Bootstrapping, uncertain semantics, and invariance

Andrea Censi Richard M. Murray

***Abstract*—In the problem of bootstrapping, an agent learns to use an unknown body, in an unknown world, starting from zero information about the models involved. This is a fascinating problem, which so far has not been given a proper formalization. In this paper, we give a rigorous definition of what it means for an agent to be able to use "uninterpreted" observations and commands: there are some disturbances, represented by group actions, that modify what we call "semantic maps". The range of disturbances tolerated by an agent indirectly encode the assumptions needed by the agent. We argue that the behavior of agent which claims optimality (in any sense) must actually be *invariant* to such disturbances, and we discuss several design principles which allow to obtain this invariance for observations nuisances.**

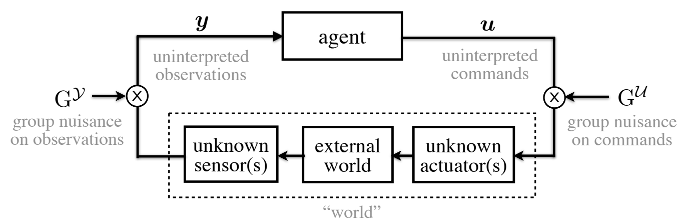
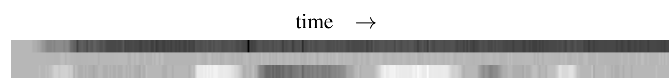


Figure 1. A bootstrapping agent interacts with the world (which is the series of unknown actuators, the external world, and unknown sensors) through two streams of "uninterpreted" observations and commands. In this paper, we argue that it is important to characterize exactly what assumptions the agent makes on these uninterpreted commands and observation, as the general case appears out of reach. We show that the assumptions can be encoded by *group nuisances* that act on the semantic maps of the agent, and that the closed-loop agent-world system must be *invariant* to such nuisances.



(a) Uninterpreted commands

(b) Uninterpreted observations (sensor #1)

(c) Uninterpreted observations (sensor #2)

Figure 2. *Raw observations and commands streams for a robotic platform.* A bootstrapping agent must learn to use uninterpreted streams of observations and commands. In these pictures, we show how such streams appear for an actual robotic platform. The robot is a planar, differential-drive robot, and the commands are taken to be the velocities. The two streams corresponds to 64 randomly sampled *sensels* of a camera and a range-finder. The data correspond to one minute of operation. Even with the advantage of knowing the two types of sensors, can the reader guess which is which?

I. Introduction

Eventually[1], robots are set to leave the structured environments of the industrial floors and share our lives, be them patient workers in our houses or embodied as intelligent cars. Still, there are many challenges ahead for such systems that must operate in unstructured environment, for long periods of time, with the necessary safety. The short-term problem, which is being addressed by projects such as ROS [2], is the software consolidation of the myriads of algorithms and processes that make up such a system. The long-term problem is that those software systems will be much more complicated than current projects, perhaps exceeding the design complexity which can be handled by a human: we remember Dijkstra's admonition that, ultimately, design complexity must be bounded by the size of a human skull [3]. It seems that the scarcest resource for designing robotic systems will not be computation, power, or an adequate sensory apparatus, but our design ability.

Therefore, it is likely that in the future all applications of learning will have an increasing role in robotics research. In particular, we speculate that developmental learning theories will be most important to design robots robust to unforeseen changes in their sensors, actuators and environment, and especially robots who are *aware* of such changes, according to Sutton's *verification principle* [4], which is an ability sorely missing at the current state of the art. However, it appears that the message of the community that gravitates towards conferences such as ICDL/EpiRob has not been received by a large part of the more applied roboticists that gravitate towards conferences such as ICRA, IROS, RSS (except for the few people dabbling in both). One possible reason is that some of the problems studied in developmental learning have evaded so far a precise mathematical formalization. This is entirely reasonable for a field which aims at imitating the complexity of the human brain, of which—it is safe to say—our colleagues in neurobiology will not have a clear, complete, functional description for many decades to come. But this lack of formalization prevents problems to be easily digested by the more applied roboticists.

In this paper, we venture towards a precise formalization of one aspect of bootstrapping problems: the idea that the agent starts its interaction with the world by "uninterpreted" observations and commands, whose semantics is unknown (Fig. 1). Even if this only interests the first layer of a complete bootstrapping architecture, it alone subsumes entire classes of problems studied in robotics, including all problems of

A. Censi and R. Murray are with the Control & Dynamical Systems department, California Institute of Technology. {andrea,murray}@cds.caltech.edu.

[1]Guessing dates is a futile attempt [1, Chapter 35].

intrinsic and extrinsic calibration. Is it possible to make the concept of "uninterpreted sensors and commands" mathematically precise? We give a formalization precise enough for the problem to be described by concrete mathematical tools, which in this case happens to be group theory. We give precise and falsifiable conditions for an optimal agent that claims to use uninterpreted observations and commands. Those conditions also allow to discuss general principles for the design of the first stages of an agent's sensory processing.

More in detail, Section II discusses the general problem of bootstrapping, and the impossibility of solving the general case, thus motivating the search for a way to clearly express the assumptions needed by an agent. Section III introduces the idea of "semantic maps", which map observations and commands to the proper events in the agent's internal model of the world. The assumptions needed by the agent can be encoded by the largest group of transformations of these semantic maps that can be robustly tolerated. These transformations are naturally represented by *group actions* that act as *nuisances* applied the representation of commands and observations. Therefore, one way to look at bootstrapping is as a problem of nuisance rejection. However, these nuisances are not time-variant "noise": they are fixed (but unknown) transformations that preserve the informative content of the signals. In Section IV we discuss that the information-preserving nature of the nuisances implies that the evolution of the closed-loop world-agent system must be exactly *invariant* to the nuisances, if there is any claim of optimality. In particular, the agent's behavior must be *invariant* to the observations nuisances, and *contra-variant* to the commands nuisances. The last sections discuss three general design principles to achieve invariance with respect to observations nuisances: group averaging (Section V), task invariance (Section VI) and observations canonization using pontifical features (Section VII).

Throughout the paper, we keep the discussion quite general, but we also give examples of group nuisances that are relevant to the case of robotic sensors, and that we have encountered in our previous research in bootstrapping models of robotic sensorimotor cascades [5], [6], to which we refer the reader interested in more complete examples of bootstrapping agents, as in this paper we focus only on the agents assumptions about observations and commands.

