

From Unknown Sensors and Actuators to Visually Guided Movement

Lars Olsson*, Chrystopher L. Nehaniv*†, Daniel Polani*†
Adaptive Systems and Algorithms† Research Groups*
School of Computer Science
College Lane, Hatfield Herts AL10 9AB, United Kingdom
{L.A.Olsson, C.L.Nehaniv, D.Polani}@herts.ac.uk

Abstract—This paper describes a developmental system implemented on a real robot that learns a model of its own sensory and actuator apparatuses. There is no innate knowledge regarding the modality or representation of the sensoric input and the actuators, and the system relies on generic properties of the robot’s world such as piecewise smooth effects of movement on sensory changes. The robot develops the model of its sensorimotor system by first performing random movements to create an informational map of the sensors. Using this map the robot then learns what effects the different possible actions have on the sensors. After this developmental process the robot can perform simple motion tracking.

Index Terms—developmental robotics, body babbling, emergence of structure

I. INTRODUCTION

All higher animals display amazing capabilities and control over their own bodies. They are also able to predict the way in which certain actions affect the environment around them. In order to achieve this, many higher animals go through a number of developmental stages, whereby the nervous system develops and the animal learns to control its own body. In the process of learning a *sense of bodily self* in young infants, systematic exploration of the sensory and perceptual consequences of their actions plays a major role [21]. For example, young infants perform the same actions over and over again [19], and it has been observed that newborn infants spend up to 20% of their time while awake touching their face with their hands [7]. This can be viewed as exploration of their possible action space as well as a method of learning relations between sensor and motor neurons. This phenomenon, in analogy with the vocal babbling in the early process of language learning, is also called *body babbling* [12].

In the research area of developmental robotics [11], there are two major driving forces. One is to build more adaptable,

autonomous, and sociable robots by drawing inspiration from developmental psychology and neuroscience, and the other major driving force is to also use robots as embodied tools to help the investigation of the development of neural systems and cognitive abilities. Here body babbling has been investigated by a number of researchers to explore the possible sensorimotor space of a specific embodiment. For example, in [2] the authors developed a system performing basic visuomotor behaviors which were categorized by a neural network. In [8] body babbling was used to develop a system capable of imitation. Berthouze and Kuniyoshi [1] also developed a system capable of performing unsupervised learning of visual sensorimotor patterns, where the classification used Kohonen maps. The resulting self-organized Kohonen maps showed four different categories of movement; vertical, ‘in-depth motions’, horizontal, and a fourth not clearly defined intermediate category.

In previous work [15],[14], we investigated how an agent can find structure in its sensory apparatus from raw uninterpreted sensory data. The agent has no innate knowledge about what kind of modalities the sensor data come from, and it does not possess a model of its own physical layout or sensor positions. In this work we take the next step by investigating how the agent, once the structure of the sensory system is discovered, can learn to use its unknown motor system to achieve certain effects in its sensors. To make an analogy, consider the main character in Franz Kafka’s novel *Metamorphosis* [6], that wakes up one morning and finds his brain “plugged” in to the body of a gigantic cockroach. The brain receives various sensory input but with no discriminable structure. The main character also notices that he can make various movements but he lacks knowledge of the relations between the motor actions and sensory input and is thus completely lost. The goal of this work is to design a system that can be “plugged” in to an unknown body and learn the relationships between sensors and motors, and after a period of experimentation, manages to perform simple tasks guided by the learned model of its sensors. These ideas are partially inspired by and algorithmically related to work described in

The work described in this paper was partially conducted within the EU Integrated Project RobotCub (“Robotic Open-architecture Technology for Cognition, Understanding, and Behaviours”) and was funded by the European Commission through the E5 Unit (Cognition) of FP6-IST under Contract FP6-004370.

[20]. In our experiments we use a SONY AIBO¹ robot that acts in the real physical world.

This work is also inspired by the more philosophical ideas described in [16], where the authors propose a new theory that tries to answer the major question of what visual experience is and where it occurs. Many traditional theories rest on the idea that the brain somehow produces an internal image of the world. Instead, the authors of [16] propose that seeing is a way of acting to explore the world mediated by the agent's mastery of the laws of sensorimotor contingencies (SMCs), which are a set of rules of interdependence between movement and stimulation of the sensors. Some of these ideas have been translated in to algorithms in [17] and extended in [18].

The structure of this paper is as follows. The next section, section 2, describes the methods used and gives a short overview of information theory and methods to compute the informational distances between sensors. Section 3 describes the performed experiments using a SONY AIBO robot along with a presentation of the results. The final section concludes and discusses some possible future avenues for further work.

