RobotDoC

Robotics for Development of Cognition

Proceedings

of the

Post-Graduate Conference on Robotics and Development of Cognition

10-12 September 2012
Lausanne, Switzerland

Edited by

Joanna Szufnarowska
Bielefeld University

Table of Contents

Abstract Submissions

Introduction

The aim of the Postgraduate Conference on Robotics and Development of Cognition (RobotDoC-PhD) was to bring together young scientists working on developmental cognitive robotics and its core disciplines. The conference aimed to provide both feedback and greater visibility to their research as lively and stimulating discussion were held amongst participating PhD students and senior researchers. The conference was open to all PhD students and post-doctoral researchers in the field. RobotDoC-PhD conference was an initiative as a part of Marie-Curie Actions ITN RobotDoC and was organized as a satellite event of the 22nd International Conference on Artificial Neural Networks ICANN 2012 held in Lausanne, Switzerland.

How to cite a paper in these proceedings

Please follow the example below, presented in the IEEE format:

[1] J. de Greef, P. Baxter, R. Wood, and T. Belpaeme, “From penguins to parakeets: a developmental approach to modelling conceptual prototypes", in *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, J. Szufnarowska, Ed., September 2012, pp. 8–11.

Organizing Committee

General Chairs

Angelo Cangelosi
Francesca Stramandinoli
Stefan Wermter

Program Chairs

Marek Ruciński
Vikram Narayan

Publication Chair

Joanna Szufnarowska

Financial Chair

Cristiano Alessandro

Publicity Chairs

Naveen Kuppuswamy
Claudia Elsner
Gauss Lee

Local Chairs

Nicolas Navarro-Guerrero
Junpei Zhong

Web administrators

Claudia Elsner, Gauss Lee

Program Committee

Paul Baxter
Tony Belpaeme
Frank Broz
Angelo Cangelosi
Hugo Gravato Marques
Andrea Handl
Kiril Kiryazov
Naveen Kuupuswamy
Qiang Li
Robert Lowe
Davide Marocco
Giorgio Metta
Francesco Nori
Alessandro di Nuovo
Katharina Rohlfing
Wiktor Sieklicki
Marko Tscherepanow
Cornelius Weber
Stefan Wermter
Nick Wilkinson
Johannes Winke
Britta Wrede
Tom Ziemke

Sponsors

RobotDoC-PhD Conference organizers would like to sincerely thank the sponsors for their support of the conference awards, for covering the expenses of the invited speakers, and for supporting students' participation through travel bursaries.

EUCog - European Network for the Advancement of Artificial Cognitive Systems, Interaction and Robotics

www.eucognition.org

RobotDoc, the Marie Curie doctoral training network in developmental robotics

www.robotdoc.org

Cyberbotics – professional mobile robot simulation

www.cyberbotics.com

Cognitive Science Society

cognitivesciencesociety.org

Marie Curie Actions

ec.europa.eu/research/mariecurieactions/

Invited Speakers

Minoru Asada
Department of Adaptive Machine Systems
Graduate School of Engineering
Osaka University
`www.er.ams.eng.osaka-u.ac.jp`

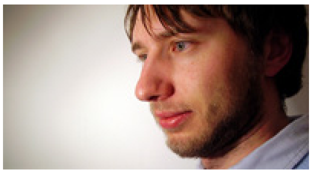
Malinda Carpenter
Department of Developmental and Comparative Psychology
Max Planck Institute for Evolutionary Anthropology
`www.eva.mpg.de/psycho/staff/carpenter/`

Lionel Boillot
Marie Curie Actions
Research Executive Agency
European Commission

Short Biographies of Authors

Beata J. Grzyb received her MSc Degree in Computer Science from Maria Curie-Skłodowska University in Lublin, Poland. Currently she is a PhD candidate in the Robotic Intelligence Lab, Jaume I University, and has already published in several journal and proceedings. In her research, she follows the approach of cognitive developmental robotics, and tackles problems related to body representation, peripersonal space representation and perception of body effectivities, by means of synthesizing neuroscience, developmental psychology, and robotics.

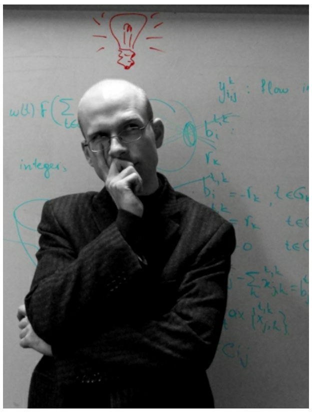
Xavier Hinaut received an M.S. in Computer Science from the University of Technology of Compiègne, France in 2008 and an M.S. in Cognitive Science from the Ecole Pratrique des Hautes Etudes, France in 2009.
Since 2009, he is pursuing a PhD in Computational Neuroscience at the INSERM Stem Cell and Brain Research Institute in Lyon, France. He works in the Cortical Networks for Cognitive Interaction team under the supervision of PhD Peter Ford Dominey. His research explores brain mechanisms of complex sequence processing, focusing on language syntactic comprehension. He is interested in using Recurrent Neural Networks to model these processes and investigating how such models can be used in developmental robotics.



Jürgen Leitner is a PhD candidate at the Università della Svizzera Italiana and the researcher at the Dalle Molle Institute for AI (IDSIA) in the field of Robotics and Artificial Intelligence (AI). He has a background in Computer Science and graduated from the "Joint European Master in Space Science and Technology" programme in the field of Space Robotics and Automation. He has been working at various universities and for a year at the European Space Agency before joining IDSIA.

Zoltán-Csaba Márton obtained a degree in Industrial Informatics from the Technical University of Cluj-Napoca. Since 2007 he is at the Technische Universität Müchen, working first on his Diploma thesis (M.Sc. equivalent), then his Ph.D. at the Intelligent Autonomous Systems Group. He is involved in the development of the Point Cloud Library and its predecessors since the beginnings in 2007. His main research interests include semantic perception, multi-cue classification, and 3D model fitting.

Avinash Ranganath is a PhD candidate at the Robotics Lab in Universidad Carlos III de Madrid. He received his bachelors degree in Computer Application from Bangalore University, and a masters degree in Artificial Intelligence from University of Edinburgh. His research interests include Collective Intelligence, Embodied Intelligence, Swarm/Modular Robotics, Morphological Computation and Evolutionary Robotics.



Marek Ruciński is a Marie Curie Early Stage Researcher at the Centre for Robotics and Neural Systems, Plymouth University, UK. His research interest is in using humanoid robots to understand better the working of the human brain. In particular his studies focus on robotic modelling of mathematical cognition, with special emphasis on the role of embodiment in mathematical thinking. This includes such topics as acquisition of the number concept and interaction of numbers with other representations in the brain. Before coming to Plymouth Marek worked for the Advanced Concepts Team of the European Space Agency. His original background is computer science, in which he graduated from the Poznań University of Technology, Poland.

Francesca Stramandinoli has a BEng in Computer Science Engineering and a MEng in Automation Engineering from the University of Calabria (Italy). She worked as External Expert at the Department of Mathematics of the University of Calabria, where she dealt with the design of CNN-based algorithms for medical images segmentation and the numerical integration of nonlinear partial differential equations. Currently, she works as Marie Curie Early Stage Researcher at the University of Plymouth (United Kingdom) in the European Project RobotDoC. Her research topic is about the grounding of language in humanoid robots with particular attention to abstract words.

From Penguins to Parakeets: a Developmental Approach to Modelling Conceptual Prototypes

Joachim de Greeff, Paul Baxter, Rachel Wood and Tony Belpaeme

Centre for Robotics and Neural Systems
University of Plymouth, United Kingdom
joachim.degreeff@plymouth.ac.uk

Abstract

The use of concepts is a fundamental capacity underlying complex, human-level cognition. A number of theories have explored the means of concept representation and their links to lower-level features, with one notable example being the Conceptual Spaces theory. While these provide an account for such essential functional processes as prototypes and typicality, it is not entirely clear how these aspects of human cognition can arise in a system undergoing continuous development - postulated to be a necessity from the developmental systems perspective. This paper seeks to establish the foundation of an approach to this question by showing that a distributed, associative and continuous development mechanism, founded on principles of biological memory, can achieve classification performance comparable to the Conceptual Spaces model. We show how qualitatively similar prototypes are formed by both systems when exposed to the same dataset, which illustrates how both models can account for the development of conceptual primitives.

Index Terms: Concepts, prototypes, typicality, Conceptual Spaces, Distributed Associative and Interactive Memory

1. Introduction

For a cognitive system to be able to perform at a level that is comparable to humans, it should be able to form conceptual structures as part of its knowledge representation capacities. As concepts are recognised as being important for many aspects of cognition, it is paramount for an artificial system to be able to model conceptual knowledge, including the formation of prototypes.

In this paper we examine two frameworks for modelling human knowledge; one is based on Conceptual Spaces (CS) [1] and the other, Distributed Associative and Interactive Memory (DAIM), is centred around the distributed nature of human memory and the temporal aspects of its functioning [2, 3]. As they are focussed on different aspects of human knowledge these frameworks have both virtues and drawbacks. CS inherently models knowledge as summary representations which makes it natural to model some of the more generic properties of concepts. However, a CS is a rather static structure and from a developmental perspective it is less clear how well a CS would capture conceptual learning over time. Also, there are no inherent temporal aspects in the model that could account for some of the temporal aspects of human memory, thus a conceptual space is more abstract as a model of human cognition. DAIM on the other hand takes a more developmental approach and emphasises the low level associative and temporal properties of human knowledge acquisition. The question of reconciliation of the two approaches thus arises: can the developmental DAIM perspective be used to account for the structures and functions hypothesised by CS models? This paper seeks to address this question by applying both approaches to the same data set, to assess the compatibility of DAIM with CS.

As an example case, we explore the ability of both frameworks to model an aspect that is considered fundamental to human-like knowledge representation, namely the formation of prototypes which display *typicality* [4]. The observation by Rosch that many everyday concepts are prototypical in nature challenged the established notion in cognitive science that concepts could be modelled using logical definitions[1]. Rosch showed that many concepts cannot be logically defined because they show typicality, that is, people judge certain instances of a specific concept to be more typical than others. For example, for the concept BIRD, a robin is thought to be more "bird-like" than a penguin, a banana is more typical for FRUIT than a pomegranate etc. It turned out that instances of a concept exhibit a graded membership to an idealised prototype, so that some instances are seen as more typical of the concept than others.

Theories advocating this prototypical view of concepts have been around for quite a while with many different flavours [5, 6], but the general gist is that concepts are represented as some kind of idealised version of the specific concept. So, for the concept BIRD people would have an idea of the idealised bird, and match any encounters they have in the real world to this prototype version. The more similar a particular observation is to the prototype, the more they are inclined to assign this observation as belonging to the prototype. It may seem unlikely that all members of BIRD could be represented by one single prototype, given the wide variety of birds. So a prototype should be thought of as a *summary representation*, which specifies the properties of the concept, where some properties are more important than others. These properties are not strictly necessary, but rather they describe what members of the concept tend to have. The process of identifying an object in the world entails a matching to known prototypes. This matching takes the form of a similarity measurement, rather than a logical "does it ticks the boxes?" type of analysis. A prototypical account provides a more naturalistic explanation of human data than a definitional approach.

[1] The idea that concepts can be represented as a list of logical definitions which specify necessary and sufficient conditions is commonly known as the Classical Theory.

2. Theory of the two frameworks

2.1. Theory of Conceptual Spaces

A conceptual space consists of a geometrical representation in vector space along various quality dimensions. A CS is a collection of one or more domains (like colour, shape, or tone), where a domain is postulated as a collection of inseparable sensory-based quality dimensions with a metric. Examples of quality dimensions are weight, temperature, brightness, pitch, loudness, and RGB values. For instance, to express a point in the colour domain using RGB encoding, the different quality dimensions *red*, *green*, and *blue* are all necessary to express a certain colour and are therefore inseparable. Other domains may consist of one or more quality dimensions. In its simplest form, a concept can be represented as a point in the conceptual space, where the coordinates of the point determine the features of the concept. For example, an instance of the concept RED may be represented as a point (255, 0, 0) in the RGB colour domain.

Crucially to modelling concepts in a CS is the ability to take a distance measurement. For each of the dimensions involved, a suitable metric to calculate distance between coordinates on this dimension must be defined. For a lot of dimensions the Euclidean distance may be the most appropriate one, but the Manhattan distance can also be used.

The notion of prototypes comes naturally to conceptual space modelling, as the inherent distance metric can easily function as a notion of typicality. Distance d_{xy} between a prototype x and and an example y takes the general form:

$$d_{xy} = \left(\sum_{i=1}^{N} w_i |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (1)$$

where r denotes the type of metric with $r = 1$ for the Manhattan distance and $r = 2$ for the Euclidean distance and w an optional weight of the dimension. To do justice to psychological evidence of how people tend to rate concepts [7, 8], we can convert the distance into a similarity measurement. Similarity s between i and j is computed as an exponentially decaying function of distance:

$$s_{ij} = e^{-cd_{ij}} \quad (2)$$

where c is a sensitivity parameter.

Within a conceptual space we can model the learning of prototypes by exposing the model to examples with associated labels. After the learning the model is able to classify new examples as belonging to some known class, and specify how typical the example is, i.e. to what extent it belongs to the class and to other learned classes.

2.2. Theory of the Distributed Memory Model

The DAIM system operates on a set of functional principles derived from the operation of memory within biological system, embedded within the context of a wider cognitive system [9, 3]. These are as follows [3]: (1) memory as being fundamentally associative; (2) memory, rather than being a passive storage device, is an active component in cognition through activation dynamics; (3) memory as having a distributed structure; and finally (4) activation-based priming as subserved by the first three points. A DAIM model has been implemented that embodies each of these principles of operation.

Assuming that this memory system is embedded within a wider agent cognitive system with multiple sensory and motor modalities, associations may be formed based on the experiences of the agent, which subsequently form the substrate for activation dynamics. Prior experience as encoded in associative networks, i.e. memory, thus play an active role in the generation of ongoing behaviour through the mechanism of priming, which is the reactivation of modality-specific representations on the basis of existing associations. These principles may be used to provide candidate mechansims for a wide range of cognitive phenomena, from visual recognition and analogies [10, 11], to episodic memory, language development and social interaction [9].

In this study, the notional 'embodiment' of the DAIM system is modelled by an idealised set of inputs i.e. the properties given in the dataset. Associations are formed between input properties, on the basis of activation dynamics (where a high activation level is assigned to a property that is present). These associations have a weight value that is manipulated throughout the operation of the system. This introduces a significant temporal effect, in that an association is continually subject to change based on the relative activation levels of the things it associates, using a Hebbian-like update mechanism. Thus, by extension, the order of learning also has effect on the behaviour of the system.

Implementation of the model is based on an extension to an Interactive Activation and Competition (IAC) model of face learning [12], and uses an explicit representation for associations: i.e. an association is encoded as an object[2], following [13]. While details of this implementation are excluded here due to space constraints, the following description outlines the primary mechanisms.

The weight update mechanism incorporates both Hebbian and anti-Hebbian rules, and essentially has the effect of turning the DAIM implementation into a pseudo-correlation engine, in which the strength of the weights encoding conjunctions of input features essentially reflects the correlation of those features based on prior experience. It should be noted that this is not a correlation in the proper sense, but only an analogue thereof, given the incremental update nature of the weight adjustment. Activation dynamics are also at play, with all input properties having an associated activation level. Activation for a particular property rises if it is present, and falls in the absence of stimulation (i.e. activation decay, to a negative activation 'resting' state). It should be noted that such stimulation can be sourced either from external stimulation, or from the result of activation flowing through already existing associations. A new association is formed between two properties if an association does not already exist, and if the activation of both properties is above zero.