II. Bootstrapping needs assumptions

We are interested in the interaction of a bootstrapping agent with an unknown world. The interaction happens through a stream of observations signals $\boldsymbol{y}(t) \in \mathcal{Y}$ and commands $\boldsymbol{u}(t) \in \mathcal{U}$. The agent does not have previous knowledge neither of the external world nor of its own sensors and actuators. In fact, with "world" we refer to the unknown series of actuators, external world, and sensors. Observations and commands are "uninterpreted", in the sense that there is no semantics associated to them. The task for such an agent can be specified either by an external reward signal, or it can be described intrinsically [7]–[9] (for example, understanding the model of the world is a task in itself that does not need an external reward signal).

The many uncertainties involved in the problem make classical techniques not applicable. Control theory techniques for adaptive systems (e.g., [10]) assume relatively small classes of models conveniently parametrized. In POMDPs [11], the uncertainty is only in the observations and in the evolution of a model that is otherwise known. Deep belief networks [12], [13] are designed as universal approximators of data distributions, but they do not deal with actions. Reinforcement learning [14] gives a mathematical framework to deal with actions in unknown state spaces, but it does not explain how to build stateful representations from raw sensory values, which seems to be the dominant problem.

In Fig. 2, we see an example of the uninterpreted data that a bootstrapping agent should be able to use. Sensels (from *sens*ory *el*ement*s*) values are shown as intensity levels versus time. These data comes from a robotic platform. Can you even guess which one is the range-finder and which one is the camera? And how would you use such a sensor? This is the problem that you would need to solve if you were a brain-in-a-jar that for the first time gets connected to an unknown body in an unknown world. In fact, this is the problem that your brain *did* solve: the cortex of the human brain is an extremely flexible computational structure that can learn to process different streams of information; for example, the cortex in blind subjects rewires itself to process auditory signals [15], [16] (lower levels of perception do exhibit adaptiveness, but not so dramatic). Nature gives us a proof of existence that such kind of problem is solvable, and that, assuming a reductionist point of view, the computation required appears to be relatively simple.

Still, we have to be careful when trying to formalize the problem. At this level of generality, it also includes problems which would be impossible to solve. In fact, the dynamical system representing the world could realize a Turing machine, and it is not difficult to think of a task which would be equivalent to solving the halting problem. The other risk of generalization is coming up with general theories that are too abstract to give any insight for the design of actual agents. For example, Hutter [17] considers the case of reward-based agents, with no further assumptions about the world. He gives a complete characterization of what would be the optimal agent in such a setting; the results, otherwise very interesting, are not applicable to real-world agents—for example, the complexity of the generic optimal algorithm is just too high for the rich stream of data of robotic sensors. Also, we are interested in agents interacting with the real world; the general diagram in Fig. 1 includes the case of a web spider exploring the web, whose commands are URLs to retrieve, and whose observations are the retrieved pages. It is unlikely that this case will give any insight for real-world agents.

All this discussion points to the fact that it is either impossible, or not really useful, to design agents that work for *any* dynamical model for doing *any* task—in fact, the brain evolved for a large but finite sets of tasks. Therefore, stating precisely what are the *assumptions* on the world needed by a bootstrapping agent appears as a fundamental step towards a more formalized theory.

In this paper, we argue that it is possible to define rigorously part of the assumptions of an agents about the world, in particular with respect to the representation of observations and commands, using the language of group theory. The aim is to make a step towards a formalization of the early stages of bootstrapping, where the agents needs to make sense of "uninterpreted" data streams (we do not claim this point of view is also useful for higher-level cognitive tasks).

III. Semantics, assumptions, and group actions

This section describes a solid mathematical formalization for the vague idea of having "uninterpreted" observations and commands. Section III-A defines the concept of *"semantic" maps*, in the sense that they map the raw observations and commands to events in the internal model of the agent (whatever such model might be). In bootstrapping, such maps are not completely known. Section III-B argues that the uncertainty of the semantic maps can be adequately represented by *group actions* acting on observations and commands. The group actions that the agent can tolerate encode the assumptions that the agent has for the signals. Section III-C gives several examples of groups nuisances relevant to robotics.

A. Semantics is given by maps between σ-algebras

The difference between a traditional design problem and bootstrapping is that, in a traditional setting, one has a certain semantics attached to the commands and observations. Let us consider first the case of the observations. If we know the semantics, it means that we can associate a particular observation to an event relative to an internal model of the world. For example, suppose that $\boldsymbol{y} = \{y_i\}_{i=1}^n$ are the readings of a range-finder. Knowing the semantics of the data means being able to state propositions such as "If $y_3 = 4$, then there are no obstacles for 4m in the direction θ_3.". Here $E_1 =$"$y_3 = 4$" is an event relative to the observations, and $F =$"there are no obstacles in the direction θ_3" is an event relative to an internal model of the world. The semantics of the data allows to make many of such correspondences:

$$\begin{array}{lcl} E_1: \text{"}y_3 = 4\text{"} & \leftrightarrow & F_1: \text{"no obstacles at 30deg for 4m."} \\ E_2: \text{"}y_3 = 5\text{"} & \leftrightarrow & F_2: \text{"no obstacles at 30deg for 5m."} \\ \vdots & \vdots & \vdots \end{array}$$

Formally, a collections of events, satisfying certain coherence properties, is called σ-algebra [18]. Here we are dealing with two σ-algebras: $\mathcal{E} = \{E_1, E_2, \dots\}$ is the σ-algebra of the observations, and $\mathcal{F} = \{F_1, F_2, \dots\}$ corresponds to the internal model. Thus, having fixed $\mathcal{E}$ and $\mathcal{F}$, "knowing the semantics of the data" can be defined precisely as knowing the map $\eta : \mathcal{E} \to \mathcal{F}$ between the two σ-algebras.

We can give a similar construction for the commands. There is a σ-algebra $\mathcal{G} = \{G_1, G_2, \dots\}$ whose events correspond to particular choices of the commands (e.g., $G_1 =$"$u_2 = 3$"), and these events can be mapped to the σ-algebra $\mathcal{F}$ of the model (e.g., $F_1 =$"angular velocity is 30deg/s") by a map $\gamma : \mathcal{G} \to \mathcal{F}$.

We are not specifying much about the internal model to which the algebra $\mathcal{F}$ refers: for talking about the issues we are interested in this paper, we just need to postulate the existence of the maps η, γ, which we call *semantic maps*.