II. METHOD

In this section we describe the proposed method. We start by discussing our assumptions about the robot and its environment. Next an overview of the algorithm is presented before each step is presented in detail.

A. Assumptions

The assumptions of the described method are the following: The robot is in an environment where movement has a piecewise smooth effect on the sensory data produced by some of the sensors. For example, most real world scenarios have this effect on visual sensors unless there is complete darkness or the visual input is random. The same is also true for, for example, infra-red sensors that measure the distance to the closest obstacle, and touch sensors. Another assumption is that the robot is capable of computing the motion flow given a sensory layout of sensors, e.g., optical flow for visual sensors.

B. Algorithm

The proposed method can be summarized as follows:

- 1) For t time steps make random movements and save all sensor data as individual time series.
- 2) Compute the informational distances between all sensors.
- 3) Using the sensory reconstruction method [15] a sensoritopic map is created.
- 4) The visual sensors can be found by grouping of sensors with piecewise smooth informational relationship using the sensoritopic map.

- 5) From the visual sensors with continuous coordinates in the sensoritopic map a discrete map is computed.
- 6) A uniformly distributed sample of possible actuator settings are tried a number of times and their effects on the discrete representation of the visual field are saved.
- 7) Given the sensory data from the movements the average effect of each possible actuator setting is computed. The set of all these sensor/actuator combinations are the sensorimotor laws.
- 8) The set of sensorimotor laws can now be used to perform visually guided movement, like motion tracking and very simple imitation.

The input data to the sensory reconstruction method is a number of time series of sensory data, one for each sensor s . Each element in each time series is a real value number normalized in the range $[0.0, 1.0]$. Thus, there are n time series s_1, s_2, \dots, s_n , one for each sensor, each t time steps long. It is important to note that these time series do not need to be ordered depending on modality, since it is the algorithm that should find the relationships between different sensors.

C. Random Movements (1)

At this stage nothing about either sensors or actuators is known. The robot should move more or less at random, trying out many different settings of its actuators, without making abrupt changes in too many actuator settings at once, in order to maintain relative continuity. This step is dependent on the particular robot used.

D. Information Distances Between Sensors (2)

Central to the method described is the informational distance between two sensors. This distance is computed using the *information metric* [4]. To understand this distance metric a brief introduction to information theory [22] is presented here.

Let \mathcal{X} be the alphabet of values of a discrete random variable (information source, in this case a sensor) X with a probability mass function $p(x)$, where $x \in \mathcal{X}$. Then the entropy, or uncertainty associated with X is

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \quad (1)$$

and the conditional entropy

$$H(Y|X) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \quad (2)$$

is the uncertainty associated with the discrete random variable Y if we know the value of X . In other words, how much more information do we need to fully predict Y once we know X .

The *mutual information* is the information shared between the two random variables X and Y and is defined as

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \quad (3)$$

¹AIBO is a registered trademark of SONY Corporation.

To measure the dissimilarity in the information in two sources Crutchfield's information distance [4] can be used. The *information metric* is the sum of two conditional entropies, or formally

$$d(X, Y) = H(X|Y) + H(Y|X). \quad (4)$$

Note that X and Y in our system are information sources whose $H(Y|X)$ and $H(X|Y)$ are estimated from the time series data from each sensor using equation 2.

E. Sensory Reconstruction Method (3)

The sensory reconstruction method is a learning algorithm used to find groups of related sensors and a spatialized description of the layout and dimensionality of the sensors, first described in [20] and extended in [15]. For example, if all the sensors in the input are vision sensors from the same visual field, the result should be a two-dimensional layout of the sensors with their relative positions. The input to the algorithm is raw uninterpreted data from a number of different sensors where each sensor is seen as a time series of sensor readings where each element is a normalized real value number. Given these sensors with their time series of data the distance between each pair of sensors is computed using the classical distance metrics [20] or the information metric where each sensor is treated as an information source.

Figure 1 shows an example of a sensoritopic map with created with sensor data from a SONY AIBO robot moving around in an office environment.