3. Modelling prototypes using the dataset

To examine how both models are able to build conceptual structures that exhibit prototypes and typicality effects, we use the Zoo Data Set from the UCI Machine Learning Repository [14] which is a simple database containing 101 example animals with 16 different properties (like airborne, aquatic, predator etc.) divided into 7 classes. All properties are binary, except for the 'number of legs'. This property is normalized as to make it more in line with the other properties. Both models are exposed to a subset of this data (50 animals), and the resulting knowledge structures are compared by using a further non-

[2] In the context of Object-Oriented Programming.

Table 1: Typicality ratings of the CS model for the 10 examples from the test set.

example	MA	BI	FI	AM	INS	INV
moth	0.08	0.12	0.05	0.09	**0.49**	0.12
newt	0.11	0.14	0.21	**0.37**	0.09	0.13
octopus	0.06	0.07	0.09	0.13	0.12	**0.32**
opossum	**0.37**	0.08	0.08	0.11	0.07	0.06
oryx	**0.53**	0.07	0.06	0.07	0.07	0.05
ostrich	0.10	**0.25**	0.08	0.09	0.10	0.08
parakeet	0.07	**0.39**	0.07	0.07	0.13	0.06
penguin	0.08	**0.20**	0.11	0.13	0.07	0.10
pheasant	0.07	**0.57**	0.08	0.09	0.15	0.08
pike	0.08	0.08	**0.40**	0.13	0.05	0.10

overlapping subset of the data as probes[3]. For the CS the training data is provided with an associated word label that specifies the class, while for the DAIM system the class label and the class type as a numerical value are supplied in the same fashion as the 16 other properties, thus this system is exposed to 18 properties per example. The test data contains 50 examples, where the breakdown into classes is as follows: 24 MAMMAL, 7 FISH, 9 BIRD, 4 INVERTEBRATE, 1 AMPHIBIAN and 5 INSECT.

3.1. Assesment

After training the systems are tested with 10 examples that are not part of the training set. Based on the learned information, an assessment of which category a newly presented instance belongs to is made. To examine the typicality ratings for the different examples the similarity measure from equation 2 is used.

3.2. Conceptual spaces

Using a CS representation, for each item in the test set we obtain typicality ratings for all classes (see Table 1). All examples from the test set are classified correctly.

Focussing more on the BIRD class, we can clearly observe typicality effects, as shown in Figure 1. For the BIRD class, the pheasant is the most typical example, followed by the parakeet, the ostrich and finally the penguin. This is in line with human typicality ratings as for instance reported in [15], [16] and [17], except for the fact that pheasant is rated as more typical than parakeet. Upon closer inspection it turns out that the property 'domestic', which is true for a parakeet, is somewhat rare for BIRD and therefore the parakeet is rated as less typical. We speculate that the contrast with typicality ratings from human data is due to the fact that a property 'domestic' may not commonly be very prominent for people when classifying birds.

3.3. Distributed memory model

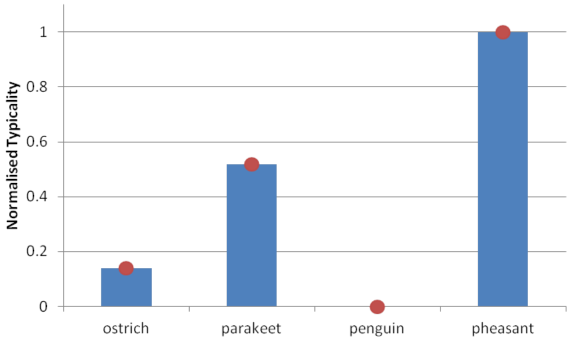


Figure 1: *Normalised typicality ratings of the CS model for the four probe trial birds for the BIRD class.*

In order to assess the effects of presentation order, we run the DAIM system twice with the same dataset; once in alphabetical order of animal name, and the second in reverse alphabetical order. Because of the inherently temporal dynamics of the system, for this case study, the properties of each animal instance are presented for 5 time-steps[4] followed by a delay of 10 time-step in which no input is presented so that all activation can decay to the resting state. For the probe trials, all of the properties for the unknown animal instances (except the name and type properties) are presented for 5 time-steps, with the activation levels on the type properties read out at the end of this period.

Table 2: Normalised results of the DAIM model for alphabetical presentation order: all correct.

PROBE	MA	BI	FI	AM	INS	INV
moth	0.01	0.08	0.00	0.22	**0.49**	0.19
newt	0.12	0.08	0.08	**0.52**	0.06	0.14
octopus	0.00	0.04	0.07	0.32	0.20	**0.34**
opossum	**0.63**	0.00	0.00	0.30	0.02	0.01
oryx	**0.78**	0.00	0.00	0.16	0.01	0.00
ostrich	0.01	**0.58**	0.00	0.30	0.08	0.01
parakeet	0.00	**0.55**	0.01	0.18	0.22	0.02
penguin	0.00	**0.50**	0.02	0.40	0.02	0.03
pheasant	0.00	**0.53**	0.01	0.25	0.15	0.02
pike	0.01	0.01	**0.42**	0.40	0.00	0.14

Table 3: Normalised results of the DAIM model for reverse-alphabetical presentation order: all but octopus are correct.

PROBE	MA	BI	FI	AM	INS	INV
moth	0.00	0.04	0.01	0.31	**0.41**	0.21
newt	0.05	0.03	0.25	**0.54**	0.00	0.13
octopus	0.00	0.04	0.13	**0.35**	0.09	0.34
opossum	**0.56**	0.00	0.02	0.36	0.01	0.02
oryx	**0.70**	0.00	0.02	0.23	0.01	0.00
ostrich	0.00	**0.52**	0.03	0.39	0.02	0.01
parakeet	0.00	**0.54**	0.04	0.25	0.08	0.03
penguin	0.00	**0.40**	0.10	0.39	0.03	0.04
pheasant	0.00	**0.48**	0.04	0.32	0.06	0.04
pike	0.01	0.02	**0.50**	0.35	0.00	0.09

The resulting typicality ratings, normalised, are shown in Table 2 and Table 3, for the two differently ordered data sets. Even though the typicality values differ for the two different data set orders, we can observe qualitatively similar results in terms of how the probe trials are classified. All but the octopus are assigned the same (correct) class and in this case of misclassification the typicality rating of the correct response is very close (0.35 and 0.34 respectively). Figure 2 shows typicality ratings for the BIRD class for the four bird examples in

[3] We chose a subset of the Zoo Data Set because to show the prototype effects the full dataset is not necessary. This is an arbitrary choice, we just choose the first 50 examples from a list in alphabetical order.

[4] A time-step resolution of 0.2s is used.

the probe trials. As can be seen, penguin and ostrich are hardly considered typical of BIRD, whereas parakeet and pheasant are rated as being much more typical. This is comparable to the result from CS, as displayed in Figure 1.

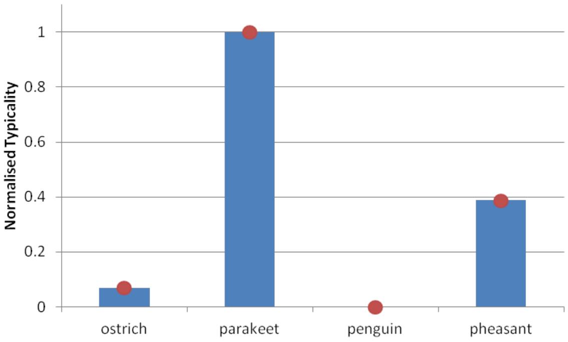


Figure 2: *Normalised mean typicality ratings of the DAIM model for the four birds presented in two probe trials. All four were classified correctly, but note that penguin and ostrich are far less typical of the bird concept than parakeet and pheasant.*

4. Discussion and conclusions

In this paper we have compared two knowledge representation frameworks for their ability to model conceptual prototypes. While Conceptual Spaces are quite suitable for this as they incorporate a notion of distance that can very easily be used as a typicality measure, it is less clear how this should happen in models that incorporate temporal aspects and are inherently distributed in nature. Whilst the temporal effects (as encountered in the order of presentation of the 50 instances to be learned) have a demonstrable effect on the behaviour of the system, the approach used in DAIM nevertheless demonstrates a robustness of ability to correctly classify the newly presented instances.

Furthermore, the DAIM results for the typicality ratings for the BIRD class exhibit prototypicality effects that are qualitatively similar to those obtained using a CS representation and to those found in human subjects. This shows the feasibility of the DAIM model, as the prototype effects are deemed important for conceptual modelling. Being inherently temporal and distributed, the use of memory models like DAIM can account for some of the more low-level functioning of the human memory, within a developmental framework (i.e. the history of interaction of the agent has a material effect on the competencies of the agent [2, 3, 13]). The comparison of the two memory conditions (normal and reverse order of data presentation) demonstrates that despite this sensitivity to interaction history (in this case order of presentation), there is nevertheless a robustness apparent in the outputs of the two trained systems. The fact that crucial aspect of modelling concepts, like prototypicality (which can more easily be modelled in a generic framework like CS) can be accounted for may be considered as an argument in favour of a distributed representation perspective; not being able to account for these aspects would constitute a shortcoming.

However, whilst the results of the DAIM system compare favourably with the standard CS implementation, it remains to be seen how such a distributed representation scheme can account for higher level concept manipulation. For example, the advantage of the CS representation scheme is the collapsing of multiple linked dimensions into a single point, that encodes a single concept or prototype. As such, it is readily available for further comparative operations with other concepts, and perhaps even higher-level processing. This property of the CS model is not so readily envisaged with the DAIM system given the entirely distributed nature of all acquired information.

Nevertheless, this study has demonstrated that some fundamental aspects of conceptual modelling can be accounted for in a distributed system that emphasises associative processes embedded within a complete cognitive system, engaged in ongoing development.

5. Acknowledgements

This work is funded by the CONCEPT project (EPSRC EP/G008353/1) and by the EU FP7 ALIZ-E project (grant 248116).

6. References

[1] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*. Cambridge, MA: the MIT Press, 2000.

[2] P. Baxter, “Foundations of a constructivist memory- based approach to cognitive robotics,” Ph.D. dissertation, University of Reading, U.K., 2010.

[3] R. Wood, P. Baxter, and T. Belpaeme, “A review of long-term memory in natural and synthetic systems,” *Adaptive Behavior*, vol. 20, no. 2, pp. 81–103, 2012.

[4] E. H. Rosch, “Natural categories,” *Cognitive Psychology*, vol. 4, no. 3, pp. 328 – 350, 1973.

[5] M. Posner and S. Keele, “On the genesis of abstract ideas,” *Journal of experimental psychology*, vol. 77, no. 3p1, p. 353, 1968.

[6] S. Reed, “Pattern recognition and categorization,” *Cognitive psychology*, vol. 3, no. 3, pp. 382–407, 1972.

[7] R. Shepard, “Toward a universal law of generalization for psychological science,” *Science*, vol. 237, no. 4820, pp. 1317–1323, 1987.

[8] R. M. Nosofsky, “Attention, similarity, and the identification-categorization relationship,” *Journal of Experimental Psychology-General*, vol. 115, no. 1, pp. 39–57, 1986.

[9] P. Baxter, R. Wood, A. Morse, and T. Belpaeme, *Memory-Centred Architectures: Perspectives on Human-level Cognitive Competencies*. AAAI Press, 2011, pp. 26–33.

[10] M. Bar, “The proactive brain: using analogies and associations to generate predictions.” *Trends in cognitive sciences*, vol. 11, no. 7, pp. 280–9, Jul 2007.

[11] R. Leech, D. Mareschal, and R. P. Cooper, “Analogy as relational priming: a developmental and computational perspective on the origins of a complex cognitive skill.” *The Behavioral and brain sciences*, vol. 31, no. 4, pp. 357–78; discussion 378–414, 2008.

[12] A. M. Burton, “Learning new faces in an interactive activation and competition model,” *Visual Cognition*, vol. 1, no. 2, pp. 313–348, 1994.

[13] P. Baxter and W. Browne, *Memory as the substrate of cognition: a developmental cognitive robotics perspective*, 2010, pp. 19–26.

[14] A. Frank and A. Asuncion, “UCI machine learning repository,” 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[15] E. Rosch, “Cognitive representations of semantic categories.” *Journal of Experimental Psychology: General*, vol. 104, no. 3, pp. 192–233, 1975.

[16] J. A. Hampton and M. M. Gardiner, “Measures of internal category structure: A correlational analysis of normative data,” *British Journal of Psychology*, vol. 74, no. 4, pp. 491–516, 1983.

[17] S. De Deyne, S. Verheyen, E. Ameel, W. Vanpaemel, M. J. Dry, W. Voorspoels, and G. Storms, “Exemplar by feature applicability matrices and other dutch normative data for semantic concepts.” *Behavior research methods*, vol. 40, no. 4, pp. 1030–48, Nov 2008.

Reachable by walking: inappropriate integration of near and far space may lead to distance errors

Beata J. Grzyb[1], *Vicente Castelló*[1] *and Angel P. del Pobil*[1,2]

[1]Robotic Intelligence Lab, Jaume I University, 12071 Castellon, Spain
[2]Interaction Science Dept., Sungkyunkwan University, Seoul, Korea
{grzyb,castellv,pobil}@icc.uji.es

Abstract

Our experimental results show that infants while learning to walk intend to reach for unreachable objects. These distance errors may result from inappropriate integration of reaching and locomotor actions, attention control and near/far visual space. Infants during their first months are fairly immobile, their attention and actions are constrained to near (reachable) space. Walking, in contrast, lures attention to distal displays and provides the information to disambiguate far space. In this paper, we make use of a reward-mediated learning to mimic the development of absolute distance perception. The results obtained with the NAO robot support further our hypothesis that the representation of near space changes after the onset of walking, which may cause the occurrence of distance errors.

Index Terms: motor development, space awareness, distance perception

1. Introduction

Infants discover and learn about their environment and about their own body through their actions. Space perception, and distance perception in particular, is action-specific [1]. Different types of actions are possible depending on the distance away of a given object. For example, if the object of interest is close to the body (in near/peripersonal space), reaching and then grasping action may be performed. If the object is beyond reach (in far/extrapersonal space), then locomotor action is needed to reach and finally grasp the object. To accomplish such a task successfully, the brain must compute the distance of the object from the agent's body correctly and activate the maps for near and far space representations appropriate to the computed distance [2]. The coding of spatial positions may not only be related to the computation of the absolute distance between the body and the stimulus, but also related to the execution of specific actions in space, such as reaching or locomotion.

Considering the problem from a develepmental perspective, not all actions are available to the infants from the beginning. Although prelocomotor infants may have depth information available, such as stereopsis, yet they lack knowledge of the absolute distance of an object in space beyond reach. Self-produced locomotion helps to calibrate visual information, resulting in more precise distance estimation of the object [3]. Such an action-based distinction between near and far space seems to be at the heart of Piaget's theory of spatial cognition in the sensorimotor period. According to his theory near space is the space calibrated by reach, and far space is that calibrated by self-produced locomotion.