B. Groups nuisances encode agents assumptions

The observations/commands are completely "uninterpreted" if the maps η, γ are completely unknown; but a bootstrapping agent that can work with no information at all about the semantics maps η, γ is only a theoretical possibility. In fact, suppose that the n range-finder observations are represented by a string of $16n$ bits, and, as part of the sensor protocol, these bits are scrambled and encrypted before being transmitted to the agent. Encryption is a 1-to-1 map that does not change the informative content of the observations: it only changes the map η that maps the particular bit string to the event in the internal model. Therefore, if one claims that an agent can work with no assumption whatsoever on η, γ, one is claiming that such agent is flexible enough to undo any encryption scheme.

Thus it is critical to formalize the *assumptions* that bootstrapping agents need; we show how this can be done with group theory. The basic insight is that, instead of describing what information the agent *needs* about η, γ, it is easier to describe what information the agent does *not* need, or, equivalently, what disturbances acting on η, γ the agent can tolerate; and those disturbances are naturally represented by group actions.

Continuing our example with the range-finder, suppose that there are n readings $\{y_i\}_{i=1}^n$. Usually, when designing robot behavior, one assumes that the sensor calibration is known. This means that the agent knows which direction θ_i each readings y_i belongs to; and therefore one knows the correspondence between events about the i-th measurement and the portion of the environment in direction θ_i. Suppose that we want to require that the agent does not need the assumption of having a calibrated sensor, in the sense that it can recover this information through an intrinsic calibration procedure [19]–[22]. This can be formalized by saying that the agent is indifferent to a relabeling of the sensor elements. A relabeling is represented by a permutation σ, such that the vector of observations $\{y_i\}_{i=1}^n$ are mapped to the new vector $\{y_{\sigma(i)}\}_{i=1}^n$, where $\sigma(i)$ is the new label for the i-th sensel.

The set $\mathsf{Perm}(n)$ of permutation of sequences of length n is the simplest example of group. A basic introduction to group theory can be found in Rothman [23], but for convenience we recall the few basic notions that we use. A group $(G, *)$ is a set G equipped with an associative operation $*$, such that, for any two elements g_1, g_2 in G, also $g_1 * g_2$ is in G. Moreover, there is an identity element e, such that, for all g, we have $g * e = e * g = g$; and there exists a g^{-1} such that $g^{-1} * g = e$. However, in general, the operation is not commutative: $g_1 * g_2 \neq g_2 * g_1$. One can verify that $\mathsf{Perm}(n)$ is a group, because two permutations can be composed to obtain a new permutation, and each permutation can be inverted. Groups are the simplest mathematical object that can be used to describe *invertible transformations* applied to the data that are not commutative. Technically, these transformations are called *group actions.* We say that a group G *acts* on a set Y, if there exists an operation "$\cdot$", such that for any g in G, and any y in Y, $g \cdot y$ (read "g applied to y") is still an element of Y, $e \cdot y = y$, and the coherence condition $(g_1 * g_2) \cdot y = g_1 \cdot (g_2 \cdot y)$ is verified.

Group actions can describe the information needed by the agent, by describing the transformations of the semantic maps η, γ to which the agent must be robust. If we wanted to give the most generic treatment, given that the "world" is a dynamical object, we would consider generic (causal) transformations of such dynamical system that preserve information (i.e., the σ-algebra of the events). However, we do not lose much of the important ideas if we limit the discussion to *instantaneous* transformations of observations and commands. Let $\mathrm{G}^{\mathcal{U}}$ be a group acting on the commands $\boldsymbol{u}(t) \in \mathcal{U}$, and $\mathrm{G}^{\mathcal{Y}}$ be a group acting on the observations $\boldsymbol{y}(t) \in \mathcal{Y}$. By acting on the spaces $\mathcal{U}, \mathcal{Y}$, the groups also act on the semantic maps η, γ. For example, if originally we had

$$\eta(\{y_3 = 4\}) = \text{"there are no obstacles in direction } \theta_3\text{"}, \quad (1)$$

then if $\sigma \in \mathrm{G}^{\mathcal{Y}} = \mathsf{Perm}(n)$, we have the new semantic map $\eta' = \sigma \cdot \eta$, such that

$$\eta'(\{y_3 = 4\}) = \text{"there are no obstacles in direction } \theta_{\sigma(3)}\text{"}. \quad (2)$$

Equivalently, the group $\mathrm{G}^{\mathcal{U}}$ acts on the map γ.

At this point, we have a way to characterize precisely what are the assumptions for an agent: an agent's assumptions are described by two groups $\mathrm{G}^{\mathcal{U}}$, $\mathrm{G}^{\mathcal{Y}}$ acting on the commands $\mathcal{U}$, and observations $\mathcal{Y}$, such that the agent's task is still attainable if its semantic maps are replaced by $(g^{\mathcal{Y}} \cdot \eta, g^{\mathcal{U}} \cdot \gamma)$, for arbitrary $g^{\mathcal{Y}} \in \mathrm{G}^{\mathcal{Y}}$, and $g^{\mathcal{U}} \in \mathrm{G}^{\mathcal{U}}$. The larger the groups, the more flexible the agent is. For an agent that cannot tolerate any disturbance, we let $(\mathrm{G}^{\mathcal{Y}}, \mathrm{G}^{\mathcal{U}}) = (\mathsf{Id}, \mathsf{Id})$, where Id is the identity group $\{e\}$.

In Section IV we shall give an argument that not only the task should be achievable, but also *invariant* to these perturbations; but before that, in the next section, we give some examples of group nuisances that are relevant for robotics.

C. Some nuisance groups relevant for robotics

Table I shows some representative examples of group nuisances relevant for robotics.

Sensel values permutations: We already discussed permutations: they exchange the values of two sensels. A permutation nuisance represents the fact that the agent makes no assumptions on the ordering of the sensels in the observation vector.

Linear transformations: Linear transformations are the simplest case of nuisances that somehow mix the different signals. Let $\mathsf{GL}(n)$ be the set of all invertible $n \times n$ matrices. This is a group under the operation of matrix multiplication. The action of $\mathsf{GL}(n)$ maps each observation into a linear combination of all other observations. A linear transformation nuisance can encode the fact that agent needs no assumptions on the coordinate frame used to represent the observation, as well as invariance to rotation, scaling, etc.