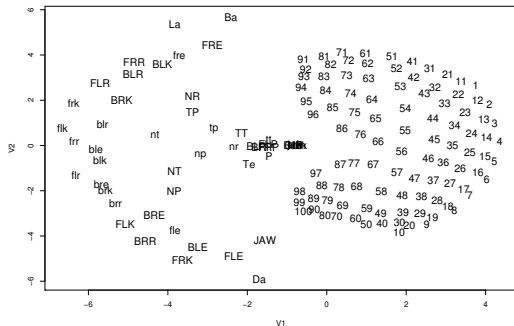


Fig. 1. A sensoritopic map created by the sensory reconstruction method taken from [15] using the information metric. In this example there are 150 sensors, including 100 image sensors that are labeled 1-100 to the right in the map.

F. Grouping of Similar Sensors (4)

Given the sensoritopic map, the next step is to find a set of these sensors on which movement has piecewise smooth effects. To begin with, find groups of sensors that are similar. Let sensor x_i and x_j be similar, written

$$x_i \approx x_j \quad \text{if} \quad d(x_i, x_j) \leq \min\{\epsilon_i, \epsilon_j\} \quad (5)$$

where the ϵ s are calculated from the minimum distance to any of its neighbours, $\epsilon_i = 2 \min_j \{d(x_i, x_j)\}$.

To form closed subgroups, [20] use the *related-to* relation which is the transitive closure of the \approx relation. This is computed recursively via

$$i \sim j \quad \text{iff} \quad i \approx j \vee \exists k : (i \sim k) \wedge (k \sim j). \quad (6)$$

Given a closed subgroup its dimension can be computed. Let $\sigma^2(m)$ be the variance accounted for by dimension m . Then the right number of dimensions can be found by maximizing $\sigma^2(m) - \sigma^2(m+1)$. For example, for a set of visual sensors arranged in a two-dimensional grid $m = 2$.

G. From Continuous to Discrete Maps (5)

To compute a discrete map from the continuous coordinate map from step 3 we have developed a quite trivial algorithm that, as it has turned out, is quite effective. The algorithm works as follows. Order all points according to their position along the x-axis in the projection with the smallest x-axis value first and add them to the list L_{all} . Pick the first \sqrt{n} points from L_{all} and order them according to their position along the y-axis in to list $column_k$, where k is counter that is incremented by 1 each \sqrt{n} points picked from L_{all} . Then delete the points in $column_k$ from L_{all} . Repeat until L_{all} is empty. Now each list $column_k$ contains the points for one column of the image ordered by k . This assumes that the layout is square. For rectangular layouts first the number of columns and rows must be found.

H. Sampling of Actuator Settings (6)

This step, again, depends on the particular robot used. The aim is to sample, in as much detail as possible, the set of possible actuator settings, to understand how all the different actuator settings affect the sensors. As a simple example, consider a robot with two actuators, a_1 and a_2 , where $a_1, a_2 \in \{-1, 0, 1\}$. This means that the whole actuator space consists of only 9 possible settings. In most robots this space is much larger, with more actuators, and more possible values (maybe continuous) for each actuator. There might also be a many-to-one mapping between the actuator vectors and the actual values sent to the motor, which means that more than one actuator vector might give the same effect in the motors.

Also, for better results each movement needs to be performed several times, resulting in a number of samples of sensor data for each possible movement. This is necessary since optical flow calculations are very sensitive to noise and the structure of the environment. There might also be moving objects in front of the robot, something that will seriously affect the optical flow calculation.

I. Discovering Sensorimotor Laws (7)

Each time a particular actuator setting is executed as described in the previous step the sensor readings for each time step are saved for the visual field discretized in step 5. Given

this data the average effect of that particular actuator settings (and resulting movement) can be computed by averaging the effect on the sensors for all the times that particular movement was performed. The effect on each sensor is the optical flow computed using the algorithm described in [9]. This requires the discrete map obtained in step 5.

Let $\vec{v}_{x,t,s}$ be the optical flow in the sensor x at time step t for actuator setting s , where the sensor readings from all performances of a particular actuator setting have been saved as one long time series. The average effect for actuator setting s , $\langle \vec{v}_s \rangle$, is then

$$\langle \vec{v}_s \rangle = \frac{\sum_{x=1}^N \sum_{j=t}^T \vec{v}_{x,t,s}}{NT}, \quad (7)$$

where N is the number of sensors and T the number of time steps actuator setting s has been performed.

With more complex movements, where the optical flow field is different in different parts of the field, the last averaging step can be omitted and the average effect of a certain actuator setting is then a vector field and not one vector.

J. Visually Guided Movement(8)

The set of sensorimotor laws can now be used to perform actions where a specific effect on the sensors is desired. For example, if the robot sees a moving object the object can be tracked by performing the action from the set of sensorimotor laws that has the inverse effect on the sensors as the observed object.

