Bootstrapping interactions with objects from raw sensorimotor data: a Novelty Search based approach

Carlos Maestre [*,†], Antoine Cully [*,†], Christophe Gonzales[§,‡] and Stephane Doncieux [*,†]
[*]UMR 7222, ISIR, Sorbonne Universites, UPMC Univ Paris 06, Paris, France
[†]UMR 7222, CNRS, ISIR, Paris, France
Email: {maestre, cully, doncieux}@isir.upmc.fr
[§]UMR 7606, LIP6, Sorbonne Universites, UPMC Univ Paris 06, Paris, France
[‡]UMR 7606, CNRS, LIP6, Paris, France
Email: christophe.gonzales@lip6.fr

Abstract—**Determining in advance all objects that a robot will interact with in an open environment is very challenging, if not impossible. It makes difficult the development of models that will allow to perceive and recognize objects, to interact with them and to predict how these objects will react to interactions with other objects or with the robot. Developmental robotics proposes to make robots learn by themselves such models through a dedicated exploration step. It raises a chicken-and-egg problem: the robot needs to learn about objects to discover how to interact with them and, to this end, it needs to interact with them. In this work, we propose Novelty-driven Evolutionary Babbling (NovEB), an approach enabling to bootstrap this process and to acquire knowledge about objects in the surrounding environment without requiring to include *a priori* knowledge about the environment, including objects, or about the means to interact with them. Our approach consists in using an evolutionary algorithm driven by a novelty criterion defined in the raw sensorimotor flow: behaviours, described by a trajectory of the robot end effector, are generated with the goal to maximize the novelty of raw perceptions. The approach is tested on a simulated PR2 robot and is compared to a random motor babbling.**

I. Introduction

A robot can act in a robust manner without any internal model [1]. Nevertheless, utilizing one of them has several advantages, like allowing the robot to predict the result of its actions. This capability may allow the robot to simulate an action before executing it, thus reducing the risks of damaging itself. It also allows the robot to use planning or learning algorithms that can explore how to reach a desired goal and, then, to execute only the required actions. The ability to predict what happens and how this changes over time is also a requirement of Intelligent Artificial Curiosity [2] that allows the robot to focus its learning efforts on what it can actually learn.

To predict the outcome of the robot's actions, a model needs to predict (1) what forces the corresponding motor commands will create, (2) what movements will result from them, taking into account the physics and potential interactions with other objects in the environment, (3) what impact they will have on the environment and (4) what consequences they will have on future robot perceptions. Parts (1) and (4) depend on robot morphology only: it defines a robot *self-model*. Parts (2) and (3) correspond to a *world model*.

While the robot's self-model can be designed or generated during its manufacturing phase, the world model should be adapted to the robot's environment and tasks. Designing the world model requires to anticipate every situation that a robot can encounter, which can be a very complex task, if not impossible, in an open environment. The Spirit mars rover relied on a team of engineers to handle any unforeseen situation [3], but it is not feasible to spend so much human effort to adapt the world model of each future autonomous robot. A promising approach consists then in allowing robots to learn on their own these models according to the situations they are dealing with.

Learning a world model requires a substantial amount of data, which should highlight physical properties of the robot's surrounding environment, and of the objects it contains. Whereas applying any motor command or any perception gathered can be useful to learn a self-model, generating useful data about the world is more tricky. Demonstrations can help, but for the system to be autonomous, it should be able to generate these data on its own, at least partly. Very few actions will generate useful interactions, like contacts with objects. Furthermore, relevant actions or perceptions are highly dependent on the robot's environment and, from an embodied intelligence perspective, it should be discovered automatically by the robot to exploit as much as possible features that a human designer may neglect [4]. But how to bootstrap a system that should learn world models with as less *a priori* knowledge about how to interact with it as possible?

In this paper, we present a method named Novelty-driven Evolutionary Babbling (NovEB), designed to perform a task-agnostic exploration of an unknown environment. Its main feature is to look for actions that maximize novelty in the raw sensorimotor space. It is based on Novelty Search [5] which relies on Evolutionary Algorithms driven by a behaviour novelty criterion. The outcome of this approach is a dataset of interactions with objects and their consequences (changes in the visual perception for example). These data are aimed at preparing a future developmental step for a robot, which would consist in training classifiers or predictors, like deep learning algorithms for instance[1]. The robot's self-model (in this paper, a forward/inverse kinematic model) is given to explore the environment directly in its workspace (a 3D Euclidean space) instead of exploring through the motor space. Nevertheless, no information about the environment are provided to the robot, making the proposed method strictly environment-agnostic.