Uniform and non-uniform sensel warping: Let $\mathsf{Diff}(\mathbb{R})$ be the set of diffeomorphisms (smooth invertible functions) from $\mathbb{R}$ to itself. These form a group, because the composition of two diffeomorphisms is an associative operation that produces another diffeomorphism, and an inverse always exist. Suppose that $\boldsymbol{y} \in \mathbb{R}^n$. A map $f \in \mathsf{Diff}(\mathbb{R})$ acts on the observations by mapping $y_i \mapsto f(y_i)$; or, more generally, we can think that there is a different nuisance for each observations: $y_i \mapsto f_i(y_i)$. A diffeomorphism nuisance encodes the fact that agent does not make any assumption on the representation (scale, measurement units, etc.) of a single command. But note that diffeomorphisms preserve (or reverse) the order relations of values ($a < b \Leftrightarrow f(a) < f(b)$), so that an agent indifferent to sensel values diffeomorphisms might still have assumptions on concepts such as "less intense" and "more intense".

Sensel space diffeomorphism: Assume that the observations are a field $\{y_s\}$, where $s \in \mathcal{S}$ is a spatial index that spans the sensel space $\mathcal{S}$, which is assumed to be a differentiable manifold. For example, suppose the sensor is a camera, and $\mathcal{S}$ is the visual sphere. A diffeomorphism nuisance $\varphi \in \mathsf{Diff}(\mathcal{S})$ maps $y_s \mapsto y_{\varphi(s)}$. The meaning of a diffeomorphism nuisance is that the agent does not know the position of its sensels on the visual sphere; using the computer vision jargon, the sensor is not *intrinsically calibrated*.

General automorphisms: The largest group of nuisances that we might consider is the group of automorphisms $\mathsf{Aut}(\mathcal{Y})$: it includes all possible 1-to-1 maps from the set $\mathcal{Y}$ to itself. This includes arbitrary encryption of the bit representation of the data, and it corresponds to the agent having no assumption whatsoever for the representation of the observations.

IV. Invariance and contra-variance

In this section, we claim that not only the agent's behavior should be robust to the group nuisances, but the overall evolution of the system must be *invariant* as well, *if* the agent has any claim of optimality. Here we need a separate discussion for the observations and the commands. For the observations (Section IV-B) the agent's behavior must be *invariant* to the nuisance; for the commands (Section IV-C), the agent must be *contra-variant*, because those nuisances must be compensated.

Table I
Groups nuisances acting on the observations

nuisance	**group** $\mathrm{G}^{\mathcal{Y}}$	$\mathcal{Y}$	**action**
Sensels values permutation	$\sigma \in \mathsf{Perm}(n)$	$\mathcal{X}^n$	$y_i \mapsto y_{\sigma(i)}$
Linear transformation	$\mathbf{A} \in \mathsf{GL}(n)$	$\mathbb{R}^n$	$y_i \mapsto \sum_j A_i^j y_j$
Sensel space diffeomorphism	$f \in \mathsf{Diff}(\mathcal{S})$	$\mathsf{C}(\mathcal{S};\mathbb{R})$	$y_s \mapsto y_{\varphi(s)},\ s \in \mathcal{S}$
Uniform sensel warping	$f \in \mathsf{Diff}(\mathbb{R})$	$\mathbb{R}^n$	$y_i \mapsto f(y_i)$
Non-uniform sensel warping	$\{f_i\} \in (\mathsf{Diff}(\mathbb{R}))^n$	$\mathbb{R}^n$	$y_i \mapsto f_i(y_i)$
Generic mapping	$\varphi \in \mathsf{Aut}(\mathcal{Y})$	any	$\boldsymbol{y}(t) \mapsto f(\boldsymbol{y}(t))$

The third column shows what are the assumptions on the observations space $\mathcal{Y}$ for the nuisance to make sense. For permutations, $\mathcal{Y} = \mathcal{X}^n$ indicates that each sensel value y_i must belong to the same space $\mathcal{X}^n$; for the diffeomorphisms, $\mathcal{Y} = \mathsf{C}(\mathcal{S};\mathbb{R})$ indicates that the observations are a continuous function over some manifold $\mathcal{S}$; automorphisms make sense for any $\mathcal{Y}$; and for the others we assume $\mathcal{Y} = \mathbb{R}^n$.

A. The closed-loop system must be invariant to nuisances

At this point, we have clearly defined what is the meaning of the nuisances that appear in Fig. 1: instead of the observations $\boldsymbol{y}(t)$, the agent receives $g \cdot \boldsymbol{y}(t)$, with g a fixed, unknown element of $\mathrm{G}^{\mathcal{Y}}$; and, likewise, the commands $\boldsymbol{u}(t)$ chosen by the agent are corrupted by the action of the group $\mathrm{G}^{\mathcal{U}}$. The larger the groups $\mathrm{G}^{\mathcal{U}}$, $\mathrm{G}^{\mathcal{Y}}$, the closer we are to the situation of completely uninterpreted data.

The diagram of Fig. 1 might look familiar; note, however, that usually nuisances are taken to be "noise", that is, time-variable quantities that perturb the signals and lose information, thereby irremediably degrading the performance of the agent. In this case, instead, the nuisances are *fixed but unknown*, and they *preserve* information: because they are groups actions, it is always possible to find the group element that reverses their effect. The nuisances do not change, in principle, what is possible for the agent to do: therefore, an *optimal* agent must necessarily act as to make the closed-loop system *invariant* to the nuisances.

B. Invariance to observations nuisances

It turns out that there is an asymmetry for the cases of observations and commands nuisances. For the observations, it is clear that the actions of the agent must be invariant to the observations nuisances. Let $\boldsymbol{u}(t) = \Theta(\{\boldsymbol{y}(t), \boldsymbol{y}(t-1), \dots\})$ be the actions of the agents as a function Θ of the history of the observations $\{\boldsymbol{y}(t), \boldsymbol{y}(t-1), \dots\}$. Then, the condition that must be satisfied by an optimal agent that the actions are invariant to any $g \in \mathrm{G}^{\mathcal{Y}}$ is that

$$\Theta(\{g\cdot\boldsymbol{y}(t), g\cdot\boldsymbol{y}(t-1), \dots\}) = \Theta(\{\boldsymbol{y}(t), \boldsymbol{y}(t-1), \dots\}). \quad (3)$$

A simple example of this optimality principle is in the pattern recognition problem, where decision rules should be invariant to position, orientation, and scale of the pattern, and this necessary invariance guides the design of detectors (e.g., [24]); in that case, the nuisance group consists of Euclidean transformation of the plane plus scaling.

The case of observations nuisances seems to be the simplest of the two, because this invariance can be achieved using some preprocessing of the observations which is relatively independent of the rest of the agent. In the next sections we shall see three design principles to achieve this invariance.