Our empirical studies showed that 12-month-old infants reach significantly more than 9-month-old for unreachable objects, and that this momentary disruption in perceived reachability may be related to infants' walking ability [4]. There are a number of ways in which walking may affect decisions to reach. Our previous work focused on motivation as a possible explanation of older infants' behavior [5]. More specifically, we suggested that infants in a new upright posture fail to correctly update the boundaries of their reachable space because of their decreased ability to learn from the errors. Such blocked ability to learn from negative outcome was suggested to be tightly connected to the sense of control and to result from extremely high level of self-efficacy. Since a primary motive for walking is to reach for something, such a diminished ability to learn from the errors may help infants to fine-tune their newly acquired walking skill. An alternative explanation of distance errors was also investigated, that is that the processes responsible for integration of different visual depth cues may reorganize themselves at the onset of walking so as to incorporate depth information from self-motion-based depth cues [6]. The developmental process of distance perception for action was mimic by using a reward-mediated learning and the results showed an increase in near/far space confusions after the onset of walking.

This paper investigates in more detail the recalibration process of distance information in accordance with new motoric factors and its relation to the occurrence of distance errors in infants. We suggest that for a novice walker getting an object one wants may consist primarily of orienting the body in that directions with the hands out, and moving until you arrive. Attending to the precise distance, in the moment may just not be as important as it is for younger nonwalking infants. Herein, the walking experience is indispensible for learning the representation of far space, and for the proper integration of reaching and locomotor actions, attention control and near/far visual space. The results obtained with the NAO robot support further our hypothesis that the representation of near space changes after the onset of walking, which may contribute to the occurrence of distance errors.

2. Distance errors in infants

The main objective of our experiments was to see how infants recalibrate or scale perceptual information for action, and more specifically, how their assessment of the reachability of objects placed at different distances changes as their bodies and motor skills change. Young infants show an early distinction between what is and what is not reachable that is evident in their reaching behavior itself. At or rapidly after the onset of reaching (around 4 months), infants clearly distinguish reachable and nonreachable distances as they systematically do not to reach to far objects. The purpose of our experiments was to examine

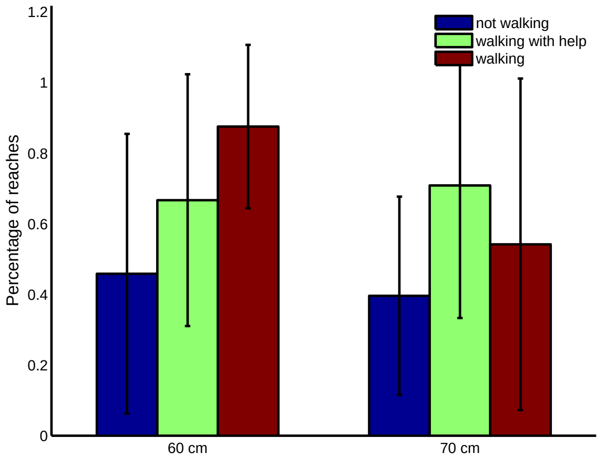


Figure 1: *Mean percentage of reaches to objects placed at far distances for 12-month olds: non-walkers, walkers with help, and independent walkers.*

the boundary between distances at which older infants, 9- and 12-month olds, do and do not reach to objects.

In total, 8 9-month-old and 8 12-month-old infants participated in our first study. Participants were seated in a modified baby car seat. The chair allowed infants to lean freely forward without a danger of falling. The balls on dowels were presented through a colorful display – like a puppet theater – that also separated the experimenter from the infants. The balls were presented at distances 30, 37, 47, 60, and 70 cm from the infant. There was no explicit reward provided to the infants after the trial for any tested distance. This helped us to avoid situations where the infants could learn to make reaching movements just to communicate their interest in obtaining a reward.

The pattern of 9-month-old reaches indicated that infants decisions to reach – in some way – take into account the relation between the body size and skill and distance of the target. For the 9-month olds, attempt reaching and successful reaching were aligned. The 12-month olds, in contrast, consistently and persistently reached to objects at distances patently unreachable showing no adjustment of their behavior with experience in the task. For the infants in the experiment, it is likely that few of the 9-month olds were walking or "cruising" upright while holding on to a support but it is highly likely that many of the 12-month olds were walking or spending time in some form of pre-walking activity in an upright posture. Thus, this developmental decline in the alignment between attempted and successful reaching distances could be related to the transition to walking.

We extended our experiment recruiting more infants with different walking abilities. A final sample constituted of 24 infants categorized into 3 equal-number groups, that is non-walkers, walkers with help, and independent walkers. Fig. 1 shows mean percentage of reaches to objects placed at far distances. As is clearly seen, walkers (with and without help) reached more for distant distances than non-walkers.

3. Reinforcement learning model

Since a reward-mediated learning have been shown to successfully mimic the development of near-optimal integration of visual and auditory cue in infants [7], a similar approach is taken here. The outline for proposed model is presented in Fig. 2. The model is composed of two – not interconnected at the moment – neural networks for reaching and for walking actions. The network architectures and neurons connections are the same in both networks, and thus will be discussed together.

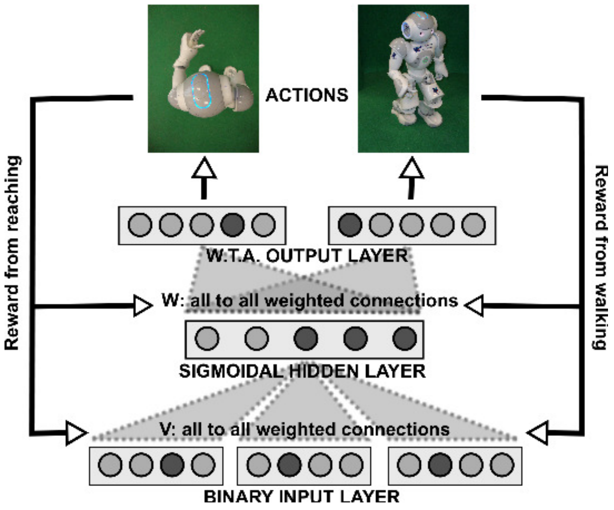


Figure 2: *General scheme of the reward-based learning model.*

A three-layer neural network is used to approximate the state-action mapping function. The input layer of the network is composed of n (in our case $n = 62$) binary neurons, that cover the range of distances from $13cm$ up to $142cm$. It is worth mentioning that the input layer may easily be extended to include more depth cues, such as stereopsis or motion parallax (as it was done in [6]). For the sake of simplicity, however, in this work we use just familiar size depth estimation. The activity of the neurons x_i is one at depth estimated by the corresponding cue, otherwise zero.

The input neurons are all-to-all connected with weights $v_{i,j}$ to j neurons in the hidden layer. A sigmoidal transfer function on the sum of the weighted inputs gives the outputs y_j of the hidden neurons:

$$y_j = \frac{1}{1 + e^{-\sum_i v_{i,j} x_i}} \tag{1}$$

The hidden neurons are fully connected to output neurons k with weights $w_{j,k}$. All weights are drawn from uniform distributions, $v_{i,j}$ between -0.1 and 0.1, and $w_{j,k}$ between -1 and 1.

Each output units represents an action. The representations of the action and their metric units are different in the both networks. While the distances in the network for reaching are represented in centimeters, the distances in the network for walking are represented in steps. In case of the network for reaching, k_r ($r = 62$) reaching actions are possible and the binning size, that is the parameter responsible for discretization of the action space is set to $1\ cm$. It is worth mentioning that the actual boundary for reachable space for the NAO robot is approximately $21cm$, and fewer neurons, in fact, are needed to encode its reachable space. Nevertheless, we set a larger number of output neurons to observe how the border of the reachable space can emerge dynamically during the learning process. In the case of the network for walking, k_w ($w = 4$) actions are possible, and the binning size is set to $1\ step$.

The activation of the output neurons z_k is given by the weighted sum of the hidden layer activity, representing an approximation of the appropriate Q-value. Based on the network's outputs, one action is chosen according to the *softmax* action selection rule [8]:

$$P_t(k) = \frac{e^{Q_t(k)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}} \tag{2}$$

where $P_t(k)$ is the probability of selecting an action k, $Q_t(k)$ is a value function for an action k, and τ is a positive parameter called *temperature* that controls the stochasticity of a decision. A high value of τ allows for more explorative behavior, whereas a low value of τ favors more exploitative behavior. We start with a high temperature parameter $\tau = \tau_0$ ($\tau_0 = 10$), so that the selection of action is only weakly influenced by the initial reward expectations. In our experiments, τ decreases exponentially with time $\tau(t) = \tau_0^{(\frac{v_\tau - t}{v_\tau})}$, where $\tau_0 = 10$ and $v_\tau = 50000$ in case of the network for reaching and $v_\tau = 5000$ in other case.

After performing the selected action $\hat{k}$ the true reward $r(\hat{k})$ is provided. The reward is maximal when $\hat{k}$ equals the true object position k_t, decaying quadratically with increasing distance within a surrounding area with radius ρ and zero otherwise ($\rho = 4$ in case of the network for reaching, and $\rho = 0$ in other case).

$$r(\hat{k}|X) = max(0, (\rho - |\hat{k} - k_t|))^2 \tag{3}$$

To minimize the error between the actual and expected reward, we make use of gradient descent method which is widely used for function approximation, and is particularly well suited for reinforcement learning.

$$v_{i,j}(t+1) = v_{i,j}(t) - \epsilon(r_{\hat{k}} - z_{\hat{k}})(-w_{j,\hat{k}})y_j(1-y_j)x_i \tag{4}$$

$$w_{j,\hat{k}}(t+1) = w_{j,\hat{k}}(t) - \epsilon(r_{\hat{k}} - z_{\hat{k}})(-y_j) \tag{5}$$

Herein, only the output weights $w_{j,k}$ connected to the winning output unit $\hat{k}$ are updated. The learning rate ϵ, decreases exponentially, according to the formula $\epsilon(t) = \frac{\epsilon_0}{ceil(\frac{t}{v_\epsilon})}$, where $\epsilon_0 = 0.05$ (for both networks), and $v_\epsilon = 50000$ in case of the network for reaching, and $v_\epsilon = 200$ in other case.

4. Experiment with the robot and results

One of the shortcomings of the reward-based methods is the large number of training examples needed for the neural network to converge. In the case of network for reaching we need approximately $t = 50000$ time steps. Such a large number of repetitions would be extremely time-consuming and unfeasible for any robotic platform. Therefore, initial weights of the neural networks are trained offline with the real data collected with the use of our robot, and then tested online on our robotic setup.

4.1. Robotic platform

Aldebaran's comercially available humanoid robot NAO with 25 DoF is used as a platform for the examined depth estimation methods. The robot is provided with two identical video cameras placed in the forehead. Their locations, however, does not allow the use of stereo vision methods for depth calculation. Within our framework, we provided the NAO robot with the reaching module, that is based on a radial basis functions (for details refer to [9]). For walking behavior, we make use of the robot built-in functions.

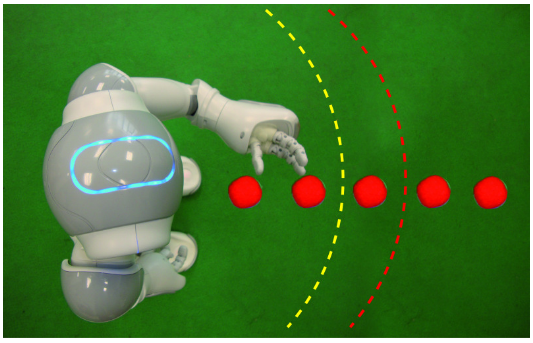


Figure 3: *Experimental setup in our study with the robot.*

4.2. Experimental setup

Our experimental setup is shown in Fig. 3. Similarly to the infant experiments, the main objective of the robot was to decide whether to reach or not for the ball. Since the NAO robot is much smaller than an average 12-month-old infant, we had to adjust the testing distances to reflect its size. Five different distances were tested, 2 close distances that easily allow the robot for reaching and grasping the objects (13 cm and 15 cm), one distance precisely at the border of reachable space (21 cm), and two distances clearly outside of the reachable space (23 cm and 26 cm). To account for the factors (others than distance) that influence the decision to reach, such as motivation or attention, we introduced some random variation on 20% of the robot's decisions.

4.3. Experiment 1: Before the onset of walking

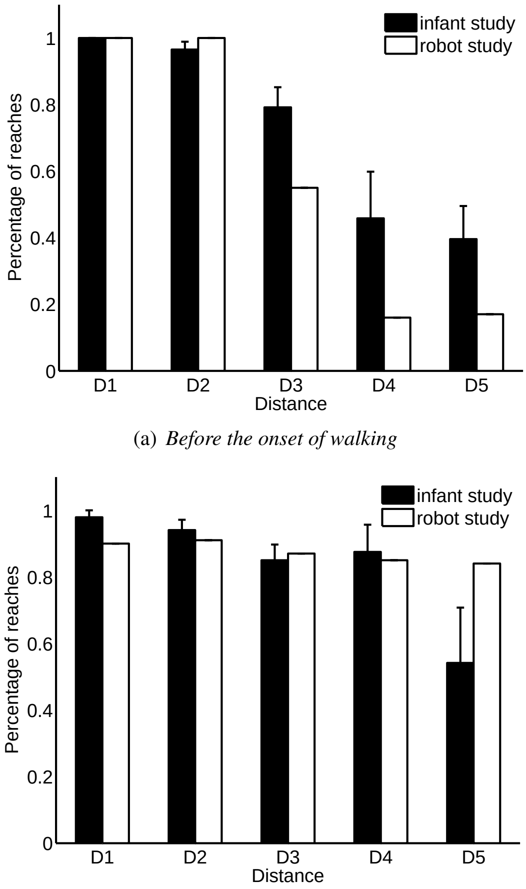
To simulate a developmental path of absolute distance perception in infants, first we train the network for reaching action, which basically constitute the near space representation. The training begins with a high temperature parameter τ, so that the selection of action is only weakly influenced by the initial reward expectations. The network is trained during 50000 time steps.

The activation of the output neurons represents a reward predictions (Q-values) which may be used to distinguish between reachable and non-reachable space. A high value of reward prediction corresponds to the near and easily reachable distance, whereas a low Q-value represent far – unreachable – distance.

The weights of the neural network are trained offline and then are employed in our robotic setup. The robot is presented with a ball at one of the five distances. Each test trial is repeated 10 times. The mean reaching attempts of the robot are shown in Fig. 4(a) along with the mean reaching attempts of 12-month-old non-walking infants. As it can be seen, the results obtained with the robot closely match the empirical results with human infants.

4.4. Experiment 2: After the onset of walking

The training of the network for walking begins with a high value of τ, so that the selection of action is only weakly influenced by the initial reward expectations. The weights of the network for reaching are also trained so that the robot can estimate the distance of an object before it gets close enough to reach for it. Simply speaking, the networks estimate the necessary number of steps for walking and the remaining distance for reaching.



(a) *Before the onset of walking*

(b) *After the onset of walking*

Figure 4: *Mean reaching attempts to various distances in the empirical experiment with infants and the robot experiment.*

The weights learned on simulation are once again employed in our robotic setup. Here, the execution of walking is blocked, similarly as in our empirical study where the infant was sitting in a chair with a seatbelt fastened. This trial test is repeated 10 times for each distance. The mean reaching attempts of the robot are shown in Fig. 4(b). It can be seen, that the representation of the near space has changed, and the distances that previously were unreachable now became reachable. It is worth mentioning, that when the execution of walking was enabled, the robot in all cases was able to walk towards and then reach and finally grasp the object successfully with the distance information provided by the networks (the error of the reaching distance was less than 1 cm).