The paper is organized as follows: Sections II and III present papers on world model learning as well as the basics on evolutionary-based Novelty Search. Then Section IV introduces our algorithm. Section V highlights its effectiveness on some experiments. Finally, a discussion is provided in Section VI and a conclusion in Section VII.

II. Related works

Developmental robotics proposes the generation of datasets used to create the world model through the exploration of the environment [6]. Some previous works ([7], [8], [9], [10]) use a random motor babbling to create this knowledge. Their key idea is to send random commands to robot joints in order to produce different movements and collect the resulting data.

In [7], a random number is thus selected as the value of a specific robot arm's joint. This value remains unaltered during a few iterations, while the values of the other arm's joints change. Once the robot has performed a given number of iterations, a new random value is set to another joint, and the process starts over again. When the total number of iterations is equal to 20000, the exploration stops, and a world model is created on the basis of the corresponding dataset, using an MDP [11]. The model generated from the random motor babbling in [9] is a Bayesian Belief Network [12].

Random motor babbling presents some limitations. On one hand, it can execute many movements that do not produce any contact with the objects composing the scene, repeatedly exploring regions not providing any data. On the other hand, when an interaction is finally generated, it has a small impact, if any, on the rest of the babbling process; meanwhile a local exploration could generate many new significant data.

Other mechanisms can be used to drive the exploration of the environment, e.g., the use of intrinsic motivations. In [2] Oudeyer et al. proposes Intelligent Artificial Curiosity to drive the exploration of a robot's actions and interaction abilities. A robot executes actions in its search space and it uses the corresponding data to train predictors, called experts, which progressively get specialized in different regions of the sensorimotor space. The next action to apply is then randomly chosen in the action space covered by the expert with the maximum learning progress. Although this approach is very promising, it requires to provide a structure to the world modeling experts: what are the dimensions to take into account? Predicting raw sensorimotor values seems out of reach when using vision, for instance, at least without a careful pretraining. The approach proposed in this paper is complementary to artificial curiosity: it may provide the data out of which relevant dimensions could be extracted to bootstrap a curiosity mechanism.

Evolutionary robotics relies on stochastic algorithms to generate robot controllers or morphology [13]. They have been used to generate controllers for locomotion, navigation or foraging tasks [14]. They have the specificity of not requiring the definition of a discrete set of actions and they can explore large sets of continuous variables, provided that they can make enough solution evaluations. It was shown recently that using task-independent behaviour-based criteria had a very significant impact on the generated results [15]. Novelty search, in particular, uses only the novelty of behaviours to drive the search [5]. It was shown in robotic applications that it actually led to better results than a task-based criterion [5]. In this paper, we propose to use this approach in a developmental robotics context.

III. Background

A. Evolutionary algorithms

Evolutionary computation relies on the variation and selection principles of natural selection in order to drive a search and optimization process [16]. They are population-based algorithms in which many candidate solutions, called individuals, are considered in parallel. Starting from a random population, an iterative process evaluates each individual, selects some of them according to their fitness value[2] and generates new solutions thanks to blind search operators (mutation, that makes small random modifications to an individual, or crossover that mixes several individuals). An individual contains a *genotype* and the corresponding fitness value. The genotype corresponds to the structure to be designed or optimized. Typical genotypes are vector of floating point values, strings of binary digits, trees or neural networks. See [16] for an introduction to this topic.