C. Contra-variance to commands nuisances

Group nuisances acting on the commands seem to be harder to handle. Here, we assume that the commands $\boldsymbol{u}(t)$ chosen by the agent are corrupted by a group $\mathrm{G}^{\mathcal{U}}$, so that the world receives the commands $\boldsymbol{u}'(t) = g\cdot\boldsymbol{u}(t)$, where g is an unknown element of $\mathrm{G}^{\mathcal{U}}$. The agent never sees the corrupted signal $\boldsymbol{u}'(t)$ directly; it is only aware of the nuisance by its effect on the world, whose model is unknown. In this case the agent, rather than being invariant to the commands nuisances, must be able to compensate for them. If the optimal commands were $\boldsymbol{u}(t)$ in the case without nuisances, then the optimal commands in the case of a nuisance g must be $g^{-1} \cdot \boldsymbol{u}(t)$, where g^{-1} is the inverse of g with respect to the group operation. In that case, the nuisance would be compensated, because $g \cdot (g^{-1} \cdot \boldsymbol{u}(t)) = (g * g^{-1}) \cdot \boldsymbol{u}(t) = \boldsymbol{u}(t)$. Technically, this is called *contra-variance.* Unfortunately, no principled, generic way is known to achieve this contra-variance.

V. Invariance by group averaging

Having established that invariance of to the observations nuisances is a worthy goal, we now turn to the question of how to achieve it. In particular, the interesting question is whether there are some design principles that give some guidance in designing the agent.

One general method found in the literature is *group averaging*. Assume the nuisance group G is applied to the data $y_0 \in Y$, so that $y = g \cdot y_0$ is observed instead of y_0, with g being an arbitrary element of G. Assume that one must compute an action from y, and this computation must be invariant to G; however, one only has a rule f which is *not* invariant to G, in the sense that $f(y_0) \neq f(g \cdot y_0)$. Then, one can simply average over the group, to obtain a smoothed version of f that is invariant by construction. Assuming G compact [25], the averaged f can be written by integrating over the Haar measure for the group:

$$f_G(y) = \int_G f(g \cdot y) dG, \quad (4)$$

However, there is no guarantee that the result of group averaging achieves a better performance than f in the task; in the worst case, the smoothed function f_G gets smoothed too much and becomes 0, which is invariant but rather useless.

VI. Invariance by invariant task design

Another way to achieve invariance to the observations nuisances is to make sure that the task is specified in a way which is invariant to the nuisances. Suppose that the task is represented by a known objective function $J(\boldsymbol{y}, \boldsymbol{u})$, that the agent must minimize with respect to $\boldsymbol{u}$ (this only works with an "intrinsic" function of known structure, not with an external reward). The idea is that if the objective function is invariant to the group nuisances acting on the observations, then it does not matter how the minimization takes place. Technically, the property that must be verified is that

$$J(g \cdot \boldsymbol{y}, \boldsymbol{u}) = J(\boldsymbol{y}, \boldsymbol{u}), \qquad \text{for all } g \in \mathrm{G}^{\mathcal{Y}}. \quad (5)$$

An example in the literature is in "natural" actor-critic algorithms [26] where the objective is shown to be invariant to a reparametrization, which acts as the nuisance in that problem.

A. Example of a servoing task

In previous work [5], we studied bootstrapping problems where the intrinsic task is *servoing*: given a goal observations $\boldsymbol{y}^\star$, choose the commands $\boldsymbol{u}$ such that $\boldsymbol{y}(t) \to \boldsymbol{y}^\star$. This is an example of an intrinsic task that makes sense for multiple sensor modalities; in fact, we used it to show that the same bootstrapping agent can deal with multiple robotic sensors (camera, range-finder, and field sampler). One way to solve the problem is to pose it as minimizing the objective function

$$J(\boldsymbol{y}(t), \boldsymbol{u}(t)) = \|\boldsymbol{y}(t+1) - \boldsymbol{y}^\star\|_2^2, \quad (6)$$

where the dependence on $\boldsymbol{u}$ is implicit in the fact that we are considering the next observation $\boldsymbol{y}(t+1)$. However, whether this is an admissible choice depends on the groups nuisances acting on the observations.

1) Linear transformation: Consider the case of $\boldsymbol{y} = \{y_i\}_{i=1}^n \in \mathbb{R}^n$, and the nuisance group $\mathsf{GL}(n)$. The objective function (6) is *not* invariant to a linear scaling $y_i \mapsto \sum_j A_i^j y_j$, for $\mathbf{A} \in \mathsf{GL}(n)$. In this case, what happens is that the nuisance action changes the relative importance of each sensel. Therefore, an agent minimizing (6) cannot be optimal in any sense, because its behavior depends on the nuisance. One way to fix the situation is to consider a slightly modified error function:

$$J(\boldsymbol{y}(t), \boldsymbol{u}(t)) = (\boldsymbol{y}(t+1) - \boldsymbol{y}^\star)^T \left(\mathrm{cov}\{\boldsymbol{y}\}\right)^{-1} (\boldsymbol{y}(t+1) - \boldsymbol{y}^\star). \tag{7}$$

One can verify that this objective function is invariant to the action of $\mathsf{GL}(n)$ as a linear scaling of $\boldsymbol{y}$ is compensated by the opposite scaling of $(\mathrm{cov}\{\boldsymbol{y}\})^{-1}$.

2) Diffeomorphisms: We consider the analogous problem when the nuisance is a diffeomorphism. Here we assume that $\boldsymbol{y} = \{y_s\}_{s\in\mathcal{S}}$, with s being a continuous index that spans over some differentiable manifold $\mathcal{S}$, so that we can write the error as an integral over $\mathcal{S}$:

$$J(\boldsymbol{y}(t), \boldsymbol{u}(t)) = \tfrac{1}{2} \int_{s\in\mathcal{S}} (y_s(t+1) - y_s^\star)^2 \; d\mathcal{S}. \tag{8}$$

The diffeomorphism nuisance $\varphi \in \mathsf{Diff}(\mathcal{S})$ maps $y_s \mapsto y_{\varphi(s)}$. If y is an image, a diffeomorphism dilates and shrinks parts of it. Therefore, the objective function 8 is not invariant to diffeomorphisms. With a bit of algebra, one can show [6] that the slightly modified version

$$J(\boldsymbol{y}(t), \boldsymbol{u}(t)) = \tfrac{1}{2} \int_{s\in\mathcal{S}} (y_s(t+1) - y_s^\star)^2 \sqrt{\det(\mathrm{cov}\{\nabla_s y_s\})} \; d\mathcal{S} \tag{9}$$

is invariant to diffeomorphism; the trick is that the term $\det(\mathrm{cov}\{\nabla_s y_s\})$ compensates for shrinking or dilation. Note that for both (7) and (9) the principle is to find a statistics of the data that can be incorporated in the objective function to compensate the nuisance.

