{a}\mathbf{b}) &= F_c \left(\begin{pmatrix} x \\ y \end{pmatrix} + R_{\theta} \begin{pmatrix} x_a \\ y_a \end{pmatrix}, \vec{\theta} + \theta_a \right) \\ &= \begin{pmatrix} x \\ y \end{pmatrix} + R_{\theta} \begin{pmatrix} x_a \\ y_a \end{pmatrix} + R_{\theta + \theta_a} F_c(0, 0, \vec{0}) \end{aligned}$$

so that

$$\overrightarrow{F_c(\mathbf{a}\mathbf{b})F_{c'}(\mathbf{b})} = R_{\theta} \left(F_{c'}(0) - \begin{pmatrix} x_a \\ y_a \end{pmatrix} - R_{\theta_a} F_c(0) \right).$$

which proves taking $\vec{d}_{a,c,c'} = \left(F_{c'}(0) - \begin{pmatrix} x_a \\ y_a \end{pmatrix} - R_{\theta_a} F_c(0) \right)$ satisfies the property. \square

It should be noted that $\vec{d}_{a,c,c'}$ captures some geometry of conservation: indeed, from the definition of $\vec{\rightarrow}$ it can easily be shown that

$$\forall a \in \mathcal{A}, \forall c, c' \in \mathcal{C}, \left(c \vec{\rightarrow} c' \Leftrightarrow \vec{d}_{a,c,c'} = 0 \right). \quad (26)$$

This greatly serves the conclusion of our argument with

Proposition 5. *Let $a \in \mathcal{A}$, $c_i, c_j \in \mathcal{C}$. Then for given environment configuration $p \in \mathcal{P}$ the two statements*

1) *There exists an absolute configuration $\mathbf{b} \in \mathcal{B}$ such that if $\mathbf{b}[n] = \mathbf{b}$ and $a[n] = a$ for some $n \in \mathbb{N}$, then $m_{a_i,j}[k] = 0 \forall k \geq n + 1$*

2) *p is not doubly periodic with periods $\vec{d}_{a,c,c'}$ and $R_{\frac{\pi}{2}} \vec{d}_{a,c,c'}$ are equivalent.*

Proof.

- Assume that p is both $\vec{d}_{a,c,c'}$ - and $R_{\frac{\pi}{2}} \vec{d}_{a,c,c'}$ -periodic. Let $n \in \mathbb{N}$ such that $a[n] = a$, let $\mathbf{b} = \mathbf{b}[n]$. We have $s_i[n] = p(F_c(\mathbf{b}[n]))$ and $s_j[n+1] = p(F_{c'}(\mathbf{b}[n+1])) = p(F_c(\mathbf{b}[n]) + R_{\theta} \vec{d}_{a,c,c'})$. But since $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, $R_{\theta} = \pm I_2$ or $R_{\theta} = \pm R_{\frac{\pi}{2}}$. Therefore by periodicity of p we have $s_j[n+1] = s_i[n]$, from which by induction on n we get $\forall n \in \mathbb{N}$, $m_{a_i,j}[n] = 1 : 1 \Rightarrow 2$.

- Without loss of generality, let us assume that p is not $\vec{d}_{a,c,c'}$ -periodic (if it is instead only not $R_{\frac{\pi}{2}} \vec{d}_{a,c,c'}$ -periodic, the same argument follows up to a rotation). Let $X_0 \in \mathbb{R}^2$ such that $p(X_0) \neq p(X_0 + \vec{d}_{a,c,c'})$, $\mathbf{b}_0 = (x, y, \vec{0}) \in \mathcal{B}$ such that $F_c(\mathbf{b}_0) = X_0$. By definition $F_{c'}(a\mathbf{b}_0) = X_0 + \vec{d}_{a,c,c'}$ so that if $\mathbf{b}[n] = \mathbf{b}_0$ for some $n \in \mathbb{N}$, $s_j[n+1] \neq s_i[n]$. From the update rule of M_a we then get $m_{a_i,j}[n+1] = 0$, which by Lemma 1 concludes the proof. \square

Finally, simultaneously applying this proof to *all* actions and pair of sensels of the agent has us deduce:

Corollary. *If $p : \mathbb{R}^2 \rightarrow \mathcal{P}$ is aperiodic, then there exists a sequence of drawings of actions $(a[n])_{n \in \mathbb{N}}$ such that*

$$\forall a_k \in \mathcal{A}_{\text{init}}, \lim_n M_{a_k}[n] = M_{\sigma_{a_k}}.$$

While the converse strictly speaking is not true, we can see from the preliminary lemma that problems in the algorithm arise from very particular periodicity properties which relate to the geometry of (receptive fields of) sensels. It therefore should be noted already how most experiments in live specimens made use of specifically engineered symmetric and *periodic* environments to try and impair the development of perception [24]. Future works could expand on the effects of such “pathological” environment configurations on the proposed algorithm.