For example, consider the sensorimotor laws of figure 3, each associated with a particular actuator setting. To track a particular motion, the robot needs to find the sensorimotor law with effect closest to the perceived movement. Given a set of learned sensorimotor vectors V , find the vector $\vec{v} \in V$, with the shortest distance to the perceived visual movement \vec{m} , given some distance function d between two vectors. This minimization, $\min_{\vec{v} \in V} d(\vec{v}, \vec{m})$, is maybe most naturally computed using $d = |\vec{v} - \vec{m}|$, which computes the distance between the end points of the two vectors. One could also imagine other distance metrics, for example one where the direction is more important than the speed. When the desired vector \vec{v} is found the robot can then perform the movement associated with \vec{v} .

III. EXPERIMENTS AND RESULTS

A. Experimental Set-up

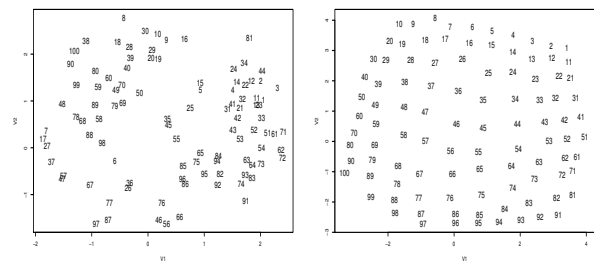
In our experiments we used a SONY AIBO robot interacting in a lab environment. The software was executed remotely on a computer connected to the robot over a dedicated wireless network, using a frame-rate of roughly 25 frames per second for the sensor readings.

The AIBO has many degrees of freedom and to simplify the experiments we decided to only use the pan (horizontal

movement) and tilt (vertical) motors of the head of the AIBO. To put less strain on the neck of the AIBO (which is the component that most often breaks down), we let the maximum pan and tilt be 0.5 and the minimum -0.5 . To move the neck the desired position of the pan and tilt motor are set 100 times per second, adding the value of the corresponding actuator setting each time step, until the maximum or minimum value of the pan or tilt have been reached. In our experiments the possible desired speed, s , for each of the two actuators was limited to five different values, $s \in \{-0.01, -0.02, 0.0, 0.01, 0.02\}$. The sensors values of each sensor was normalized to values between $[0.0, 1.0]$, similar to [15]. The AIBO was placed in a sitting position overlooking the lab. Each possible actuator vector was executed 15 times.

B. Results

Figure 2 shows an example of the development of the sensoritopic map of the visual sensors after 50 time steps, where some order can be found, and after 500 time steps, where an approximate layout of the visual sensors has been found.



(a) 50 frames

(b) 500 frames

Fig. 2. Development of the sensoritopic map of the visual sensors.

From this sensoritopic map a discrete map of the visual sensor can be computed using the algorithm described in step 5. Then the possible actuator space was explored as described above. Each possible movement was executed 15 times and the optical flow for each frame computed. The optical flow in each pixel between consecutive frames was highly variable. This high variance has several reasons. One might be the optical flow algorithm used and another one the light sensitivity of the camera on the AIBO. The properties of the static environment are also important, e.g., light conditions and the colouration of walls and objects, as well as the informational properties of the scene [13]. Of course, moving objects in the environment also highly affect the result.

As described before, each movement was performed many times to overcome these problems, and the vectors averaged.

The vectors for all sensors are finally averaged for each possible actuator setting. Figure 3 shows 16 of the resulting vectors (the other 8 are omitted for clarity). Vector v_1 and v_9 correspond to the motor settings $(0.02, 0.0)$ and $(0.01, 0.0)$, respectively. The angles of the vectors do not correspond exactly to the expected value. For example, if the AIBO is sitting straight up and is only executing its pan engine, vectors v_1 , v_9 , v_5 , and v_{13} , the expected angle of the corresponding vector should be zero. In figure 3 this is only approximately the case. This, again, is probably due to noise in the environment, the angle of the head while it is moving, and maybe vibrations in the head.

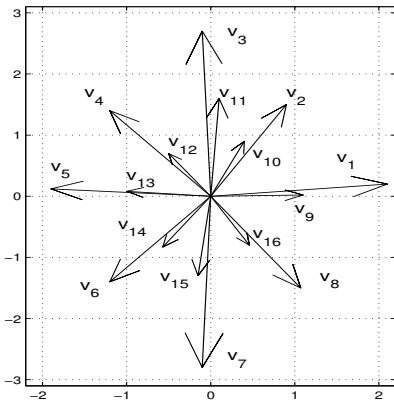


Fig. 3. **Sensorimotor laws.** The computed sensorimotor laws ($v_1 - v_{16}$) for 16 different actuator settings, where $v_1 - v_8$ are actuator settings resulting in faster movements. In this graph v_3 and v_{11} corresponds to tilt movements straight down and v_5 and v_{13} to pan movements to the right (from the robot's perspective).