VII. Invariance by pontifical canonization

Soatto [27] introduced a principled way to achieve invariance to group nuisances which can be generalized to our case. The basic idea is to make use of "pontifical" features that allow to define "canonical" representation of the data, which are invariant to the nuisances (Section VII-A). We generalize the idea to use "weak" pontifical features that identify a canonical element up to a transformation (Section VII-B). Then we apply the theory to the case of bootstrapping (Section VII-C) and study canonization procedures for the nuisances previously introduced (Table II).

A. Pontifical features identify canonical representations

It is convenient to first discuss the theory with relation to static observations. Suppose that some process produces data $y \in Y$, which is then corrupted by a group nuisance G, so that we observe $\tilde{y} = g \cdot y$, with g being an arbitrary and unknown element of G. Thus, for each original y, we can observe any element of the set $G \cdot y \triangleq \{g \cdot y \mid g \in G\}$, which is called the *orbit* of y. Pontifical features allow to choose, for each orbit $G \cdot y$, a canonical representation, thus achieving invariance with respect to the group nuisance.

Definition 1. A map $\phi : Y \mapsto \mathbb{R}^{\geq 1}$, is a *strong pontifical feature* for the group G acting on Y if, fixed y, the equation

$$\phi(g \cdot y) = 0, \qquad g \in G, \tag{10}$$

has exactly one solution in G, which is denoted by $g_y^\phi \in G$.

Equation (10) can be interpreted as follows: $\phi(y) = 0$ is a certain property that the data need satisfy; the fact that $\phi(g \cdot y) = 0$ is always solvable for g means that one can always find an element of the nuisance group that transforms the data as to satisfy that property. We call such a feature "pontifical" because, given its knowledge, one can find a "canonical" representation for y. In fact, suppose that we observe the data $y_1 = g_1 \cdot y$, where $g_1 \in G$ is the unknown nuisance acting on the pristine data y. If we know a strong pontifical feature ϕ, we can solve (10) to find $g_{y_1}^\phi$ and then compute the representation $\hat{y} = g_{y_1}^\phi \cdot y_1$.

Definition 2. The *canonical representation* of y corresponding to the strong pontifical feature ϕ is the element $\hat{y} = g_y^\phi \cdot y$, which by construction satisfies $\phi(\hat{y}) = 0$.

The canonical representation $\hat{y}$ does not depend on the particular group nuisance g_1 that corrupted the data in the first place, and it is unique for each orbit[2].

Example 3. Suppose we have a signal $y \in \mathbb{R}^n$, with $y_i > 0$, representing intensity values (thus the positivity constraint) that are corrupted by an unknown gain $k > 0$, which maps $y \mapsto ky$. One can obtain a canonical representation using the mapping $y \mapsto y/\|y\|_2$. Here, the unknown gain can be represented by the group nuisance $G = (\mathbb{R}_0^+, \times)$, and the strong pontifical feature used is $\phi(y) = \|y\|_2 - 1$.

B. Weak pontifical features identify heresy subgroups

A strong pontifical feature can be a rare luxury. In practice, we often have available features which do not quite identify a unique canonical representation.

Definition 4. A map $\phi : Y \mapsto \mathbb{R}^{\geq 1}$ is a *weak pontifical feature* for the group G acting on Y if, once a data y is fixed, the constraint equation (10) has a non-empty set of solutions $\{h \cdot g_y^\phi | h \in H_y^\phi\} \subset G$, where $g_y^\phi \in G$ and H_y^ϕ is a subgroup G.

According to this definition, a strong pontifical feature is a particular case of a weak pontifical feature where H_y^ϕ is the identity group. If we only have a weak pontifical feature, we can choose multiple elements $\{g_i\} \subset G_y^\phi$ to compute a set of

[2] In fact, suppose that, instead of $y_1 = g_1 \cdot y$, we had observed $y_2 = g_2 \cdot y$ (same original data y, different group nuisance g_2). Let $\hat{y}_i = g_{y_i}^\phi \cdot y_i$ be the result of canonization. By construction, we know that $\phi(\hat{y}_i) = \phi(g_{y_i}^\phi \cdot g_i \cdot y) = 0$. By virtue of associativity of group action, this can be written as $\phi((g_{y_i}^\phi g_i) \cdot y) = 0$. Because ϕ is pontifical, given a fixed y, there is only one solution to the group element acting on it. Thus $g_{y_1}^\phi g_1 = g_{y_2}^\phi g_2$ and consequently $\hat{y}_1 = (g_{y_1}^\phi g_1) \cdot y = (g_{y_2}^\phi g_2) \cdot y = \hat{y}_2$.

Table II
PONTIFICAL-FEATURE BASED CANONIZATION OF NUISANCES

nuisance	**group** G**,** **action on** W	**pontifical feature constraint** $\phi(\mathsf{W})=0$	**canonization map**	**heresy subgroup** $H_{\mathsf{W}} \leq G$
Sensel values bias	$\boldsymbol{v} \in \mathbb{R}^n$ $y_i(t) \mapsto y_i(t) + v_i$	$\mathbb{E}\{y_i\} = 0$	$\boldsymbol{v}_{\mathsf{W}}^{\phi} = -\mathbb{E}\{y_i\}$	Id
Linear transformation	$\mathbf{A} \in \mathsf{GL}(n)$ $y_i(t) \mapsto A_j^i y_i(t)$	$\mathrm{cov}(\boldsymbol{y}) = I$	$\mathbf{A}_{\mathsf{W}}^{\phi} = \mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}$	$\mathsf{O}(n)$
Sensel space diffeomorphism	$\varphi \in \mathsf{Diff}(\mathcal{S})$ $y(s,t) \mapsto y(\varphi(s),t)$	$\mathrm{cov}(\nabla y) = \alpha \mathbf{I}$	For $\mathcal{S} = \mathbb{S}^1$: $\varphi_{\mathsf{W}}^{\phi}(\theta) = c \pm \alpha \int_0^{\theta} (\mathsf{var}\{\nabla y(\beta)\})^{-\frac{1}{2}} \, d\beta$	$\mathsf{Isom}(\mathcal{S})$
Non-uniform sensel warping	$f \in \mathsf{Diff}(\mathbb{R})$ $y_i(t) \mapsto f_i(y_i(t))$	$y_i \sim \mathrm{Uniform}(0,1)$	$f_{\mathsf{W}}^{\phi} : y_i(t) \mapsto \mathrm{percentile}(y_i(t), y^{\mathbb{T}})$	$(\pm 1, \times)$

transformed data $\{\hat{y}_i\} = \{g_i \cdot y\}$, all of them satisfying $\phi(\hat{y}_i) = 0$. There cannot be multiple "canonical" elements, thus all of them (possibly except one, which we do not know anyway) must be "heretic", hence we call H_y^{ϕ} the *heresy subgroup* of ϕ with respect to G.