5. Discussion

The representation of space, not only far space, but also near space changes with the onset of locomotion so that the newly emerging representation of far space can be integrated into a coherent space representation. Planning and coordination of walking and reaching behaviors are only possible when a certain level of the infant's walking proficiency has been achieved, as well as the infant has sufficient cognitive capacity to process and store the action plan. Novice walkers may reach more frequently for objects at far distances because they are not able to mentally immobilize the body's remaining degrees of freedom. The proper coordination of near and far space, and the locomotion and reaching action is required for successful executing of actions in far space. Our robot experiment suggest that had not been fastened by the seatbelt, the infants in our study would actually walk (possible with an extended hand) to reach and finally grasp the object at far distances. Further empirical experiment would be needed to verify this hypothesis.

The proposed mechanism is just one of the possible explanations of the observed distance errors in early development. Our previous works investigated the role of locomotion in observed changes in infants motivation as well as changes in the integration of various previously unattended depth-specifying cues. This paper investigated in more detail the possible mechanism for calibration of absolute distance perception that also alters the representation of near space. Nethertheless, these explanations are not mutually exclusive, and may be overlapping in underlying mechanisms.

6. Conclusions

This paper presented the phenomenon of distance errors seen in infants during the transition to walking and suggested that the calibration of absolute distance perception contributes to the appearance of these errors. The results obtained with the use of the reward-mediated approach to learning taken here provided further support for our hypothesis.

7. Acknowledgements

This research was partly supported by Ministerio de Ciencia e Innovación (FPU grant AP2007-02565, DPI2011-27846), by Generalitat Valenciana (PROMETEO/2009/052) and by Fundació Caixa Castelló-Bancaixa (P1-1B2011-54).

8. References

[1] D. R. Proffitt, "Distance perception," *Current Directions in psychological science*, vol. 15(3), pp. 131–135, 2006.

[2] A. Berti, N. Smania, and A. Allport, "Coding of far and near space in neglect patients," *Neuroimage. 2001 Jul;14(1 Pt 2):S98-102.*, vol. 14, pp. 98–102, 2001.

[3] J. J. Campos, D. I. Anderson, M. A. Barbu-Roth, E. M. Hubbard, M. J. Hertenstein, and D. Witherington, "Travel broadens the mind," *Infancy*, vol. 1, pp. 149–219, 2000.

[4] B. J. Grzyb, L. B. Smith, and A. P. del Pobil, "Reaching for the unreachable: (mis)perception of body effectivity in older infants: Developmental decline between 9 and 12 months in the link between reaching behavior and the reachability of the target," submitted.

[5] B. J. Grzyb, J. Boedecker, M. Asada, A. P. del Pobil, and L. B. Smith, "Trying anyways: how ignoring the errors may help in learning new skills," in *IEEE International Conference on Development and Learning and Epigenetic Robotics*, 2011.

[6] B. J. Grzyb, A. P. del Pobil, and L. B. Smith, "How walking influences the development of absolute distance perception," in *12th International Conference on Adaptive Behaviour*, 2012.

[7] T. H. Weisswange, C. A. Rothkopf, T. Rodemann, and J. Triesch, "Bayesian cue integration as a developmental outcome of reward mediated learning," *PLoS ONE*, vol. 6(7), pp. 1–11, 2011.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[9] M. Antonelli, B. J. Grzyb, V. Castelló, A. P. del Pobil, and L. B. Smith, "Plastic representation of the reachable space for a humanoid robot," in *12th International Conference on Adaptive Behaviour*, 2012.

A synchrony based approach for human robot interaction

Syed Khursheed Hasnain, Philippe Gaussier and Ghiles Mostafaoui

Neurocybernetic team, ETIS, ENSEA, University of Cergy-Pontoise France

syed-khursheed.hasnain@ensea.fr, gaussier@ensea.fr and ghiles.mostafaoui@ensea.fr

Abstract

As psychologists considered synchrony as an important parameter for social interaction, we hypothesize that in the case of social interaction, people focus their attention on regions of interest where the visual stimuli are synchronized with their inner dynamics. Then, we assume that a mechanism able to detect synchrony between internal dynamics of a robot and external visual stimuli can be used as a starting point for human robot interaction. Inspired by human psychological and neurobiological data, we propose a synchrony based neural network architecture capable of selecting the robot interaction partner and of locating Focus of Attention.

Index Terms: Human Robot Interaction, Synchrony, Focus of Attention, Partner Selection, Dynamical Systems.

1. Introduction

Human verbal interaction is not only speech dependent. In fact, many non-verbal behaviors such as facial expressions, pauses during discussion, hand movements etc. are also involved [1]. An important aspect of these non-verbal communications is their timing and synchrony according to the partner's behavior. Psychological Studies of dyadic interactions shows that synchrony is a necessary condition for interaction between an infant and his mother [2]. Recently, Dumas et al.[3] revealed, using hyperscanning, the emergence of inter-brain synchronization across multiple frequency bands during social interaction. Interpersonal motor coordination between people can be observed while walking along with someone [4]. Marin et al. underlined that motor resonance between robots (humanoid) and humans could optimize the social competence of human-robot interactions [5]. Qiming Shen et al. also did related experiments [6].

By the above discussion, it is clear that synchrony is an important parameter for social interaction as well as largely witnessed in natural dynamical systems. In this paper, we use immediate synchronous imitation as a communication tool. We present here a neural network architecture for socially interacting robots.

2. Materials and Methods

We used a minimal setup for our experiments as shown in figure 1. Components includes Nao robot, basic automata (1 degree of freedom), human and cameras. To avoid the frame rate limitation of the Nao's camera through the ethernet connection (limited to 10 Hz), a new camera has been added for Nao's vision. The frame rate for our experiments is 30 Hz.

To analyze synchrony, we need to investigate the dynamics of interaction between two signals. To do so, we use the Phase Locking Value (PLV) which is a practical method presented by Lachaux et al. [7]. The PLV for two signals is defined by $PLV_{n,r} = \frac{1}{T}|\sum_{t=1}^{T} exp(i(\phi_n - \phi_r))|$, where T is the number of samples and $\phi_n - \phi_r$ is the phase difference between two signals. When there is synchronization the PLV value is close to 1 otherwise the PLV value approaches to 0. Videos of our experiments can be found on: http://www.etis.ensea.fr/neurocyber/Videos/synchro/

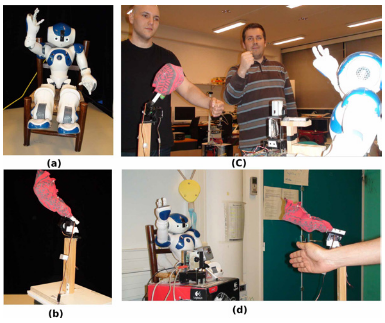


Figure 1: *Setup for our experiments. (a) Nao robot (b) Basic Automata (made in the lab) (c) and (d) Overall setup for human-robot and robot-robot interaction.*

3. Human Robot Interaction

Here, we propose a model based on dynamical interactions of two agents. Agent 1 (Nao robot) dynamically adopts or imitates the behavior of agent 2 (human / automata). Our aim is to provide to Nao limited capabilities to interact with other agents by dynamically adopting the frequency and phase of the other agents. Velocity vectors estimated by an optical flow algorithm represent the visual stimuli and inputs for our architecture.

The oscillator model is shown in figure 3(a). It is made of two neurons $N1$ and $N2$, fed by a constant signal and multiplied by the parameters $\alpha1$ and $\alpha2$ (equation 1 and 2). These two neurons inhibit each other proportionality to the parameter β.

$$N_1(n+1) = N_1(n) - \beta N_2(n) + \alpha 1 \quad (1)$$

$$N_2(n+1) = N_1(n) + \beta N_2(n) + \alpha 2 \quad (2)$$

The frequency of the oscillator depends on the parameters $\alpha1$, $\alpha2$ and β. In addition, a reservoir of oscillators (echo state network) could be used to work with a larger range of frequencies.

As shows in figure 3(a), the oscillator is connected with Nao's arm and oscillates normally at its own frequency and amplitude. Motion in the visual field of Nao is estimated by an optical flow algorithm, velocity vectors are then converted into positive and negative activities. If the perceived movements are in the upward direction, the oscillator gets the positive activity and its amplitude increases. On contrary, if the negative activity is perceived amplitude goes down. When an agent interacts with a motion frequency close to NAO's frequency, Nao's oscillator

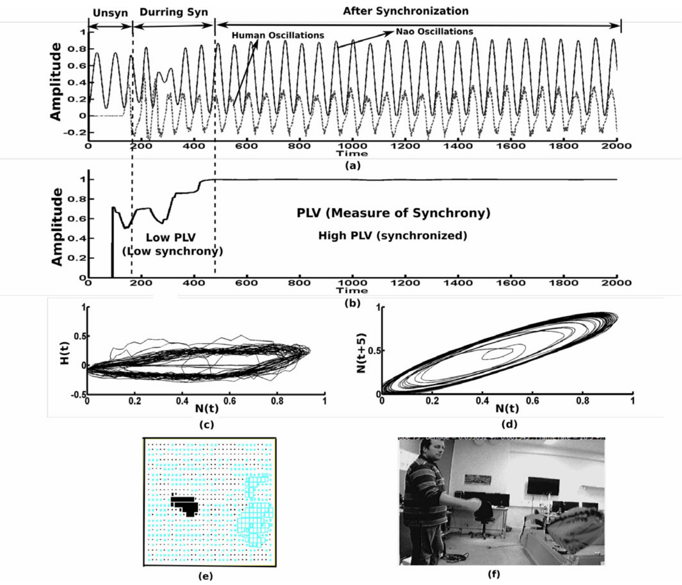


Figure 2: *(a) Shows two motion signals (human and Nao). (b) PLV measurement. (c) Lissajous curve between $N(t)$ (Nao' oscillation) and $H(t)$ (Human's movements), (d) Lissajous curve between $N(t)$ and $N(t{+}5)$. (e) Positive and Negative activities deduced from optical flow. (f) Real image seen by camera.*

can be modified within certain limits otherwise it continues to his default frequency. Mathematical equation of the oscillator can be rephrased as $N_1(n+1) = N_1(n) - \beta N_2(n) + \alpha 1 + f'$. Where f' is the induced energy.

As shown in figure 3(a) a modifiable oscillator is connected with Nao's arm. When there is no visual input it oscillates normally but if a human comes and interact with Nao by imitating him, Nao synchronizes with human by modifying his frequency and phase . Figure 2(a) shows the motion signals of both NAO's and the human arm while trying to interact by imitating games. Initially, both are unsynchronized. PLV (indicator of synchrony) has its lowest value (see figure 2(b)). As shown in Figure 2(a) and 2(b), during the interaction both Nao and human are synchronizing little by little similar to a pendulum coupling. The increasing PLV values also show the emerging synchrony. Figures 2(a) and 2(b) also clearly illustrate that, after a certain time, the agents are completely synchronized, the corresponding PLV values are at the highest possible range. Figure 2(c) shows Lissajous curve between the motion signals of Nao's ($N(t)$) and human's movements ($H(t)$). The elliptic shape of the curve indicates that both signals are almost identical. Figure 2(e) is a snapshot taken during experiment illustrating positive and negative activities in the visual field deduced using the optical flow velocity vectors. Figure 2(e) shows two moving objects in the field of view of Nao. One moves upward and induces positive activities (shown by filled black color pixels) while the other moves downward and induces negative activities (unfilled pixels). Figure 2(f) shows the real image seen by the camera.

Interesting facts are observed during experiments, some of these observations were also made by Pantaleone in his study of metronomes synchronization [11]. First, if the natural frequency of the two agents (in his case two pendulums) differs by more than a certain limit, synchronization will not occur. The range of interacting frequency (that can be synchronized with Nao) can be expanded by increasing the coupling energy f'(by scaling coupling factor) that feeds the Nao's oscillator. With low scaling factor both agents can be synchronized if their natural frequency differs by more than few percents similarly, higher scaling factor leads to higher range of frequencies. For this human/robot interaction, the default frequency of Nao's oscillator was 0.428 Hz while human's interacting frequency (measured by adding the active pixels of motion estimation) was between 0.4615 Hz to 0.476 Hz (7.8% to 11% higher than Nao's frequency) with 0.15 as a scaling factor and 15% as the corresponding Δf (difference between the natural frequencies that can be synchronization). A coupling factor of 0.3 leads to $\Delta f = 29\%$ with little variations on the amplitude, a scaling factor of 0.5 results to $\Delta f = 72\%$ but this higher coupling introduces amplitude saturation. We also observed that for the same parametric conditions, if the natural frequencies of both agents are the same no phase leg was observed but as the Δf increases to a certain limit the phase leg increases too. We experienced 0^0 to 90^0 of phase shift in our experiments.

4. Selection of Partner

We propose a neural network architecture (Figure 3(b)) that selects an interacting partner on the basis of synchrony detection among various interacting agents. Previously, the modifiable Nao's oscillator controlling the arm movement was directly connected to the visual stimuli (f'). Now, the coupling is made through an oscillator-prediction module (f''). The reason for indirect coupling is to make sure that the architecture will entertain the visual stimuli (optical flow) that is similar to its own motion (learnt by the oscillator-prediction module). Equation of modifiable oscillator can be rewritten as $N_1(n + 1) = N_1(n) - \beta N_2(n) + \alpha 1 + N + f''$. Where f'' is the energy induced by the Oscillator-prediction module.

The oscillator-prediction block (represented by y') is linked to the robot's oscillator (represented by y) with a non modifiable link while the image of visual activities (represented by X) is linked with a modifiable link. The Oscillator-prediction (y') module learns the robot's oscillation as a weighted sum of active pixels. The neuron activity in the Oscillator-prediction (y') can be computed using $X \rightarrow y'$ synapses by : $y'_i(t) = \sum_{k\epsilon X} W_{X_k - y'_i} X_k$ that corresponds to the predicted future value.The learning of $X \rightarrow y'$ synaptic weights can be computed by equation 3 and is based on NLMS (Normalized Least Mean Square) algorithm (Synaptic learning modulation η is additionally added) [8].

$$W_{X_j-y'_i}(t+dt) = W_{X_j-y'_i}(t)+\alpha\eta.\frac{y_i(t)-y'_i(t)}{\sum_{k\epsilon X} X_k(t)^2+\sigma 1}.X_j(t) \quad (3)$$

Where y' stands for the Oscillator-prediction, X for the image of visual activities and y for the NAO's arm Oscillator, α is the learning rate and $W_{X_j-y'_i}$ represents the synaptic weights from X_j to $Oscillator - prediction$ neuron i, y_i is the activity transmitted to neuron i by the oscillator, it is a target signal for the Least Mean Square (LMS) algorithm [9]. To improve the LMS convergence during the learning phase, we introduced the learning modulation η. The normalization term $\sum_{k\epsilon X} X_k(t)^2 + \sigma 1$ is specific to the NLMS and $\sigma 1$ is a small value used to avoid the divergence of the synaptic weights if the visual activities (X) values are too small.