B. Novelty Search

Novelty Search is a method to search for novel behaviours [5]. The novelty of a behaviour is defined as the average behavioural distance between this behaviour and its k-nearest neighbours in the current population and in an archive of previously explored behaviours:

$$Novelty(i,p,a) = \frac{1}{k}\sum_{j=0}^{k} dist(i, neigh(i,p,a)_j) \qquad (1)$$

where $neigh(i,p,a)_j$ is the jth-nearest neighbour of individual i, including the current population p and the archive a, with respect to distance *dist*, representing the distance between the corresponding behaviours. $Novelty(i,p,a)$ is then used as a fitness function in the evolutionary process. The method strongly relies on the behavioural distance used to compute novelty. This distance is typically defined in a space of behaviour descriptors and is problem-specific. This work

[1]This part is out of the scope of this paper

[2]The fitness function describes what is expected from individuals, this is the function maximized by the evolutionary process

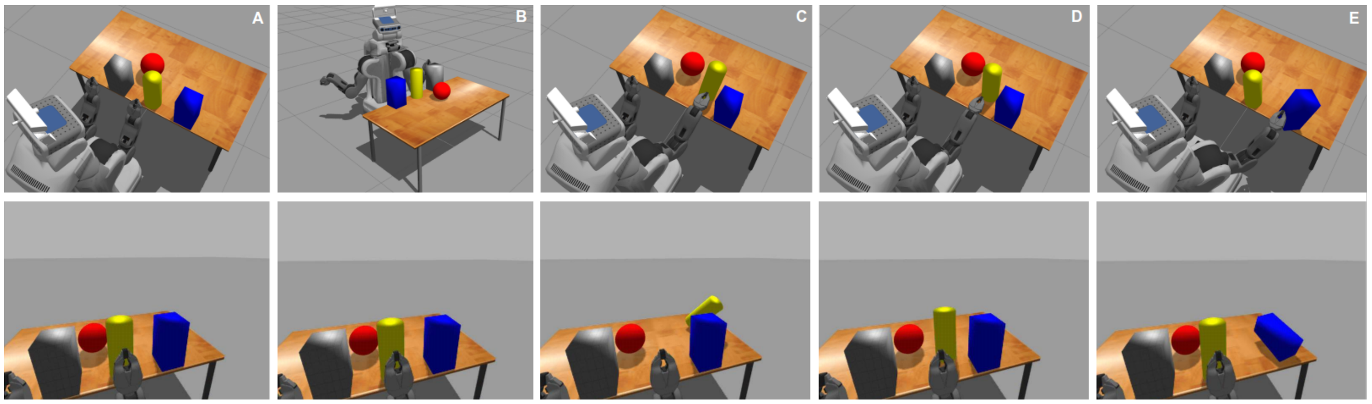

Fig. 1. Examples of different moments during the experiments using NovEB. Each column represents the execution of a trajectory (except for column A that displays the initial state before any trajectory is executed). The top row represents the moment just after the execution of a trajectory, i.e., when the robot's arm has completed the trajectory. The bottom row shows the final image obtained once the arm has come back to its initial position. (Column B) A trajectory without any object on the table being touched does not produce any change in the environment. (Column C) A trajectory in which an object is touched, hence producing a change in its position. (Column D) A trajectory in which the contact with the object is slightly different can result in a very different output. (Column E) A trajectory in which several objects are touched. An illustrating video is available on: https://youtu.be/zCO7qOIvKOU

draws inspiration from [17], where a hexapod robot quickly and autonomously learns to walk in any possible direction in its vicinity, using Novelty Search to modify the robot's controllers.

IV. Novelty-driven evolutionary babbling

At the beginning of the studied developmental step, the robot knows nothing about its surrounding environment. It needs to generate sensorimotor data to be used in a next developmental step to define the structure and relevant dimensions of its world models. The robot thus needs to explore possible interactions with surrounding objects so that its future world models can extract information about them and start predicting how they behave. To this end, we propose *Novelty-driven Evolutionary Babbling (NovEB)*. This method relies on Novelty Search [5] to explore possible robot's movements, while focusing on those that generate the highest novelty from the perception point of view. NovEB makes the assumption that new perceptions result from robot actions. It generates many different robot arm trajectories, $c^1, \ldots, c^S$, and it looks at the modifications they may create in the environment, as perceived from the robot's sensors (vision in particular). Due to Novelty Search principles, a movement that generates perceptions that have never been encountered before has a higher chance to survive, namely to be selected to generate new close movements through the mutation operator. Movements that do not generate any perceptual novelty are discarded, thus focusing the search on movements generating new perceptions.