A weak pontifical feature still helps in achieving invariance, by allowing a *partial* canonization that reduces the nuisance group G to the smallest subgroup H_y^{ϕ}. This suggests a modular architecture where pontifical features are used in succession to reduce the original nuisance group to incrementally smaller uncertainties, until complete invariance is achieved.

Example 5. To generalize the previous example, consider the case of a multiplicative scalar gain $k \in \mathbb{R}$ acting on a signal $y \in \mathbb{R}^n$ (we dropped the positivity constraint, allowing $y_i \lesseqgtr 0$). A reasonable normalization step consists in computing $y \mapsto y/\|y\|_2$, or $y \mapsto -y/\|y\|_2$. The two possible choices arise because the feature $\phi(y) = \|y\|_2 - 1$ is only a weak pontifical feature for the group $G = (\mathbb{R}, \times)$, and its corresponding heresy subgroup is $H_y^{\phi} = (\{-1, +1\}, \times)$. To achieve complete invariance, we can use a successive canonization stage, by using, for example, the feature $\phi'(y) = \mathrm{sign}(y_1) - 1$.

C. Examples of invariance via canonization

To apply this theory to the case of bootstrapping, we have to be careful that, even though we have been considering instantaneous nuisances, we aim at obtaining a canonization of the whole dynamical system W representing the world that produces those observations. Instead of canonizing the single observation, what we need to do is finding a canonization $\hat{\mathsf{W}}$ of W. Consequently, a pontifical feature $\phi(\mathsf{W})$ is, in general, a function of statistics of the complete time series of observations and commands.

1) Canonization of sensel values bias: The simplest case is compensating for a bias in the measurements. Here, the group $\mathrm{G}^{\mathcal{Y}}$ is simply the vector space $\mathbb{R}^n$ equipped with addition as the group operation. The pontifical feature that we can use is

$$\phi(\mathsf{W}) = \mathbb{E}\{\boldsymbol{y}\} = 0. \tag{11}$$

Note the feature is a function of the entire dynamical system W, and is written using a statistical operation, in this case, the expectation of the observations. By solving the equation $\phi(g \cdot \mathsf{W}) = 0$, we obtain that the canonical transformation is $g_{\mathsf{W}}^{\phi} = -\mathbb{E}\{\boldsymbol{y}\}$, and the canonical representation $\hat{\mathsf{W}}$ is W in series with an instantaneous filter which removes the mean. We note that, in this case, ϕ is a *strong* pontifical feature, therefore its heresy group is the identity group.

2) Canonization of sensel value linear transformation: Another simple example is the case of $\mathrm{G}^{\mathcal{Y}} = \mathsf{GL}(n)$ acting on the observations $\boldsymbol{y}(t) \in \mathbb{R}^n$. The nuisances maps $\boldsymbol{y} \mapsto \mathbf{A}\boldsymbol{y}$, where $\mathbf{A} \in \mathsf{GL}(n)$ is any invertible matrix. Consider the candidate feature

$$\phi(\mathsf{W}) = \mathrm{cov}\{\boldsymbol{y}\} - \mathbf{I}. \tag{12}$$

where cov indicates the variance-covariance matrix of the observations. One solution of the canonization transformation is given by $g_{\mathsf{W}}^{\phi} = (\mathrm{cov}\{\boldsymbol{y}\})^{-\frac{1}{2}}$, where the square root is taken in the sense of the operator square root [28]. One can verify that such transformation, called *whitening* in the signal processing literature [29], guarantees that the covariance matrix of the data be the the identity matrix.

In this case, the feature ϕ is only a *weak* pontifical feature, because we can find other solutions. For example, $g_{\mathsf{W}}^{\phi} = -\mathrm{cov}\{\boldsymbol{y}\}^{-\frac{1}{2}}$ is a different canonization transformation which still satisfies the feature constraint. In general, one finds that a successive mapping $\boldsymbol{y} \mapsto \mathbf{M}\boldsymbol{y}$ preserves the covariance matrix if and only if $\mathbf{M} \in \mathsf{O}(n)$, and therefore $\mathsf{O}(n)$ is the heresy group for the feature.

3) Canonization of sensel space diffeomorphism: As a slightly more advanced example, we consider the case of a diffeomorphism nuisance, as introduced in Section III-C. We can show that a weak pontifical feature is

$$\phi(\mathsf{W}) = \mathrm{cov}\{\nabla_s y(s)\} - \alpha \mathbf{I}, \qquad s \in \mathcal{S}, \tag{13}$$

where $\alpha > 0$ is a parameter to be determined. Suppose that we are talking about an image. We are imposing that the covariance of the image gradients is constant at each point of the sensel space $\mathcal{S}$. A diffeomorphism nuisance dilates or expands certain parts of the image; this pontifical feature ensures that the statistics are uniform on the visual sphere.

Let $\varphi \in \mathsf{Diff}(\mathcal{S})$ be a group element acting on the image. The constraint equation $\phi(\varphi \cdot \mathsf{W}) = 0$ can be written as

$$\mathbf{J}\mathrm{cov}\{\nabla_s y(s)\}\mathbf{J}^* = \mathbf{I}, \qquad s \in \mathcal{S}, \tag{14}$$

where $\mathbf{J} = \partial\varphi/\partial s$ is the Jacobian of the transformation; note that, even if φ is in general nonlinear, the gradients are transformed linearly by the Jacobian. Equation (14) needs to be integrated to obtain the canonization transformation[3]. Also in this case the feature is only a weak feature, and we can identify the group of isometries of $\mathcal{S}$ as the heresy group for the feature. In fact, another diffeomorphism preserves the feature if and only if the Jacobian is orthogonal everywhere, and this implies that the diffeomorphism is an isometry [30].