Now we consider the complete scenario. For the selection of partner, the architecture works in two phases: learning phase and testing phase. During the learning phase, NAO oscillates according to its default frequency (no visual stimulus). NAO

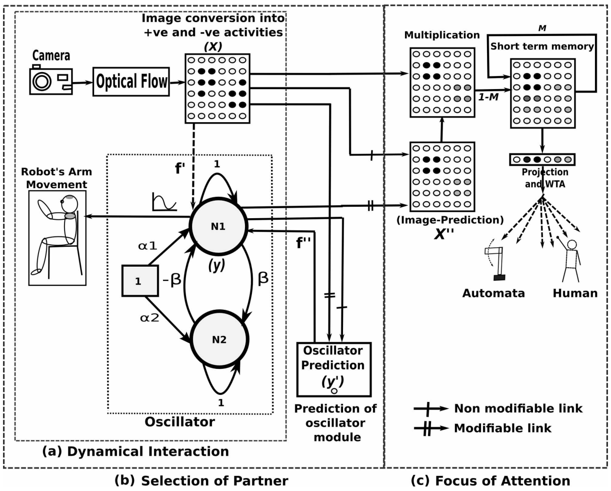


Figure 3: *(a) Dynamical Interaction model (b) Selection of Partner: select a interacting partner on the basis of synchrony detection. (c) Shows attentional mechanism architecture.*

looks at its own hand. It initiates two processes. First the oscillator prediction module which was zero due to non availability of visual stimuli starts now predicting robot's modifiable oscillator as a weighted sum of its own visual stimuli. The oscillator-prediction module learns associations between NAO's motion and the visual activities induced by NAO's arm. As a consequence, it also modifies the NAO's oscillator (as described in section 3). This process of modifying, learning and adapting continues and converge after some time. This adjustment can be assumed as a basic process by which infants gain self reflective abilities as underlined by Rochat [10]. After this phase, NAO learns to predict oscillatory movements similar to his own movement. When an agent interacts with a frequency similar to the learned one, weights (that are already learnt on modifiable links) are associated with the visual activities induced by the human movements and Nao's modifiable oscillator adopts the interacting frequency and phase. If the interacting frequency is different from the learnt one, the weights (modifiable links) could not be associated with the visual stimuli and NAO continues to move at his default frequency. Same is true for multiple agents case. Among two interactants only the agent having a similar frequency as Nao is selected. In this experiment, the coupling factor was 0.07, Nao's default frequency was 0.407 Hz, automata synchronized frequency was 0.4318 Hz (6% higher) and human synchronized frequency was 0.36 Hz (11% less). When a subject interacts with a frequency close to the learnt one, this selection of partner algorithm selects this agent as a good interacting partner and NAO modifiable oscillator synchronizes with it. Good results are obtained with this architecture, they are collectively shown in the next section.

5. Attentional Mechanism

Here, we use prediction of synchrony as a parameter to attract the attention of the robot. If two visual stimuli are presented at the same time and only one of them has the same frequency as NAO. NAO will then synchronize with the "interacting" partner corresponding to his frequency and select him as a partner (by selection of partner algorithm). However, NAO will not be able to locate the good interacting partner in its visual field, because this algorithm (partner selection) works on the perceived energy irrespective of the spatial information (agent location). To locate the correct interacting partner, the proposed FOA algorithm dynamically locate the correct interacting partner (defined by the selection of partner algorithm) using spatial predictions. Figure 3(c) shows the architecture of FOA. When a human interacts (using arm / hand), the image-prediction block (X'') learns the image of these movements as a weighted sum of Nao's synchronized frequency. This makes it possible to predict the corresponding human movements. After a short while, an other agent comes and moves with a different frequency (lower or higher than Nao), X'' which already learnt synchronized rhythmic movements predicts strongly the first synchronized agent compared to the unsynchronized one. Our algorithm modulates this predicted synchrony with the current visual stimuli and calculates the average value (acting as short term memory). As the synchronized image is well predicted its correlation values are higher than the asynchronous movements. Figure 3(c) shows that all the pixels of the memory block is projected on y axis (i.e all pixels in each column are added to find the highest correlated column). Then a Winner Takes All (WTA) selects the highest activated column. This selected column indicates the location of synchronized movement and the robot can point to the synchronized region to show the current Focus Of Attention (FOA). For this experiment the resolution of the predicted image of optical flow is $32\times$ 24 (32 columns or location), these 32 possible locations are realized in 60^0 (-30^0 to 30^0) circular angles. The learning rule of the movement-prediction (X'') module is almost the same as the oscillator-prediction module and the weights are normalized to smooth the learning processes.

5.1. Results

we examine our selection of partner algorithm along with FOA architecture (figure 3(c)) in two situations: one Automata (1-DoF) and one human (only one of them is synchronized at a time). Results show that when the Automata moves similarly to Nao's movements while human oscillates with a different frequency, Nao synchronizes with the Automata (selection of partner) and FOA mechanisms turns towards Automata. If the human adopts his frequency close to Nao, Nao aligns himself with the human and FOA moves towards human.

These results of both algorithms are shown in figure 4 by two sets of graphs. Figure 4(a) shows the onset of the experiment, where the Automata enters in the visual field of Nao from the left side (about -20^0) and imitates him. Consequently, both become synchronized using our selection of partner algorithm. Figure 4(a1) sketches the signals of Nao modifiable oscillator and Automata illustrating how they become synchronized. Figure 4(a3) shows the PLV value (measure of synchrony) of the two agents. Initially, PLV is low but as the interaction gets longer it increases to higher value. As the Automata interacts, FOA moves towards Automata as Shown in Figure 4(a4). Figure 4(a2) shows signals of Nao and human illustrating that initially there is no interaction by human from the right side of the robot. After 700 time units (23.33 seconds) human comes with a different frequency. He does not succeeded in disturbing the selection of partner (PLV remains high for Automata) and FOA remains towards the Automata.

Now, the automata is tuned to a low frequency and human is instructed to imitate NAO (figure 4(b1) and (b2)). As a result, Nao switches the synchronized region, from left (-20^0) to right side (about 27^0). The PLV related to human increases to the highest value while the Automata PLV shifts to a lowest

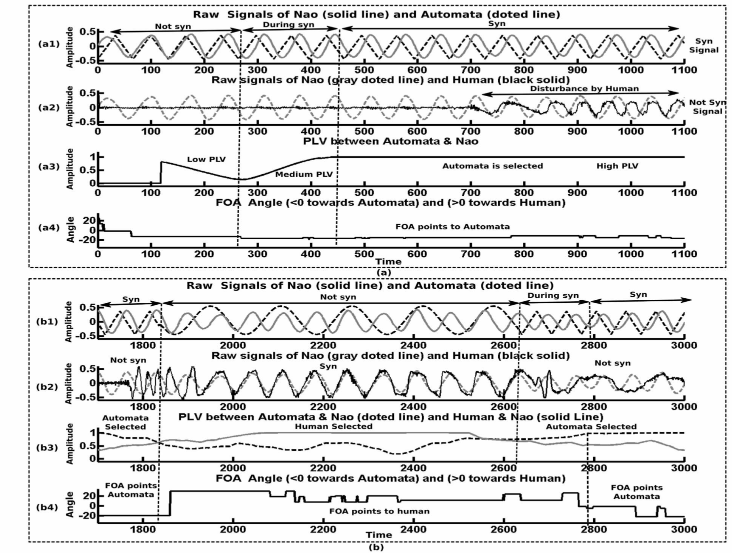


Figure 4: *Results: (a) shows start of experiment with single agent and then disturbed by the other agent. (b) Different frequency agents interact with Nao.*

one (figure 4(c3)). FOA shifts from the automata to the human (figure 4(c4)). After 2650 time units (88.3 sec), the Automata is tuned to its previous frequency again and the human is instructed to make different oscillations. Consequently, this induces a switch of the FOA and the recognized interacting partner (figure 4(b)).

6. Conclusion and discussion

We proposed a novel approach for building autonomous robots that can interact with multiple agents and select an interacting partner among several on the basis of synchrony detection. We also showed that synchrony prediction could be used as a way to establish focus of attention. From the psychological point of view, we were inspired by the unconscious communications between humans. The synchronous exchanges during social interactions are directly associated to the sensorimotor information of the two agents. These inter brain networks are ”symmetric” in low frequency band while ”asymmetric” in high frequency bands [3]. This could reflect the different processing levels of information. In our case, synchronization between two agents can be assumed as “symmetric” in low frequency band and Focus of attention can be associated with high frequency carrier.

Actually we are studying three human-robot applications for synchrony detection. The first and most obvious one is to extend the model to learn more complex interactions (complex gestures). Indeed, synchrony detection and selection of partner permit to maintain interaction with a partner moving synchronously with the robot in terms of low fundamental temporal frequency of interaction. As a result, more complex gestures (higher temporal frequencies) can be imitated and learnt autonomously by the robot while interacting with the human partner. Similarly, we aim to use our architecture for navigation tasks. A mobile robot can choose a synchronous agent to interact with and consequently learn complex navigation tasks by keeping synchrony while moving with the selected partner. Finally and in a global point of view, we question the use of synchrony detection, focus of attention and selection of partner in turn-taking games during interaction. In fact, synchrony can not only be considered as a starting point for social interaction but also as a way to re-engage the interaction with a selected partner.

7. References

[1] A. Kendon. Conducting Interaction: Patterns of behaviors in focus encounters. Cambridge University press, Cambridge, UK. 1990.

[2] J. Nadel, I. Carchon, C. Kervella, D. Marcelli, D, Reserbat. Expectancies for social contingency in 2 months old. Developmental science 2 (1999) 164-173.

[3] G. Dumas, J. Nadel, R. Soussignan, J. Martinerie, L. Garnero. Inter-Brain Synchronization during Social Interaction. PLos ONE 2010;5(8):e12166.

[4] N. R. van Ulzen, C. J. Lamoth, A. Daffertshofer, G. R. Semin and P. J. Beek. Characteristics of instructed and uninstructed interpersonal coordination while walking side-by-side. Neuroscience letters, 432, 88-93

[5] L. Marin, J. Issartel and T. Chaminade. Interpersonal motor coordination, from humanhuman to humanrobot interactions.In: Dautenhahn, Kerstin (ed.), Robots in the Wild: Exploring human-robot interaction in naturalistic environments: Special Issue of Interaction Studies 10:3 (2009). 2009. vi, 239 pp. (pp. 479504)

[6] Qiming Shen, H. Kose-Bagci, J. Saunders, K. Dautenhahn. The Impact of Participants' Beliefs on Motor Interference and Motor Coordination in HumanHumanoid Interactions. Autonomous Mental Development, IEEE Transactions on. 04/2011;

[7] J. P Lachaux, E. Rodriguez, J. Martinerie and F.J. Varela. Measuring phase synchrony in brain signals. Human Brain Mapping 8 : 194-208 (1999).

[8] J. Nagumo. A learning method for system identification. IEEE Trans. Autom. Control 12(3) (1967) 282-287 NLMS reference.

[9] B. Widrow, M. E. Hoff. Adaptive switching circuits. In IRE WESCON, pages 96-104, New York, 1960. Convention Record.

[10] P. Rochat. Ego functions of early imitiation. In A. Meltzoff & W. Prinz (Eds.), The imitative mind (pp. 85-97). Cambridge University Press, 2002.

[11] J. Pantaleone, Synchronization of metronomes. American Journal of Physics, Vol. 70, Nr. 10AAPT (2002) , p. 992–1000.

Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition
10-12 September 2012, Lausanne, Switzerland

Towards Spatial Perception: Learning to Locate Objects From Vision

Jürgen Leitner, Simon Harding, Mikhail Frank, Alexander Förster, Jürgen Schmidhuber

Dalle Molle Institute for Artificial Intelligence (IDSIA),
Scuola universitaria professionale della Svizzera Italiana (SUPSI), and
Universitá della Svizzera Italiana (USI), Lugano, Switzerland
juxi@idsia.ch

Abstract

Our humanoid robot learns to provide position estimates of objects placed on a table, even while the robot is moving its torso, head and eyes (cm range accuracy). These estimates are provided by trained artificial neural networks (ANN) and a genetic programming (GP) method, based solely on the inputs from the two cameras and the joint encoder positions. No prior camera calibration and kinematic model is used. We find that ANN and GP are both able to localise objects robustly regardless of the robot's pose and without an explicit kinematic model or camera calibration. These approaches yield an accuracy comparable to current techniques used on the *iCub*.

Index Terms: spatial understanding, object localisation, humanoid robot, neural network, genetic programming

1. Introduction

The majority of robotic systems used nowadays are still mainly performing pre-programmed automation tasks. In recent years progress has been made in enabling these robotic systems to perform more autonomous behaviours. Increasing these capabilities is necessary for future use of robots in interesting settings of daily living, such as, household tasks, grocery shopping and elderly care. An important step to perform autonomous decisions and actions is to perceive the state of the environment. Perception though is still a hard problem in robotics.

We are interested in robust approaches to visual perception, with applications to object localisation while the robot is controlling its torso, head and gaze. The localisation will be used in combination with on-line motion planning for object manipulation tasks on a real humanoid robot. In this work, we focus on a machine learning setup that provides the robot with a method to estimate the location of objects relative to itself in 3D Cartesian space. Our research platform is the *iCub* humanoid robot [1], an open robotic system, providing a 41 degree-of-freedom (DOF) upper-body, comprising two arms, a head and a torso. Its visual system is a pair of cameras mounted in the head in a human-like fashion (see Fig. 1), providing passive, binocular images.

The problem of localising objects in 3D Cartesian space given two images from cameras in different locations is widely known in the computer vision literature as 'Stereo Vision'. In the following discussion, *CSL* and *CSR* refer to the local reference frames of the left and right cameras respectively, the reference frame of the body is *CSBody*, but as it is mounted at a fixed point this is also the reference frame chosen for the environment. Therefore *CSWorld* denotes the common environmental reference frame, in which we seek to express object locations. Cameras that photograph the same scene from two different locations provide different 2D projections of the 3D scene. If the 'intrinsic parameters' that specify each camera's projection from 3D to 2D, as well as the 'fundamental matrix' that is the rigid-body transformation between *CSL* and *CSR* are known, and if there are some features of the scene that can be identified in both images, then the 3D locations of those features can be triangulated. For a thorough review of approaches based on this principle, we refer the interested reader to [2]. While traditional stereo vision approaches, based on projective geometry, have been proven effective under carefully controlled experimental circumstances, they are not ideally suited to most robotics applications. Intrinsic camera parameters and the fundamental matrix may be unknown or time varying, and this requires the frequent repetition of lengthly calibration procedures, wherein known, structured objects are viewed by the stereo vision system, and the required parameters are estimated by numerical algorithms. Assuming a solution to the standard stereo vision problem, applying it to a real physical robot to facilitate object manipulation remains a challenge. In many robotics applications, it is somewhat inconvenient to express the environment with respect to a camera. For example, from a planning and control standpoint, the most logical choice of coordinate system is *CSWorld*, the reference frame at the base of the manipulator, which does not move with respect to the environment. In order to transform coordinates from *CSL* or *CSR* to *CSWorld*, such that we can model objects and control the robot in the same frame of reference, an accurate kinematic model of the robot is required. If such a model is available, it must be carefully calibrated against the actual hardware, and even then its accuracy may be limited by un-modelled nonlinearities.

We show that localising can be learned without explicit knowledge of the camera parameters and the kinematic model.

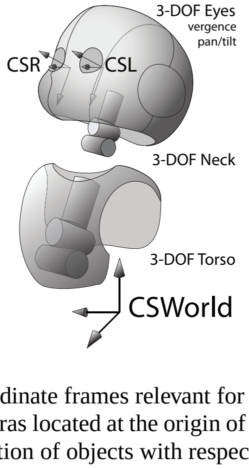


Figure 1: The coordinate frames relevant for object localisation on the *iCub*. Cameras located at the origin of *CSL*/*CSR* are used to express the position of objects with respect to the *CSWorld*.

Table 1: A typical entry from the dataset and the limits used to scale the features and locations for the neural network.