Generate_children_pop(pop) and *Select(pop)* are based on NSGA-II [18], a "Pareto-based multi-objective evolutionary algorithm", which is a state-of-the art algorithm for multi-objective problems; but it is also very efficient in mono-objective ones. *Generate_children_pop(pop)* is the function that creates a new population with mutation and crossover. *Novelty(i, p, a)* is defined as in equation (1), on the basis of the nearest neighbours in the current population; and in an archive of past behaviours with a distance defined in a behavioural space described below. At each generation, the individual with the highest novelty is added to the archive. In mono-objective problems, *Select(pop)* is an elitist algorithm that selects the best individuals among the parent and the children populations, using the function *Fitness(i)*.

The main algorithm of NovEB is the following:

```
1:  $pop \leftarrow c^1, c^2, ..., c^S$                      ▷ random population
2:  $a \leftarrow \emptyset$                        ▷ $a$ stands for the novelty archive
3:  $g \leftarrow 0$                ▷ number of generations in the population
4:  while $g < g_{max}$ do
5:      for $i \in pop$ do
6:          Execute_trajectory (i)      ▷ compute behaviour of $i$
7:      end for
8:      $max_{novelty} \leftarrow 0$
9:      for $i \in pop$ do
10:         $Fitness(i) \leftarrow Novelty(i, p, a)$
11:         if $Novelty(i, p, a) > max_{novelty}$ then
12:             $max_{novelty} \leftarrow Novelty(i, p, a)$
13:             $best_{candidate} \leftarrow i$
14:         end if
15:     end for
16:     $archive \leftarrow archive \cup \{best_{candidate}\}$
17:     $pop \leftarrow pop \cup Generate\_children\_pop(pop)$
18:     $pop \leftarrow Select(pop)$
19:     $g \leftarrow g + 1$
20: end while
```

The genotype is a vector of waypoints in an Euclidean space, used to define a trajectory of the robot's end effector[3]. In the initial population, these values are randomly generated within a defined range. Afterwards, the values are modified by the mutation operator, without any restriction on the resulting

[3]This type of environment exploration is inspired by the goal-directed exploration [19]. We do not follow the same terminology to avoid any misunderstanding with the use of the term goal as a synonym of a task to solve.

values. Two mutation operators are defined. The first one adds a random Gaussian noise to a given trajectory. The second one can change the complexity of the trajectory by adding one waypoint. Added points are put in the middle of two other points of the trajectory. This ability to change the complexity of the genotype is inherited from NEAT [20], and is also a feature of Novelty Search: the search starts with simple solutions, considers the behaviours they can generate and progressively considers solutions of higher complexities [5].

The behaviour associated with an individual is an image of the scene as seen by the robot once its arm has come back to its initial position[4]. The robot's movements eventually change the scene by moving objects, being this reflected in the gathered images. A behavioural distance used to compute novelty among the images has to be defined.

V. Experiments

A. Evaluated scenario

The objective of this experiment is the generation of contacts of the robot's end effector with objects around the robot. The results of this babbling is a set of observations aimed at building world models (including models of objects). The scene defined for the experiment is composed of a table with objects on top of it: on the left a gray box, on the front-center a can, on the right a blue box, and on the back-center a ball. All the objects are within reach of the right arm of the robot, except for the ball, that can be moved only if it is pushed by another object. The two boxes and the can are located at the border of the table, and the ball is located behind them. The can is located just in front of the robot, in the middle of the table (see Fig. 1). The weight chosen for the objects on the table is low to facilitate their movement when the end effector of the robot makes contact with them. The experiment was run for around 20000 iterations, as in [7]. At the end of each iteration, the objects are put back to their original position.

A trajectory is initially composed of an initial position, common to all trajectories, and a final point to be reached. The randomly generated final coordinates are in the range [0,1], being able to go beyond these bounds afterwards. The robot relies on a motion planner, called OMPL[5], to plan a trajectory. It is not always possible to compute a trajectory given a final point, due to this is without reach of the robot, or the collision includes collisions with the table, the ground or the robot. Only the safe and feasible trajectories are executed.

The behaviour descriptor associated to a trajectory is the final image of the scene, which is taken once the arm came back to the initial position. To compute the distances required for the novelty objective, images are encoded into a numeric string by using the pHash library, due to "perceptual hashes are close to one another if the features [of the two images] are similar" [21]. The distance between two strings is then computed with the Hamming distance [22].

[4]This is meant to avoid to take into account robot's arm movements as a source of novelty. This could be useful to generate a self-model, but it would be misleading for building a world model.