VIII. Conclusions

It is critical to establish what are the assumptions that a bootstrapping agent needs on the world, as it appears extremely hard (and perhaps not useful) to solve the problem in full generality. In classical control problems, one has assumptions about the family of models under consideration (e.g., linear systems); in bootstrapping problems there are in addition assumptions about the semantics of the data. We showed that those assumptions can be described by the *group nuisances* acting on the agent's *semantic maps* that the agent can tolerate. Moreover, for an agent that claims any kind of optimality, it is necessary that the world-agent loop evolution be exactly invariant to such nuisances. The agent itself must be *invariant* to the nuisances acting on the observations, and *contra-variant* to those acting on the commands. For the nuisances acting the observations, we described three design principles to achieve the required invariance (group averaging, task invariance, and pontifical features), and we gave several examples for the nuisances common in a robotic system.

This shows that it is possible to formalize problems that at first sight are quite vague (the observations and commands are "uninterpreted") so that they can be approached with the relevant mathematical tools. Future work involves understanding whether generic design principles can be established for the nuisances on the commands, and whether this formalisms of semantic group nuisances is useful for understanding properties of bootstrapping agents at a higher level than the instantaneous continuous/analog sensorimotor interaction considered here.

Acknowledgments. Thanks to Scott Livingston for the many insightful comments on a very short notice.

[3]We can give a closed form solution for the case $\mathcal{S} = \mathbb{S}^1$ (the unit circle). We need to find a diffeomorphism $\varphi \in \mathsf{Diff}(\mathbb{S}^1)$. In 1D, we can parametrize the circle by the angle θ, the jacobian $\mathbf{J}$ is simply $\partial\varphi/\partial\theta$, and equation (14) can be written as $\left(\frac{\partial\varphi}{\partial\theta}\right)^2 \mathsf{var}\{\nabla_\theta y(\theta)\} = \alpha$. Because φ is a diffeomorphism, we know $\partial\varphi/\partial\theta \neq 0$; therefore, it has the same sign everywhere. We can choose the positive sign and obtain $\partial\varphi/\partial\theta = \alpha\, c(\theta)$ with $c(\theta) = 1/\sqrt{\mathsf{var}\{\nabla_\theta y(\theta)\}}$ is a known statistic of the data. Moreover, we have the constraint that $\int_0^{2\pi} \partial\varphi/\partial\theta = 2\pi$, because the circle must be mapped onto itself, which allows to choose the constant α. One final solution is given by

$$\varphi(\theta) = \alpha \int_0^\theta \frac{1}{\sqrt{\mathsf{var}\{\nabla y(\beta)\}}}\, d\beta, \qquad \alpha = 2\pi \Big/ \int_0^{2\pi} \frac{1}{\sqrt{\mathsf{var}\{\nabla y(\theta)\}}}\, d\theta. \tag{15}$$

To this solution, we have to add the isometries, consisting in the reflection $\varphi'(\theta) = \varphi(-\theta) \pmod{2\pi}$ and the rotations $\varphi'(\theta) = \varphi(\theta) + c \pmod{2\pi}$.

References

[1] N. J. Nilsson, *The Quest for AI*. Cambridge University Press, 2010.
[2] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.
[3] E. W. Dijkstra, "The humble programmer," *Communications of the ACM*, vol. 15, no. 10, 1972. DOI.
[4] A. Stoytchev, "Some Basic Principles of Developmental Robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, 2009. DOI.
[5] A. Censi and R. M. Murray, "Bootstrapping bilinear models of robotic sensorimotor cascades," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011. To appear. (link).
[6] A. Censi and R. M. Murray, "A group-theoretic approach to formalizing bootstrapping problems," tech. rep., California Institute of Technology, 2011. (link).
[7] A. Baranes and P. Y. Oudeyer, "R-IAC: Robust intrinsically motivated exploration and active learning," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 3, 2009. DOI.
[8] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, 2010. DOI.
[9] C. M. Vigorito and A. G. Barto, "Intrinsically motivated hierarchical skill learning in structured environments," *IEEE Transactions on Autonomous Mental Development*, 2010.
[10] P. Ioannou, *Adaptive Control Tutorial (Advances in Design and Control)*. SIAM, 2006.
[11] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, 1998. DOI.
[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, 2006. DOI.
[13] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, 2009. DOI.
[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
[15] Cohen *et al.*, "Functional relevance of cross-modal plasticity in blind humans," *Nature*, vol. 389, no. 6647, 1997. DOI.
[16] O. Collignon, P. Voss, M. Lassonde, and F. Lepore, "Cross-modal plasticity for the spatial processing of sounds in visually deprived subjects.," *Experimental brain research*, vol. 192, no. 3, 2009. DOI.
[17] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Berlin: Springer, 2004. (link).
[18] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
[19] J. Stober, L. Fishgold, and B. Kuipers, "Sensor map discovery for developing robots," in *AAAI Fall Symposium on Manifold Learning and Its Applications*, 2009. (link).
[20] M. Boerlin, T. Delbruck, and K. Eng, "Getting to know your neighbors: unsupervised learning of topography from real-world, event-based input," *Neural computation*, vol. 21, no. 1, 2009. (link).
[21] E. Grossmann, J. A. Gaspar, and F. Orabona, "Discrete camera calibration from pixel streams," *Computer Vision and Image Understanding*, vol. 114, no. 2, 2010. DOI.
[22] J. Modayil, "Discovering sensor space: Constructing spatial embeddings that explain sensor correlations," in *Proceedings of the International Conference on Development and Learning (ICDL)*, 2010. DOI.
[23] J. Rothman, *An introduction to the theory of groups*. Springer-Verlag, 1995.
[24] Q. Le, J. Ngiam, Z. Chen, D. Chia, and P. Koh, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.
[25] G. S. Chirikjian and A. B. Kyatkin, *Engineering applications of noncommutative harmonic analysis*. CRC Press, 2001.
[26] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, 2009. DOI.
[27] S. Soatto, "On the distance between non-stationary time series," in *Modeling, Estimation and Control* (A. Chiuso, S. Pinzoni, and A. Ferrante, eds.), vol. 364 of *Lecture Notes in Control and Information Sciences*, Springer Berlin / Heidelberg, 2007. DOI.
[28] R. Bhatia, *Matrix Analysis*. Springer, 1997.
[29] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.
[30] M. do Carmo, *Riemannian Geometry*. Birkhauser, 1994.