	ImageL		**ImageR**		**Neck**			**Eyes**			**Torso**			**Location**		
	X	Y	X	Y	0	1	2	3	4	5	0	1	2	X	Y	Z
Vector	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	p_0	p_1	p_2
Example	479	411	503	437	-10.0	0.0	0.0	-19.9	-19.9	0.0	-0.1	-9.9	10.1	0.42	0.27	-0.12
max	640	480	640	480	25	25	10	20	15	5	20	20	50	0.66	0.5	0.55
min	0	0	0	0	-25	-25	-10	-20	-15	0	-20	-20	0	0.00	-0.5	-0.15

2. Previous Work

Several different localisation systems have previously been developed for the *iCub*. A popular representation for (stereo) vision research is based on log-polar transformed images. This biologically inspired approach first applies a transformation to the camera images before typical stereo vision algorithms are used. The available module currently supports only a static head, ie. it puts the object position in the *CSL/R* coordinate frame. The 'Cartesian controller module' provides another basic 3D position estimation functionality [3]. This module works well on the simulated robot, however its performance on the hardware platform is weak, this is because of inaccuracies in the robot model and camera parameters. The most accurate, currently available localisation module for the *iCub* exists in the 'stereoVision' module providing centimeter accuracy. Unlike the presented log-polar approach, this module employs the entire *iCub* kinematic model, providing a position estimate in the *CSWorld* coordinate frame. The module requires the previously mentioned 'Cartesian controller' and uses tracking of features to improve the kinematic model of the camera pair by estimating a new fundamental matrix continuously. The precision of all of these approaches depends upon an accurate kinematic model of the iCub. A very accurate model, or estimation of the model, is therefore necessary.

There exists currently no module estimating the kinematics of the iCub, for other robotic systems this has been done: Gloye et al. used visual feedback to learn the model of a holonomic wheeled robot [4] and Bongard et al. used sensory feedback to learn the model of a legged robot [5], but their method uses no high-dimensional sensory information (such as images).

In robot learning, especially imitation learning, various approaches have been investigated to tackle these problems. Sauser & Billard have investigated the problem of reference frame transformations from a neuroscience perspective [6]. They were able to imitate gestures from a teacher on a Hoap-2 humanoid robot with external fixed cameras. Though promising their approach has so far not been extended to systems with non-stationary cameras.

3. Machine Learning Approach

In this paper we investigate two biologically inspired machine learning approaches: a feed-forward artificial neural network (ANN) and genetic programming (GP) approach. These techniques use supervised learning, requiring a dataset including both inputs and outputs (ground truth). More formally, the task is to estimate the position of an object $p \in \mathbb{R}^3$ in the robot's reference frame (*CSWorld*) given an input, also called feature vector, v. Here we defined $v \in \mathbb{R}^{13}$ containing the state of the robot as described by 9 joint encoder values (ie. the 9 controlled DOF) and the observed position in both camera images.

A dataset of reference points (RPs) was collected on the real hardware. A YARP [7] module registering the robot state and storing the camera images was implemented. To obtain the position of an object in the images, an object detection algorithm [8] was used to filter the raw stream from the camera. The hand-measured position of the object in 3D space was then added as the correlating output. The dataset contains 32 RPs on the table, with more than 30 robot poses per point. They lie in a region where the iCub is able to reach with its arms and were distributed in a grid with a spacing of 6 cm.

3.1. Artificial Neural Network (ANN)

An ANN, more precisely a multi-layer perceptron [9] was trained applying a standard error back-propagation [9] method on the dataset collected. The neural network approach requires a pre-processing step, in which the dataset (input vector) is scaled using the limits given in Table 1 to get values in the range $[-1, +1]$. The limits are based on to the maximum image size for the first 4 values, and the joint limits (range of motion in the stochastic controller) of the robot, for the encoder values. The output of the neural network is in the same limited range and needs to be un-scaled.

For training the network the (scaled) dataset was first randomly split into a training (80% of the data) and test set (20%). The test set allows to verify that the results obtained via learning are not over-fitting. Separate networks were trained for the estimation in the X and Y direction. Each network consists of one input layer with dimension 13, a hidden layer, and an output layer. The network uses bias terms and is fully connected. The hidden layer consists of 10 neurons, which use a sigmoidal activation function of the form $\sigma(u) = \frac{1}{1+e^{-u}}$. Finally the output layer is a single neuron representing the estimated position along one axis. The ANNs were trained using PyBrain [10] with a learning rate of 0.35 and a moment of 0.1. The errors reported are the average of 10 runs.

3.2. Genetic Programming

Genetic Programming (GP) is a search technique, most commonly used for symbolic regression and classification tasks. It is inspired by concepts from Darwinian evolution [11]. Herein we use GP to find expressions mapping the inputs to the outputs (3D coordinates). The basic algorithm works as follows: a population is initialised randomly. Each individual represents a tree, encoding a mathematical expression. The nodes encode a function, with the leaf nodes either being an available input or a constant value. For a given set of input values, the output of the expression can be found by recursing from the root node through to the terminal nodes. The individuals are then tested to calculate their 'fitness' (in our case the sum of the mean error). The lower this error, the better the individual is at performing the mapping. In the next step a new population is generated out of the old, by taking pairs of the best performing individuals and performing functions analogous to recombination and mutation. The process of test and generate is repeated until a solution is

Table 2: The mathematical functions available for the genetic programming (GP) method to select from.

add	subtract	multiply	devide
power	sqrt	exp	log
sin	sinh	cos	cosh
tan	tanh	asin	acos
atan2	min	max	abs

Table 3: Estimation accuracy on the dataset for both techniques.

	ANN	**GP**
Average Error 2D (cm)	**0.846**	**3.325**
Standard Deviation 2D (cm)	0.504	2.210
Average Error X (cm)	0.540	2.028
Standard Deviation X (cm)	0.445	1.760
Average Error Y (cm)	0.5433	2.210
Standard Deviation Y (cm)	0.4304	1.716

found or a certain number of individuals have been evaluated. A comprehensive introduction to genetic programming and its applications can be found in [12].

Herein we use a freely available software ‘Eureqa’ [13]. It produces compact, human readable expressions from datasets employing the above mentioned techniques. The input values do not have to be scaled in this approach and can remain in the original form. As with the neural network regression, data was shuffled and then split into training and validation sets. The standard settings were used. These including a population of 64 individuals, a crossover rate of 0.5 and a mutation rate of 1.5% and the mean square error of the predication was used as a fitness metric. The generated solution can contain any of mathematical functions in in Table 2.

4. Experiments and Results

To learn the ability to generalise, the techniques need a dataset representing the robot in various configurations and object locations on the table. Our first approach was to place a single object at different known positions on the table and collect data. To simplify the image processing, a red LED was used. The LED was placed at a known position in the grid to mark the reference point, while the *iCub* moved into different poses. For each pose the joint angles and camera images were collected. After collecting data for a number of poses, the LED was moved to another position and the process repeated.

For the table case the problem is simplified as we can assume a constant height (Z axis) estimation. Table 3 compares the position prediction errors of the ANN and GP techniques. It shows that the neural network is performing better during learning, which can also be seen in Fig. 2. Both approaches perform similarly when generalising to unseen data (test set). The ANN training necessitates a longer runtime, as the back-propagation algorithm repeats to update the neural network until the network performance is satisfactory. As described above, two separate networks were trained to predict the coordinates on the X and Y axes independently. This approach was chosen as it allowed for faster learning (i.e. less generations needed to yield the results) and the ability to run the learning in parallel. On average about 1700 epochs were needed per network for its prediction error to converge. After training the network produces estimates with an average accuracy of 0.8 cm, with lower separate errors on the axes (see Table 3). This makes the ANN approach the best performing approach on the dataset.

The GP method, while converging faster than the neural network, performs with a lower average accuracy of 3.3 cm. Although this performance is worse than the ANN, it is still sufficiently accurate to allow for simple reaching and grasping tasks on the *iCub*. However, there are a number of advantages to be considered. The output is in a human-readable form, which can easily and quickly be transferred and tested on the robot. Table 4 shows the evolved equations. An interesting observation is that only one of the camera images is used (features v_0 and v_1). This allows to reduce the (complete) runtime of the estimation as only one images needs to be processed with object detection algorithms before the expression can be evaluated.

During off-line training it appeared that both the ANN and GP approaches provide sufficient accuracy for object manipulation. Both approaches were implemented on the *iCub* to perform real time distance estimation of objects and to allow for further verification. The object position in the images (provided by an object detection filter from a separate iCub vision system [8]) and joint encoder values were provided to both the trained neural network and the GP evolved formulae, to allow easy comparison of the position predictions.

The validation results were obtained using locations on the

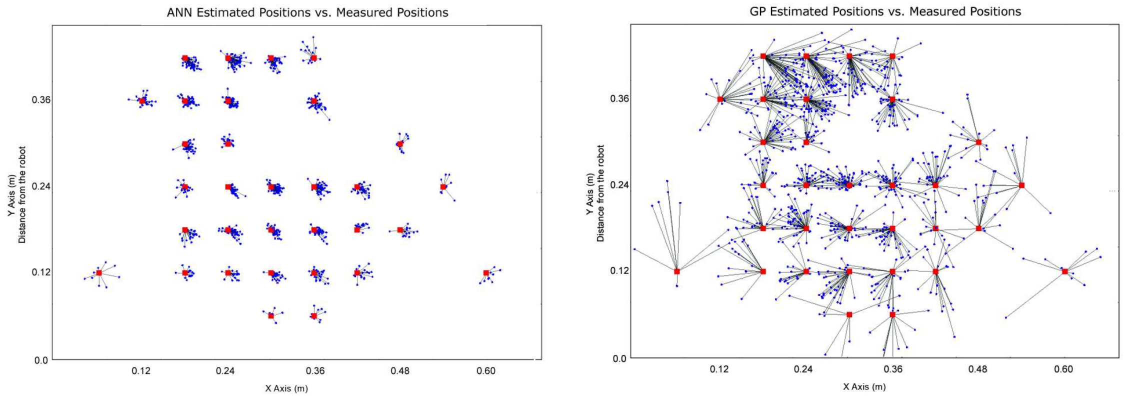


Figure 2: The estimated object position (blue dots) vs. the measured object position (red blocks) for the machine learning approaches: on the left the result obtained from artificial neural networks (ANN), on the right the results using genetic programming (GP).

Table 4: The equations generated using Genetic Programming.

$$\begin{aligned} x &= 17.81 - 0.01906\, v_1 + 0.1527\, v_4 + 0.1378\, v_7 + 0.01108\, v_{10} - 0.0296\, v_{11} - 0.1207\, v_{12} \\ y &= 1.124224045 + 0.1295920897\, v_{10} + 0.1156011386\, v_8 + 0.01695234993\, v_0 \end{aligned}$$

Table 5: The relative estimation errors (in cm) when estimating the position using fixed poses of the robot and object locations not in the training nor test set.

		ANN		**GP**		current *iCub*	
dX	dY	estX	estY	estX	estY	estX	estY
0	**+2**	0.10	**1.93**	0.51	**2.28**	0.0	**2.17**
0	**+1**	0.10	**0.78**	0.30	**0.91**	0.05	**1.0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**-1**	0.03	**1.14**	0.31	**1.35**	0.03	**1.07**
0	**-2**	0.11	**2.08**	0.61	**2.40**	0.03	**2.07**
+2	0	**1.70**	0.01	**1.93**	0.57	**2.01**	0.17
+1	0	**0.71**	0.10	**0.81**	0.34	**0.92**	0.11
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
-1	0	**0.99**	0.21	**1.12**	0.11	**1.17**	0.06
-2	0	**1.98**	0.30	**2.24**	0.34	**2.33**	0.1

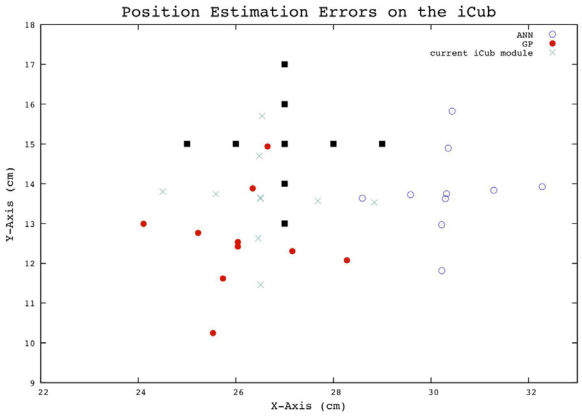


Figure 3: The relative localisation errors on the real hardware. The ground truth is shown in black, the circles represent the learning approaches, ANN (empty circle) and GP (filled). Results from the *iCub* 'stereoVision' module is plotted in green.

table and poses that were not in the original training not test set. It was found that the GP out-performed (average error of 2.7 cm) the ANN (average error of 3.5 cm) on localisation. Both techniques performed slightly worse than a fully calibrated *iCub*'s 'stereoVision' module (1.8 cm accuracy). The performance on the relative error (where the target object was moved by small increments away from a central point) was very high for both implementation with the ANN yielding better results, as can be seen by the values in Table 5 and Fig. 3. The results of the current *iCub* localisation module are added for comparison.

To test these approaches under moving conditions, we scripted the robot to move a given trajectory and recorded the position estimates for an object at a fixed location. The errors were tested for using only head/neck joints, for only using torso and for a combination of both. These all ranged in 2-4 centimetres. The faster the movement the higher the error was, this lead us to believe that it might mainly be an issue of getting the images from both cameras synchronised as much as possible.

We also performed this test with a moving test object, the error though is harder to measure when both objects are moving. In visual verification no big errors were found[1].

5. Conclusions

To estimate the positions of objects placed on a table in front of an iCub robot we compared artificial neural networks (ANN) and genetic programming (GP). No explicit robot model nor a time-consuming stereo camera calibration procedure is needed to learn. Results of locating objects on the table (2D) are sufficient for real world reaching scenarios, with the GP approach performing worse than the ANN method on the training set but generalising better when used on the hardware. The results on the first 3D dataset show that the method can be scaled to perform full 3D estimation. That said a more thorough experimental testing on the iCub will need to be conducted.

The results show that the *iCub* can learn simpler ways to perceive the location of objects than the human engineered methods. Both approaches provide simple and fast methods that can be used in real time on the robot. As the learnt models are 'light weight' they could easily be incorporated into embedded systems and other robotic platforms.

6. References

[1] N. G. Tsagarakis *et al.*, "iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research," *Advanced Robotics*, vol. 21, pp. 1151–1175, Jan. 2007.

[2] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge University Press, 2000.

[3] U. Pattacini, "Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub," Ph.D. dissertation, RBCS, Italian Institute of Technology, Genova, 2011.

[4] A. Gloye, F. Wiesel, O. Tenchio, and M. Simon, "Reinforcing the Driving Quality of Soccer Playing Robots by Anticipation," *IT - Information Technology*, vol. 47, no. 5, 2005.

[5] J. Bongard and V. Zykov, "Resilient machines through continuous self-modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, 2006.

[6] E. Sauser and A. Billard, "View sensitive cells as a neural basis for the representation of others in a self-centered frame of reference," in *Int'l. Symposium on Imitation in Animals and Artifacts*, 2005.

[7] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet Another Robot Platform," *Advanced Robotic Systems*, vol. 3, 2006.

[8] J. Leitner, S. Harding, M. Frank, A. Förster, and J. Schmidhuber, "icVision: A Modular Vision System for Cognitive Robotics Research," in *International Conference on Cognitive Systems*, 2012.