Given the motion flow vectors in figure 3 and their associated actuator settings it is now possible to perform visually guided movement, like object tracking or simple motion imitation. First observe a specific flow in the visual sensors, then find the closest sensorimotor vector using the method in step 8 of the algorithm. Finally execute the actuator settings associated with the closest sensorimotor vector. Figure 4 shows the AIBO with an experimenter moving a pink ball from the right to the left of the visual field of the robot. Figure 5 shows an example of the visual input and the pixelated version of the input. The robot captured 20 frames of data and then calculated the average effect on the visual sensors. In figure 6 the flow of the hand and ball between two consecutive frames is shown. The average motion vector is found by averaging all flow vectors with a magnitude greater than some small constant to get make the computation more robust to noise and to only take in to account the parts of the visual field where there is motion. If the robot is moving at the same time, the average motion vector given the actuator settings from 3 can be subtracted from each vector in the flow field. Experiments showed that the robot using this algorithm was able to perform simple motion tracking.



Fig. 4. The SONY AIBO with a ball moved from right to left.



(a) Raw image (b) Pixelated image

Fig. 5. Example of visual input. Figure 5(a) shows the raw input and figure 5(b) the pixelated version.

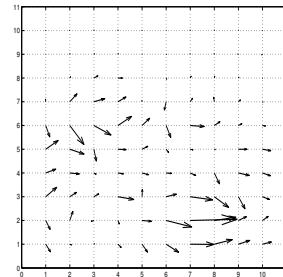


Fig. 6. Flow when hand and ball moves to the left.

IV. CONCLUSIONS AND FUTURE WORK

This paper has presented a robotic control system that learns a model of its sensors and actuators starting from raw and uninterpreted sensor data. The robot also lacks innate knowledge of the modality of the sensors and the effect of the actuators. It learns by first creating a sensoritopic map of the sensoric input created from random movement. From this map a subset of the sensors are selected that represent the visual field or possibly another two-dimensional sensor array like touch sensors. Using this visual field the the effects of the actuators are found by learning a model of how different actuator settings affect the visual sensors. The results show

that the robot can perform simple motion tracking after the developmental training period is completed.

The learning of how the actuators affect the sensors exploits the correlations between the actuators and sensors found by exploring the sensorimotor space of the robot, similar to the body babbling [12] found in human infants and maybe other higher animals. The correlations between sensors and actuators seem to be important for both robots and animals. In [24] Sporns and Edelman hypothesize about how exploratory activity may cause correlations between sensory and motor neurons, something that is important for reducing the number of possible degrees of freedom in motor control (Bernstein's problem). It might also be advantageous to explore the correlations from an adaptive point of view; if the embodiment changes, e.g., growth or bodily injuries, or if moved to another environment or faced with a complete new task, the motor system needs to adjust its behaviour.

In [16] the authors propose that seeing is a way of acting to explore the environment, which is mediated by the agent's mastery of the laws of sensorimotor contingencies. In an interesting commentary, [3], the author proposes three experiments that could be conducted to validate the proposed theory. One of the proposed experiments is to use goggles that completely scramble the vision, instead of only reversing or inverting as in the traditional experiments with inverting goggles [5]. It would, if the theory of sensorimotor contingencies is correct, be possible to learn to use the scrambling goggles and learn to de-scramble the image. In [15], it was showed that this is possible using the sensory reconstruction method described above.

Several avenues for future research lead from the presented work. Naturally it would be interesting to increase the resolution of the actuators, as well as the number of actuators, possibly having each motor in each leg be one actuator. Here it would be interesting to explore how limiting the number of degrees of freedom and adding more degrees of freedom as the development continues affect learning, similar in spirit to the work in [10]. It also seems like constraints on perception during development may improve the perceptual efficiency by reducing the information complexity of the perceived environment [23]. This is something that will be explored in future work. Another interesting area to explore is how body babbling can help a robot adapt to different kinds of mechanical or sensory failures.