REFERENCES

[1] C. Lee, M. Kim, Y. Kim, N. Hong, S. Ryu, and S. Kim, “Soft robot review,” *International Journal of Control, Automation and Systems*, vol. 15, 01 2017.

- [2] T. Nguyen, S. Sreedharan, and S. Kambhampati, “Robust planning with incomplete domain models,” *Artificial Intelligence*, vol. 245, pp. 134–161, 2017.
- [3] A. Noë, *Action in perception*. The MIT Press, 2004.
- [4] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, “Interactive perception: Leveraging action in perception and perception in action,” *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–19, 2017.
- [5] H. Min, C. Yi, R. Luo, J. Zhu, and S. Bi, “Affordance research in developmental robotics: A survey,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 237–255, 2016.
- [6] J. K. O’Regan and A. Noë, “A sensorimotor account of vision and visual consciousness,” *The Behavioral and Brain Sciences*, vol. 24, no. 5, pp. 939–973; discussion 973–1031, Oct. 2001.
- [7] A. Laflaquière, J. K. O’Regan, S. Argentiari, B. Gas, and A. Terekhov, “Learning agents spatial configuration from sensorimotor invariants,” *Robotics and Autonomous Systems*, vol. 71, pp. 49–59, September 2015.
- [8] V. Marcel, S. Argentiari, and B. Gas, “Building a sensorimotor representation of a naive agent’s tactile space,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 2, pp. 141–152, June 2017.
- [9] V. Marcel, S. Argentiari, and B. Gas, “Where do I move my sensors? emergence of a topological representation of sensors poses from the sensorimotor flow,” *IEEE Transactions on Cognitive and Developmental Systems*, p. In press, 2019.
- [10] H. von Helmholtz, J. Southall, W. Nagel, A. Gullstrand, and J. von Kries, *Helmholtz’s Treatise on Physiological Optics*, ser. Helmholtz’s Treatise on Physiological Optics. Optical Society of America, 1925, no. vol. 3. [Online]. Available: <https://books.google.fr/books?id=1EzPAAAAAAAJ>
- [11] R. W. Sperry, “Neural basis of the spontaneous optokinetic response produced by visual inversion,” *J Comp Physiol Psychol*, vol. 43, no. 6, pp. 482–489, Dec 1950.
- [12] E. von Holst and H. Mittelstaedt, “Das reafferenzprinzip,” *Naturwissenschaften*, vol. 37, no. 20, pp. 464–476, Jan 1950. [Online]. Available: <https://doi.org/10.1007/BF00622503>
- [13] A. Stock and C. Stock, “A short history of ideomotor action,” *Psychol Res*, vol. 68, no. 2-3, pp. 176–188, Apr 2004.
- [14] L. W. Barsalou, “Grounded cognition,” *Annual Review of Psychology*, vol. 59, no. 1, pp. 617–645, 2008.
- [15] R. Rao and D. Ballard, “Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects,” *Nature neuroscience*, vol. 2, pp. 79–87, 02 1999.
- [16] J. Schröder-Schetelig, P. Manoonpong, and F. Wörgötter, “Using efference copy and a forward internal model for adaptive biped walking,” *Autonomous Robots*, vol. 29, no. 3, pp. 357–366, Nov 2010. [Online]. Available: <https://doi.org/10.1007/s10514-010-9199-7>
- [17] M. G. Ortiz and A. Laflaquière, “Learning representations of spatial displacement through sensorimotor prediction,” in *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2018, pp. 7–12.
- [18] N. Le Hir, O. Sigaud, and A. Laflaquière, “Identification of invariant sensorimotor structures as a prerequisite for the discovery of objects,” *Frontiers in Robotics and AI*, vol. 5, p. 70, 2018.
- [19] M. Garcia Ortiz, “Sensorimotor prediction with neural networks on continuous spaces,” in *Lecture Notes in Computer Science*, 10 2017, pp. 76–83.
- [20] R. Saegusa, F. Nori, G. Sandini, G. Metta, and S. Sakka, “Sensory prediction for autonomous robots,” 01 2008, pp. 102–108.
- [21] A. Seth, “A predictive processing theory of sensorimotor contingencies: Explaining the puzzle of perceptual presence and its absence in synesthesia,” *Cognitive neuroscience*, vol. 5, 01 2014.
- [22] D. Philipona, J. K. O’Regan, and J.-P. Nadal, “Is there something out there?: Inferring space from sensorimotor dependencies,” *Neural Comput.*, vol. 15, no. 9, pp. 2029–2049, 2003.
- [23] R. P. Rao and D. H. Ballard, “Chapter 91 - probabilistic models of attention based on iconic representations and predictive coding,” in *Neurobiology of Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds. Burlington: Academic Press, 2005, pp. 553–561.
- [24] R. Held and A. Hein, “Movement-produced stimulation in the development of visually guided behavior,” *Journal of comparative and physiological psychology*, vol. 56, pp. 872–6, 11 1963.