[9] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[10] T. Schaul *et al.*, "PyBrain," *Journal of Machine Learning Research*, 2010.

[11] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[12] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*. Published at `http://lulu.com`; Freely available at `http://www.gp-field-guide.org.uk`, 2008.

[13] M. Schmidt and H. Lipson, "Distilling Free-Form Natural Laws from Experimental Data," *Science*, pp. 1–5, Apr. 2009.

[1] A video showing localisation while the iCub and the object is moving can be found at `http://Juxi.net/projects/icVision/`.

Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition
10-12 September 2012, Lausanne, Switzerland

Edge and plane classification with a biomimetic iCub fingertip sensor

Uriel Martinez-Hernandez[1], *Nathan F. Lepora*[2], *Hector Barron-Gonzalez*[2], *Tony Dodd*[1], *Tony J. Prescott*[2]

[1]ACSE Department, University of Sheffield, Sheffield, UK
[2]Psychology Department, University of Sheffield, Sheffield, UK
{uriel.martinez, n.lepora, h.barron, t.j.dodd, t.j.prescott}@sheffield.ac.uk

Abstract

The exploration and interaction of humanoid robots with the environment through tactile sensing is an important task for achieving truly autonomous agents. Recently much research has been focused on the development of new technologies for tactile sensors and new methods for tactile exploration. Edge detection is one of the tasks required in robots and humanoids to explore and recognise objects. In this work we propose a method for edge and plane classification with a biomimetic iCub fingertip using a probabilistic approach. The iCub fingertip mounted on an xy-table robot is able to tap and collect the data from the surface and edge of a plastic wall. Using a maximum likelihood classifier the xy-table knows when the iCub fingertip has reached the edge of the object. The study presented here is also biologically inspired by the tactile exploration performed in animals.

Index Terms: tactile sensing, edge detection, probabilistic classification, biomimetic

1. Introduction

Nowadays most robots are equipped with haptic systems to improve their ability to interact with and learn from the environment. This is a required and important feature for humanoid robots in order to perform tasks safely. Haptics is considered as a perceptual system [1], which is mainly based on information provided by two types of sensing systems: proprioceptive sensing and extereoceptive sensing. Proprioceptive sensing detects body position, weight, and joints, whilst extereoceptive sensing refers to tactile sensing which provides physical properties of objects through physical contact [2].

Humans use the sense of touch, or tactile sensing, to explore their environment. Different predefined exploratory procedures (EPs) performed by humans with their hands and fingers allow them to recognise objects. The type of EP depends on the type of information required – for instance, sliding, pressure and contour following provide information about texture, hardness and shape respectively [3]. The way humans perform tactile sensing is considered as an active process rather than a passive one, because the movement of the hand and fingers is purposely guided to obtain more information. This process of tactile exploration is not only used by humans but is also present in the animal kingdom. Some examples of active tactile sensing are the antennae of insects and the whiskers (vibrissae) of rodents, which exhibit fascinating sensory capabilities [4]. For instance, antennae allow cockroaches to explore, detect objects and maintain their balance while climbing; rats are able to discriminate texture using their whiskers with high accuracy; seals can track fish using their whiskers, which are the most finely tuned in the animal kingdom.

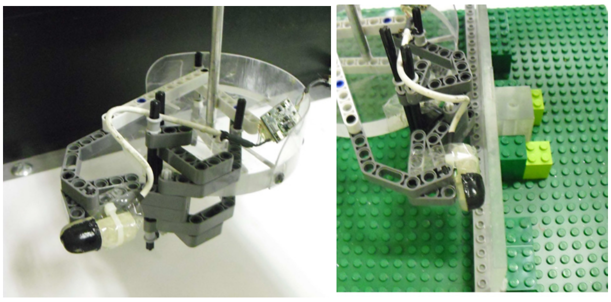


Figure 1: (a) iCub finger mounted on an *xy*-table robot to allow the movement of the finger across a plastic wall (b) to collect data from plane and edge.

Recent developments of haptic systems in robotics have allowed research in exploratory procedures inspired by the human and the animal kingdom. A three-fingered robotic hand has the capability to grasp through tactile sensing [5]. This robotic hand is able to recognise objects through the shape of the hand given by the joint angles. Another method for shape classification, using a robotic five-fingered hand, employs continuous rotational manipulation and pressure contact [6]. Texture recognition commonly is done by humans through a lateral motion or sliding EP. A robotic finger equipped with tactile sensors is able to recognise textures by sliding over materials [7], sliding either in vertical or horizontal direction. Hardness and texture recognition with a robotic hand is done with squeezing and tapping EPs [8]. This approach shows that the hardness can be measured based on the variation of joint angles while squeezing, and textures can be recognised through a tapping procedure analogous to the whisking performed by rats.

In this paper we consider tactile sensing with the iCub humanoid robot, which has recently been equipped with tactile sensors in its palm and fingers, allowing it to interact with the environment [9]. The iCub humanoid has 108 taxels (tactile elements) in total; 48 taxels in the palm and 12 taxels in each finger that respond to pressure when there is a contact. To analyse the tactile data from the iCub we employ recent advances in probabilistic perception methods inspired by tactile exploration in animals, especially rats [10, 11, 12]. In these developments, a maximum likelihood classifier (also called naive Bayes) was used for a variety of discrimination tasks, including texture, shape, position and velocity.

A key task in robotics that will be the focus of this study is to do object exploration by using edge detection through tactile sensing. Early research on edge detection has been influenced by digital image processing techniques. Low level tactile primitives have been proposed for a tactile sensor with an array of 10×16 taxels [13]. These primitives define an edge as a series of edge contacts. Another approach for edge detection uses a median filter which preserves edges and removes noise without blurring the edges [14]. In [15], image processing techniques are also applied using an edge detector which uses a threshold to remove noise. In order to obtain the location and orientation of the edge, an adaptive Hu transform is applied. Edge detection, location and orientation are obtained through the first three moments from the tactile image [16]. A new method for a low-resolution tactile sensor uses heuristics for edge detection [17]. This method has been designed for a 2×2 planar tactile sensor array.

This work presents an implementation of tapping exploratory procedure in a biomimetic robot based on the iCub fingertip applied to edge and plane detection. We apply a probabilistic method based on biologically-inspired tactile perception to perform the classification.

2. Methods

A. Tactile sensory system: iCub finger

For the experiments presented in this work, we used the tactile sensory system of the iCub humanoid. This humanoid resembles a child of 3 years old. It has 53 degrees of freedom and is equipped with digital cameras, gyroscopes, microphones and recently tactile sensors have been integrated in the forearm, palm and fingertips [18]. These tactile sensors allow the iCub humanoid to interact with the environment performing tasks safely e.g. exploring and grasping. Each fingertip has 12 contact pads called taxels, which are distributed in the base, sides and tip of the finger with a separation of about 4 mm between them. These taxels are built using a capacitive sensor technology that enables the fingers to respond to contact pressure. The measurements from the 12 taxels are sampled at 50 Hz. These measurements are digitised locally in the fingertip with a capacitive-to-digital converter (CDC) [19]. The result of the digitisation provides capacitive measurements in the range of 0 to 244, where 0 is for a maximum pressure in the fingertip and 244 is when there is no pressure. The data collected from the fingertip sensor are then passed through a drift compensation module, which converts the measurements to double precision.

B. Exploratory architecture: XY-table robot

To enable the iCub fingertip to move across a plastic wall for collecting data, it was mounted on an *xy*-table robot capable of achieving precise positioning (Figure 1). This platform enables the iCub finger to perform a tapping exploration procedure over *y* axis (vertically) whilst moving in *x* axis (horizontally). Also this platform allows the data to be collected systematically with precise movements in *x*-axis. The finger is mounted at an appropriate angle in order to have contact with most possible taxels. The *xy*-table robot moves the fingertip across appropriate regions of the stimulus to collect and store the pressure measurements from the taxels and the position for the fingertip. Figure 2 shows the two regions defined for collecting data: a 10 mm range for the plane and a 10 mm range for the edge. The *xy*-table robot performed a periodic movement across the *x*-axis of 1 mm spacing. This gave 10 taps for the plane stimulus and 10 taps for the edge stimulus.

This experiment was developed for two cases: first, moving the iCub fingertip backwards (from base to tip), and second moving in lateral motion (from left to right). For the backward case, the iCub fingertip was first placed on the plane and then placed on the edge. In the lateral motion case, the iCub fingertip was first moved over plane and then returned to its initial position and started again over the plane region. There were collected 10 sets of data for the backward case and 6 sets for the lateral motion case. The first set of plane and edge data were used for the training phase and the remaining sets for testing.

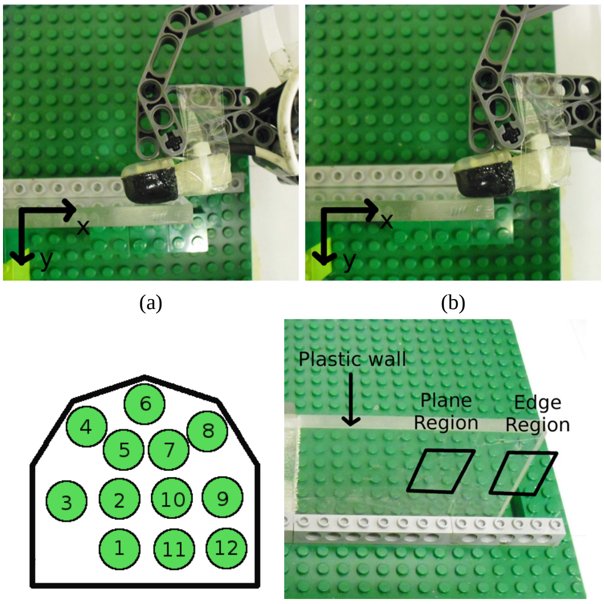


Figure 2: The iCub fingertip moved by the *xy*-table robot; *(a)* tapping in a plane region, *(b)* tapping in an edge region, *(c)* distribution of taxels in the iCub fingertip, *(d)* edge and plane regions. Note the positioning of the fingertip relative to the stimulus.

C. Probabilistic classifier

Probabilistic techniques are the state of the art for robot performance under uncertainty [20]. The measurements are considered as being caused by the world with given probabilities. This study employs previous work on probabilistic classifiers used for tactile perception based on a maximum likelihood procedure [10, 12]. Equation 1 shows the accumulated log likelihoods estimator considering the measurements to be conditionally independent

$$\log P(x_1,\ldots,x_n|C_l) = \sum_{i=1}^{n} \log P(x_i|C_l) \qquad (1)$$

The log likelihoods $\log P(x_1,\ldots,x_n|C_l)$ are accumulated over n samples of data. The single sample log likelihoods $\log P(x_i|C_l)$ are estimated from the training data using histogram methods to determine the sampling distribution [12]. The decision-making for a choice of a class C_l which can be

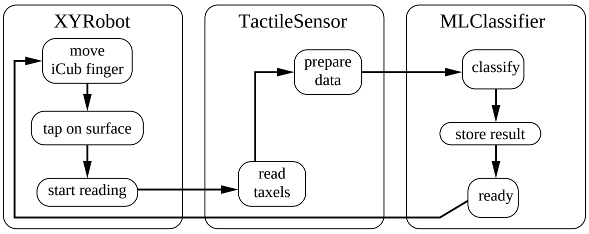


Figure 3: Interaction of modules in the experimental setup. Modules developed with C/C++ and YARP library to a straightforward implementation on the iCub humanoid.

an edge or plane is made through the maximum likelihood

$$\begin{aligned} C &= \arg\max_{C_l} P(x_1,\ldots,x_n|C_l) \\ &= \arg\max_{C_l} \left[\sum_{i=1}^{n} \log P(x_i|C_l)\right] \qquad (2) \end{aligned}$$

where $\arg\max$ provides the maximum probability for a given dataset measurement from a edge or plane contact.

In this study there are two classes; *plane* and *edge*. The classifier takes as input the measurements from the 12 taxels of the iCub finger as a time series. The maximum probability calculated by equation 2 returns the class C for the current contact. In section 3, the training and testing phases for the classification are explained.

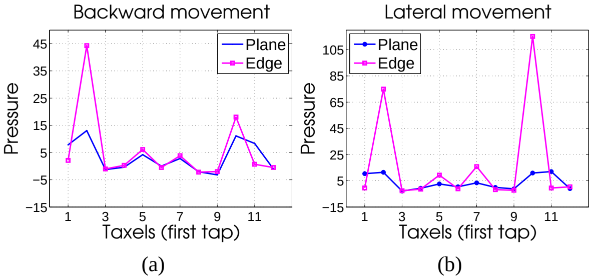
D. Experimental setup

For the experiments, several computational modules were used for control and classification: first, the *XYRobot* module for communication and control of the *xy*-table robot; second, the *TactileSensor* module for reading and preparing the measurements from taxels in the correct format to feed the classifier; and, third, the *MLClassifier* module to detect if the contact is over an edge or plane region. This experiment is based on the biomimetic iCub fingertip. However, the modules have been designed to be implemented straightforwardly on the iCub humanoid. Figure 3 shows the interaction between these modules.

3. Results

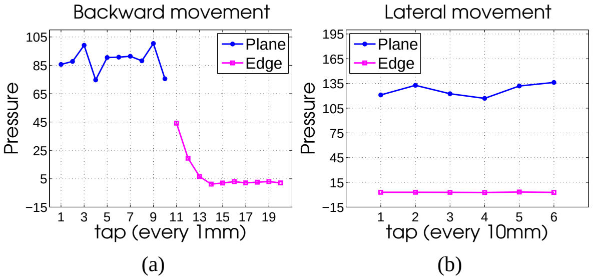
A. Training phase

The iCub finger was placed and adjusted to have enough pressure contact with the most possible taxels. The plane and edge regions were defined on a plastic wall of 6 cm×19.5 cm dimensions. A 10 mm region was defined for the plane class and a 10 mm region for the edge class. The iCub finger was configured to collect data at 50 Hz. A *drift compensation module* from the iCub repository was used to pre-process the data before classifying. For the training phase, two sets of data were collected: one for the plane and one for the edge. These datasets were taken from the first tap of the finger over the plastic wall. The datasets provided to the classifier had 12 dimensions from the number of taxels and were over 5 seconds (250 samples). Figure 4a shows the mean of pressure contact from the twelve taxels during the first tap on plane and edge regions. Similarly, the first tap for plane and edge in lateral movement is shown in Figure 4b. These datasets are the input for the classifier.



(a) (b)

Figure 4: Pressure contact of first tap over plane and edge; (a) backward movement, (b) lateral movement.



(a) (b)

Figure 5: Edge and plane detection in testing phase; (a) backward movement, (b) lateral movement.

It can be seen that the pressure is higher for taxels 2 and 10 when the finger is on the edge for both backward and lateral movement, giving a discriminator for the edge from plane.

B. Edge and plane testing phase

Two scenarios were set up for edge and plane detection validation: (1) moving the finger backward and (2) moving laterally over the edge and plane regions. For scenarios 1 and 2, there were collected 20 and 12 datasets respectively. In scenario 1, the iCub finger moved across 20 mm; first 10 mm for plane and second 10 mm for edge. The taps were taken every 1 mm. Figure 5a shows the classification across the plane and edge for scenario 1 (backwards movement). It can be observed in the *x*-axis that the position of contact by the finger and the class (edge and plane) were well predicted. The first 10 taps correspond to the plane and the second 10 taps to the edge. A clear separation of the two classes is observed (Figure 5b).