[5]http://ompl.kavrakilab.org/

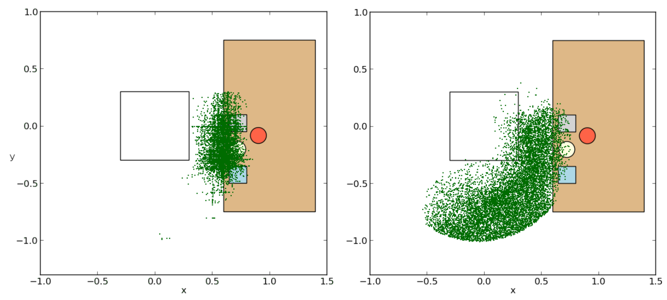

Fig. 2. View from above showing the space covered during the execution of NovEB (left) and during that of the control experiment (right). The representation is composed of the PR2 (white box) in front of the table (brown box). The small circle and boxes represents the objects on the table. The green dots represent the final position of each trajectory executed by the robot. For the control experiment, dots show that a majority of the space within reach of the robot's right arm is searched, whereas NovEB clearly focuses on the interesting parts of the space.

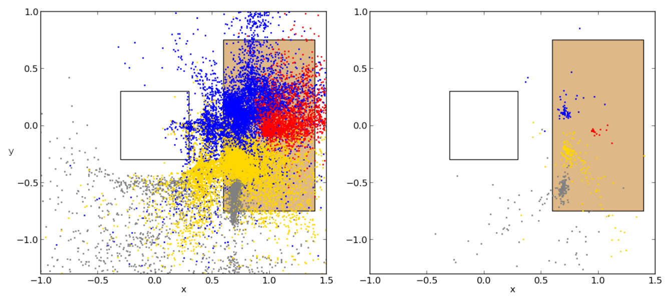

Fig. 3. View from above showing the final positions of the objects during the execution of NovEB (left) and during that of the control experiment (right). The representation is similar to that of Fig. 2. The blue, gray, yellow and red dots represent the final positions of the blue box, of the gray box, of the can, and of the ball respectively. Some dots are located behind the robot due to the simulator's physics engine not always handling correctly the dynamics of the objects. However, this has no impact on the results of the experiments because such objects are out of the field of vision of the robot. Note again that NovEB produces much more changes than the control experiment.

To assess the performance of our approach, the final positions of the objects of the scene have also been recorded (objects and their positions are unknown to the robot).

The experiments have been performed with a simulated PR2 robot[6]. The main features of the robot, regarding our experiment, are two arms with 7-DoF ended up with grippers, and a set of high quality cameras located on its head. The code used for the experiments is available online[7].

ROS Hydro Medusa[8] was used to manage the robot. The simulation has been executed in Gazebo 1.9[9]. MoveIt[10] provides the robot with the capability of defining safe trajectories

[6]https://www.willowgarage.com/pages/pr2/overview

[7]http://pages.isir.upmc.fr/evorob_db/

[8]http://wiki.ros.org/hydro

[9]http://gazebosim.org/

[10]http://moveit.ros.org/

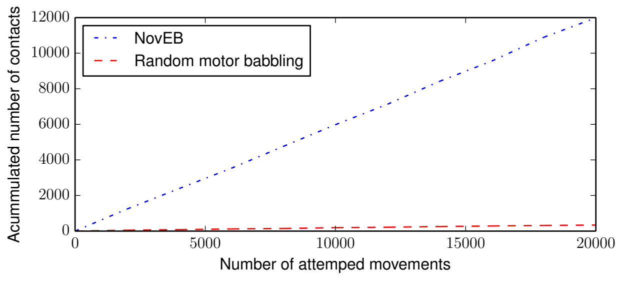

Fig. 4. Comparison of the accumulated number of contacts produced in both experiments w.r.t. the number of attempted movements.