REFERENCES

- [1] L. Berthouze and Y. Kuniyoshi, "Emergence and categorization of coordinated visual behavior through embodied interaction," *Machine Learning*, vol. 31, pp. 187–200, 1998.
- [2] L. Berthouze, Y. Shigematsu, and Y. Kuniyoshi, "Dynamic categorization of explorative behaviors for emergence of stable sensorimotor configurations," in *Proceedings of the International Conference on Simulation of Adaptive Behavior (SAB1998)*, 1998, pp. 67–72.
- [3] S. J. Blackmore, "Three experiments to test the sensorimotor theory of vision: commentary on a sensorimotor account of vision and visual consciousness by J. Kevin O'Regan and Alva noe," *Behavioral and Brain Sciences*, vol. 24, no. 5, pp. 1011–1031, 2001.
- [4] J. P. Crutchfield, "Information and its Metric," in *Nonlinear Structures in Physical Systems – Pattern Formation, Chaos and Waves*, L. Lam and H. C. Morris, Eds. Springer Verlag, 1990, pp. 119–130. [Online]. Available: citeseer.nj.nec.com/crutchfield90information.html
- [5] C. S. Harris, "Perceptual adaptation to inverted, reversed, and displaced vision," *Psychological review*, vol. 72, no. 6, pp. 419–444, 1965.
- [6] F. Kafka, *The Transformation (Metamorphosis) and Other Stories*. Penguin Books, 1992/1915.
- [7] A. Korner and H. Kraemer, "Individual differences in spontaneous oral behavior in neonates," in *Proceedings of the 3rd Symposium on Oral Sensation and Perception*, J. Bosma, Ed., 1972, pp. 335–346.
- [8] Y. Kuniyoshi, Y. Yoroze, M. Inaba, and H. Inoue, "From visuo-motor self learning to early imitation - a neural architecture for humanoid learning," in *International conference on robotics and Automation*, 2003, pp. 67–72.
- [9] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.
- [10] M. Lungarella and L. Berthouze, "On the interplay between morphological, neural, and environmental dynamics: a case-study," *Adaptive Behaviour*, vol. 10, no. (3/4), pp. 223–241, 2002.
- [11] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2004.
- [12] A. Meltzoff and M. Moore, "Explaining facial imitation: a theoretical model," *Early development and parenting*, vol. 6, pp. 179–192, 1997.
- [13] L. Olsson, C. L. Nehaniv, and D. Polani, "The effects on visual information in a robot in environments with oriented contours," in *Proceedings of the Fourth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, August 25-27th Genoa, Italy*, L. Berthouze, H. Kozima, C. G. Prince, G. Sandini, G. Stojanov, G. Metta, , and C. Balkenius, Eds. Lund University Cognitive Studies, 2004, pp. 83–88.
- [14] —, "Information trade-offs and the evolution of sensory layouts," in *Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9) September 12-15th 2004 Boston, Massachusetts, USA*. MIT Press, 2004, pp. 119–124.
- [15] —, "Sensory channel grouping and structure from uninterpreted sensor data," in *2004 NASA/DoD Conference on Evolvable Hardware June 24-26, 2004 Seattle, Washington, USA*. IEEE Computer Society Press, 2004, pp. 153–160.
- [16] J. O'Regan and A. Noe, "A sensorimotor account of vision and visual consciousness," *Behavioral and Brain Sciences*, vol. 24, no. 5, pp. 1011–1031, 2001.
- [17] D. Phillipona, J. O'Regan, and J. Nadal, "Is there something out there? inferring space from sensorimotor dependencies," *Neural Computation*, vol. 15, no. 9), 2003.
- [18] —, "Perception of the structure of the physical world using multimodal unknown sensors and effectors," *Advances in Neural Information Processing Systems*, vol. 17, 2004.
- [19] J. Piaget, *The origins of intelligence*. Routledge, 1953.
- [20] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, pp. 169–229, 1997.
- [21] P. Rochat, "Self-perception and action in infancy," *Experimental Brain Research*, vol. 123, pp. 102–109, 1998.
- [22] C. E. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
- [23] A. Slater and S. Johnson, "Visual sensory and perceptual abilities of the newborn: beyond the blooming, buzzing confusion," in *The development of sensory, motor, and cognitive abilities in early infancy: from sensation to cognition*, F. Simion and G. Butterworth, Eds. Hove: Psychology Press, 1997, pp. 121–141.
- [24] O. Sporns and G. Edelman, "Solving Bernstein's problem: a proposal for the development of coordinated movement by selection," *Child Development*, vol. 64, no. 4, pp. 960–981, 1993.