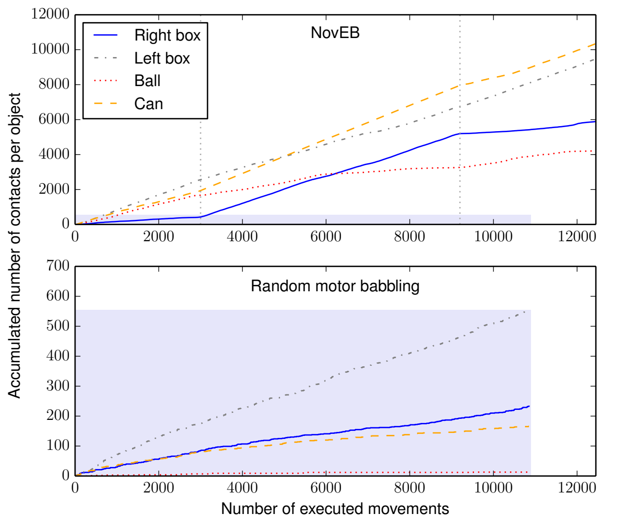

Fig. 5. On top, accumulated number of contacts created by NovEB for each object. Three regions can be distinguished, based on the growth of the number of contacts. At the bottom, accumulated number of contacts created during the control experiment for each object w.r.t the number of executed movements. The shaded areas represent the same area in both figures.

for the end effector, using OMPL, based on a set of points in the space. It also provides collision avoidance. The evolutionary algorithm, on which the execution of this method relies, is executed in Sferes$_{v2}$ [23], a framework for evolutionary computation designed for multi-core parallelization.

B. Control experiment

A random motor babbling is defined as a control experiment, due to its wide use in different works to generate knowledge about the world. In this experiment the motor babbling consists in the definition of different sets of joint values provided to the right arm of the PR2. These values are randomly generated, and used by the robot to perform a movement. The attempted movements are defined sequentially, until reaching the value of 20000. Only the safe and feasible movements are performed, called executed movements.

C. Results

The final positions of the robot's end effector reached during one run of the execution of two experiments are shown in Fig. 2. The space covered by NovEB is focused on the table, while the space explored by the random babbling covers a big part of the joint space of the right arm of the robot. During the exploration performed by NovEB, some final positions are located far from the table (isolated points at the bottom of the left part of Fig. 2). These points correspond to the initial randomly generated solutions, and the exploration around them was stopped as they did not generate novelty.

In this context, a contact is counted when the end effector of the right arm touches at least one object on the table. NovEB generated a high number of contacts with the objects of the scenario (see Fig. 4). When observing the final positions of the objects, it also appears that they spread over a larger portion of the space for NovEB than for the control experiment (see Fig. 3). Therefore, the contacts generated by NovEB produced a high diversity of behaviours in the scenario.

The random babbling generated a small number of contacts with each object (bottom of Fig. 5). In contrast, the number of contacts produced with each object is large in NovEB (top of Fig. 5). For instance, the number of changes produced to the can is over 10000, meaning that in almost each executed trajectory during the exploration the position of the can was modified. This result was expected as the can is located in the center of the table, and interactions with any of the boxes could modify its position. The babbling can be split up into three different phases: (1) At the beginning, the robot gets focused on the objects on the left side of the table. (2) When reaching 3000 executed movements, the robot changes its focus to the can. The growth of the number of interactions with the ball and the left box decreases and the contacts with the can and the right box increase. (3) After 9000 executed movements, the novelty found in the right part of the table decreases, and the robot comes back to search new behaviours in the left side.

VI. Discussion

NovEB can be considered as an intrinsic motivation for exploration, like the Artificial Curiosity defined by Oudeyer *et al.* [2]. One of the main difference between these two approaches lies in the assumptions on which these methods are based. The only requirement for NovEB is to define a distance between two perceptions (in this paper, between two images). This is specific to the robots sensors and is independent from the task or the environment. Conversely, the Intelligent Artificial Curiosity, at least in its current implementation, requires to train predictors in order to estimate the learning progress. Training such algorithms to predict the consequences of an action only on the basis of raw perceptions is a challenge *per se*. For instance, it seems difficult to predict the image that the robot's camera will capture based only on previously captured raw images and the executed actions. Current implementations of Artificial Curiosity rely on higher level information, for example on the position of the objects in the scene [2]. However, providing the tools that extract these higher information from raw perceptions cannot be environment-agnostic. For instance, when predicting the positions of the objects, the algorithm needs to know how many objects compose the scene, or how to extract these objects from the raw perceptions (using large object database, for instance). Based on this observation, these two approaches can be complementary. NovEB can be used to generate a large amount of data that can afterwards be used to extract information from the scene (number or shape of objects, for instance). Then, the high level information extracted can be used to run Intelligent Artificial Curiosity for a detailed or goal-oriented exploration [24].

As NovEB is driven by novelty only, it suffers from some of the limitations that have motivated the development of Intelligent Artificial Curiosity: it should get focused on interactions that generate perceptions with a large variability. Intelligent Artificial Curiosity can avoid this phenomenon as the learning progress in such situations will remain low. This would be a strong limitation for the exploration ability of the system if NovEB was expected to handle the whole developmental process. But this is not an issue, as NovEB is aimed only at acquiring the data to bootstrap other developmental processes.

VII. Conclusion

In this paper, we have proposed a novel method for generating interactions with the objects surrounding a robot through babbling. The approach has been applied on a virtual robot, which discovers on its own which regions of the workspace generate novel perceptions and focuses its exploration around them. The results show that NovEB is able to generate several thousand different interactions with this environment, an order of magnitude higher than the number of interactions produced with a random motor babbling approach. This difference is obtained thanks to the ability of NovEB to focus its exploration in regions that lead to novel visual perceptions.

Acknowledgment

The authors would like to thank Martin Molina for his advice during the elaboration of this work. This research is sponsored in part by the DREAM project[11]. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 640891. This work was performed within the Labex SMART (ANR-11-LABX-65) supported by French state funds managed by the ANR within the Investissements d'Avenir programme under reference ANR-11-IDEX-0004-02. This work has been funded by a DGA/UPMC scholarship for Antoine Cully.

References

[1] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, pp. 139–159, 1991.

[2] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic Motivation Systems for Autonomous Mental Development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[3] K. Sanderson, "Mars rover Spirit (200310)," *Nature*, vol. 463, no. February, p. 2010, 2010.

[4] J. Bongard and R. Pfeifer, *How the body shapes the way we think: a new view of intelligence*, 2007.

[5] J. Lehman and K. O. Stanley, "Abandoning objectives: evolution through the search for novelty alone." *Evolutionary computation*, vol. 19, pp. 189–223, 2011.

[6] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, Dec. 2003.

[7] J. Mugan and B. J. Kuipers, "Autonomous Learning of High-Level States and Actions in Continuous Environments," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 70–86, Mar. 2012.

[8] J. Modayil and B. J. Kuipers, "The Initial Development of Object Knowledge by a Learning Robot." *Robotics and autonomous systems*, vol. 56, no. 11, pp. 879–890, Nov. 2008.

[9] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: a robot developmental pathway," in *Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 2005, pp. 31–37.

[10] P. Gaudiano and S. Grossberg, "Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories," *Neural Networks*, vol. 4, pp. 147–183, 1991.

[11] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, Sep. 1998.

[12] J. Pearl, "Fusion, propagation, and structuring in belief networks," pp. 241–288, 1986.

[13] S. Doncieux, N. Bredeche, J.-b. Mouret, and A. E. Eiben, "Evolutionary robotics: what, why, and where to," *Frontiers in Robotics and AI*, vol. 2, 2015.

[14] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 345–370, 2009.

[15] S. Doncieux and J.-b. Mouret, "Beyond black-box optimization: A review of selective pressures for evolutionary robotics," *Evolutionary Intelligence*, vol. 7, no. 2, pp. 71–93, 2014.

[16] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2008.

[17] A. Cully and J.-b. Mouret, "Evolving a Behavioral Repertoire for a Walking Robot," *Evolutionary computation*, no. 2005, 2015.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[19] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 216–229, 2010.

[20] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, pp. 99–127, 2002.

[21] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," Ph.D. dissertation, Upper Austria University of Applied Sciences, 2010.

[22] R. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.

[23] J.-B. Mouret and S. Doncieux, "Sferes v2: Evolvin' in the multi-core world," *IEEE Congress on Evolutionary Computation*, no. 2, pp. 1–8, Jul. 2010.

[24] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, Jan. 2013.

Appendix

NovEB parameters used in the experiment:

- Population size: 12
- Number of generations: 2000
- Number of k-closest neighbours: 3
- Individuals maintained among populations: 50%
- Probability of mutating an individual: 30%
- Sigma used for the Gaussian mutation: 0.3
- Probability of extending an individual: 5%

[11] http://www.robotsthatdream.eu/