For scenario 2, the iCub fingertip firstly moved over six different positions on the edge with a range of 60 mm with a tap every 10 mm. The same procedure was followed for the plane. In this case, the iCub finger was rotated manually to the vertical position to allow lateral movements. This manual rotation may cause systematic changes in the data collection procedure followed in scenario 1. However, good results were found for both edge and plane lateral movements. Figure 5b shows the classification in lateral motion. Similar to the backward movement, there is a clear separation of the two classes. Both, plane and edge are plotted in the same *x*-axis, since the taps were from same positions for the plane and edge.

Tables 1 and 2 show the confusion matrices for backward and lateral movements respectively. Both matrices present successful classification accuracy of 100%. Interestingly, for the scenario 2, even though the vertical rotation of the finger was done manually, a 100% of classification accuracy was achieved.

Table 1: *Classification of edge and plane for scenario 1.*

Class	Edge	Plane
Edge	100%	0
Plane	0	100%

Table 2: *Classification of edge and plane for scenario 2.*

Class	Edge	Plane
Edge	100%	0
Plane	0	100%

4. Conclusions

This work has been motivated by the study of tactile sensing capabilities in humans and animals which suggest probabilistic methods for perception. A biomimetic iCub fingertip that resembles the human fingertip was used for the experiments. This finger was mounted in an *xy* positioning robot to allow systematic movements in two dimensions. Different modules were developed to implement the architecture for communication, control, data acquisition and probabilistic classification.

It was demonstrated that a tapping exploratory procedure can successfully detect object features. A plane and edge region were defined for exploration and collecting data over a plastic wall. The platform developed allowed a systematic implementation of the experiments. The classification was performed in two scenarios: (1) the iCub finger moving backwards and (2) in lateral motion. For scenario 2, the experimental setup was changed manually by orienting the iCub fingertip to point in a vertical direction. For both cases the classification showed perfect results, in that the classification accuracies were 100%. The modules used in this work for the iCub finger were designed to be implemented straightforwardly on the iCub humanoid. As such, the results presented in this work are a first step towards studying and implementing exploratory procedures performed by humans and animals on humanoid robots.

5. Acknowledgements

This work was supported by EU Framework project EFAA (ICT-270490) and the Mexican National Council of Science and Technology (CONACYT).

6. References

[1] Lederman, S. and Klatzky, R., “Haptic perception: A tutorial”, Attention, Perception, and Psychophysics, 71:1439–1459, 2009. 10.3758/APP.71.7.1439.

[2] Najarian S., Dargahi J. and Mehrizi, A. A., “Artificial tactile sensing in biomedical engineering”, McGraw-Hill biophotonics. McGraw-Hill, 2009.

[3] Lederman, S. J. and Klatzky, R. “Hand movements: A window into haptic object recognition”, Cognitive Psychology, 19(19):342–368, 1987.

[4] Prescott, T. J., Diamond, M. E. and Wing, A. M., “Active touch sensing”, Phil. Trans. R. Soc. B, 366:2989–2995, 2011.

[5] Bekiroglu, Y., Huebner, K. and Kragic, D. “Integrating grasp planning with online stability assessment using tactile sensing”, IEEE International Conference on In Robotics and Automation (ICRA), 4750–4755, 2011.

[6] Nakamoto, H., Fukui, W., Kobayashi, F., Kojima, F., Imamura, N. and Shirasawa, H., “Shape classification based on tactile information by Universal Robot Hand”, 35th Annual Conference of IEEE Industrial Electronics, 2360–2365, 2009. IECON ’09.

[7] Nawid Jamali, P Byrnes-Preston, Rudino Salleh, and Claude Sammut, “Texture recognition by tactile sensing”, araaasnau, 2009.

[8] S. Takamuku, G. Gomez, K. Hosoda, and R. Pfeifer, “Haptic discrimination of material properties by a robotic hand”, IEEE 6th International Conference on In Development and Learning, 1–6, 2007. ICDL 2007.

[9] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta, “A tactile sensor for the fingertips of the humanoid robot icub”, IEEE/RSJ International Conference on In Intelligent Robots and Systems (IROS), 2212–2217, 2010.

[10] N.F. Lepora, M. Evans, C.W. Fox, M.E. Diamond, K. Gurney, and T.J. Prescott, “Naive Bayes texture classification applied to whisker data from a moving robot”, In Proceedings of the 2010 International Joint Conference on Neural Networks, 1-8, July 2010.

[11] N. F. Lepora, C. Fox, M. Evans, B. Mitchinson, A. Motiwala, J. Sullivan, M. Pearson, J. Welsby, T. Pipe, K. Gurney, T. Prescott, “A general classifier of whisker data using stationary naive bayes: application to BIOTACT robots”, In Springer LNCS Proceedings of the 2011 Conference on Towards Autonomous Robotic Systems, 13-23, September 2011.

[12] N.F. Lepora, C.W. Fox, M. Evans, M.E. Diamond, K. Gurney, and T.J. Prescott, “Optimal decision-making in mammals: Insights from a robot study of rodent texture discrimination”, Journal of The Royal Society Interface, published online, 25 Jan, 2012.

[13] Stansfield, S., “Primitives, features, and exploratory procedures: Building a robot tactile perception system”, IEEE International Conference on Robotics and Automation. Proceedings., 7:1274–1279, 1986.

[14] Muthukrishnan, C., Smith, D., Myers, D., Rebman, J. and Koivo, A., “Edge detection in tactile images”, IEEE International Conference on Robotics and Automation. Proceedings., 4:1500–1505, 1987.

[15] A. D. Berger and P. K. Khoslar, “Edge dectection for tactile sensing”, In Proc. N A S A conf. on space telerobotics, 163–172, 1989.

[16] Chen Ning, Rink Ray and Zhang, H, “Efficient edge detection from tactile data”, IEEE Proceedings, Comput. Soc. Press, 386–391, 1995.

[17] Phung Tri, Ihn Yong, Koo Ja, and Choi Hyouk, “Edge identification of a small object through a low-resolution tactile sensor array”, International Journal of Precision Engineering and Manufacturing, 11(2):247–254, 2010.

[18] Schmitz, A., Maggiali, M., Natale, L. and Metta, G., “Touch sensors for humanoid hands”, IEEE, RO-MAN, 692–697, September 2010,

[19] Schmitz, A., Pattacini, U., Nori, F., Natale, L., Metta, G. and Sandini, G., “Design, realization and sensorization of the dexterous iCub hand”, 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 186–191, December 2010,

[20] S. Thrun, W. Burgard, and D. Fox, “Probabilistic Robotics”, MIT Press, Cambridge, MA, 2005.

Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition
10-12 September 2012, Lausanne, Switzerland

Towards Modular Spatio-temporal Perception for Task-adapting Robots

Zoltan-Csaba Marton, Florian Seidel, Michael Beetz

Intelligent Autonomous Systems, Technische Universität München, Munich, Germany
{marton, seidelf, beetz}@cs.tum.edu

Abstract

In perception systems for object recognition, the advantage of multiple modalities, of combining approaches, and several views is emphasized, as they improve accuracy. However, there are great variances in the implementation, suggesting that there is no consensus yet on how to approach this problem. Nonetheless, we can identify some common features of the methods and propose a flexible system where existing and future approaches can be tested, compared and combined. We present a modular system in which perception routines can be easily added, and define the logic of making them work together based on the lessons learned from different experiments.

Index Terms: robotics, multi-cue vision, machine learning

1. Motivation

Autonomous agents working in human environments have a huge variety of objects to deal with, and some of them present special problems (texture-less, transparent, etc). There are multiple approaches that have been shown to be able to segment, detect, categorize and/or classify some of the objects such robots might encounter. There are, however, inherent limitations in these approaches, and there is no robust and large-scale solutions yet [1]. As each perception method captures only some aspect of the objects, the situation is similar to the old story about the six blind men trying to describe an elephant based on a single touch. Clearly, a correct combination of different sensor modalities, segmentations, features, classifiers would improve results. Additionally, in [2] it is argued, that a cognitive agent needs to be embodied to gather experiences, and presents different paradigms on how to approach the learning and grounding of new information. Similar ideas are discussed in [3] as well, where the task and environment adaptation of a robot improves its capability to perceive objects.

In this work, we focus on taking advantage of exploration capabilities of the robot, and the fact that a high-level task specification is typically available. Therefore we propose a system that can take advantage of the fact that only some objects are probable to be at different places in the close surrounding of the robot, and of these ones, only some are relevant for the task at hand. Different perception methods can then be activated (or tuned) and combined, in order to improve detection rates. Additionally, multiple observations over time can be incorporated to obtain higher quality results. In short, the main propositions of this paper for a perception system are as follows:

- common input-output defined for segmentation and detection methods,
- support for consecutive or parallel methods to correct or support each-other in a probabilistic framework,
- enable the specialization of each method to a subset of objects and to group objects into categories,
- incorporating information from multiple views to disambiguate complex cases.

To support our approach, we evaluate these principles, and:

- show the advantage of combining different cues,
- evaluate different ensemble methods and discuss their benefits and drawbacks,
- describe our practical solutions to increase the robustness and accuracy of perception systems,
- present proof-of-concept experiments.

After an overview of the related work, we will outline the basis for our proposal in Section 3, followed by the details of a multi-cue perception system in Section 4. As it will be detailed, the modular combination of task-adapting perception routines performing spatio-temporal integration of multiple modalities holds great potential for the development of robust computer vision. We argue that a deep integration of various levels of a cognitive architecture will be required, and present the connections we found to be most important in our experiments.

2. Related Work

Inspired by earlier work based on developmental psychology, object categorization using multiple modalities is explored in [4] and the advantage of accumulating information over time is shown. While sychological findings do suggest that a single sensory modality is often not enough, they leave out the most descriptive modality, vision, and focus on proprioceptive and auditory feedback [5].

In [6] the authors validate the use of different visual modalities, showing that color-based cues are more important for instance recognition, while geometric ones are better suited for categorization, and that their combination improves on both.

Existing perception systems that use multiple modalities for object detection, either combine these in a fixed feature [7] or use them in a fixed framework [8]. Selecting only relevant features for a specific task was explored in [9], but in a sequential framework with a fixed order of features/modalities. Here we propose a parallel architecture with a heuristic decision on which perception primitives should be applied to identify different objects, and with an incremental merging and verification step to provide the final result.

Systems that use validation of the detections through geometric consistency relied on a single modality so far [10, 11], however the advantage of scoring or voting for different solutions is an important lesson that we incorporated in the system.

There is growing evidence that human vision combines top-down (concept driven) and bottom-up (data driven) approaches [12], thus extending classification systems with context information is a natural way of increasing performance. In our framework we use the prior distribution over the possible objects/locations (and the known object models) as the context.

Most of the perception systems rely either on color/black&white camera (e.g. SIFT [13]) or 3D (e.g. VFH [14])information, although image processing techniques can be applied on different image sources as well (e.g. thermal cameras). There are approaches that combine geometry and color descriptors, but properly balancing these two is not straightforward as discussed in [15].

3. Experimental Support

Some aspects of the proposed solution have been verified already in different experiments. The following subsections give details on the evaluation of some of the natural ways how object perception results can be improved.

3.1. Multi-Modal Perception

Combining multiple sensor modalities to improve detection can be done in general either by combining multiple features in a single classification pipeline or by separate processing pipelines for each modality, whose results are combined. The former approach is pursued in [6], where a combination of visual and depth cues is used. We explored the latter approach in [9], highlighting the limitations of the different sensors, and exploiting that not all features need to be check if there is a subset of them that uniquely describes the object. In this work, we present our approach for combining the results of different modules by forming ensembles, as discussed in the following subsection.

3.2. Ensemble Learning

We evaluated the accuracy of standard off-the-shelf classifiers, trained on image-based and 3D features, and ensembles of such classifiers on the large RGB-D object dataset from [16]. As visual features we used SURF [17] and Opponent SURF with a Bag of Features approach and VFH [14] and GRSD-, the geometric part of VOSCH [15], as geometric features. Our interest lies predominantly in simple, non-parametric ensemble methods, since such simple ensembles can endow the proposed system with the required modularity. Hence, the goal was to investigate how simple, non-parametric ensemble methods compare to more sophisticated but parametric classifiers and ensembles.

As a benchmark we considered the the task of identifying the category to which an object belongs for all of the 300 objects and 51 categories in the dataset. All the objects are seen during training time and half of the over 200,000 scans in the dataset are used for training the classifiers. A quarter is used for evaluation and another quarter as hold-out data to estimate the accuracy of the ensemble methods.

We tested SVM and boosted decision trees (AdaBoost) as classifiers, and different voting based methods and stacking for merging their results, as these were suggested in the literature as promising approaches [18, 19]. Classifiers trained on the concatenation of all the features are used as a baseline to which the performance of the ensembles is compared (see Table 1).

Table 1: *Error rates for single features and the concatenation of all features – linear SVM (top) and AdaBoost (bottom).*

VFH	GRSD-	SURF	O.SURF	All
0.133	0.409	0.281	0.301	0.031
0.149	0.435	0.360	0.361	0.0991

After trying several weightings for the voting methods, the best one was found to be the weighting with the estimated class accuracy. For stacking we used real AdaBoost as level-0 classifiers and real AdaBoost, LogitBoost and Gentle Boost as well as linear SVM and SVM with Radial Basis Function kernel as the level-1 classifier, and found Gentle Boost to give best results.

Table 2: *Voting vs tacking for ensembles of single features*

Base classifier	Voting error rate	Level-1 classifier	Stacking error rate
SVM-Linear	0.100	GentleB	0.054

As shown in Table 2, combining different cues is advantageous, and (while more tests could be made) it seems that concatenating the features outperforms the simple weighted voting and the learning based stacking approach. Nonetheless, both approaches improve the result over those of the best single feature, and we found that using pairwise concatenations of features the error rates can be lowered even below that of the classifiers trained on the concatenation of all the features (see Table 3). This suggests that with the right feature combinations and weighting factors, voting could be a great solution as well – increasing the modularity of the perception system.

Table 3: *Stacking with classifiers of single + double features.*

L-SVM	RBF-SVM	AdaB	LogitB	GentleB
0.031	0.065	0.02	0.019	0.019

3.3. Spatio-Temporal Integration

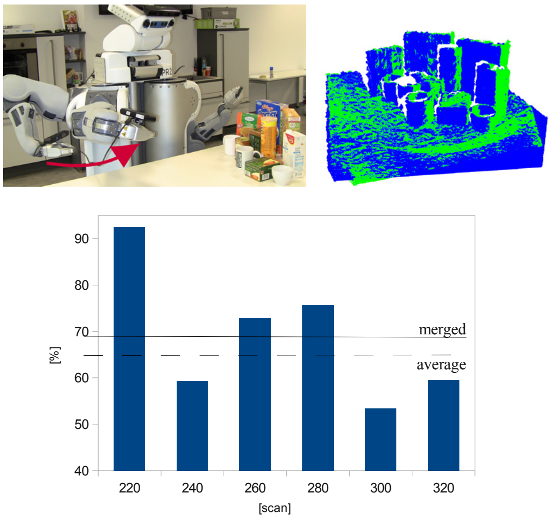


Figure 1: *As the camera is moved (left), multiple frames can be captured that cover different parts of the objects in the scene (right), increasing the overall classification accuracy (bottom).*

We showed the advantage of merging the object detection results from multiple 3D scans in a voting framework previously in [11]. There, we also proposed the use of multiple segmentations of the same input to be merged in the same manner. This approach is employed for image segmentation as well, showing improved results. In Figure 1 the same idea is applied

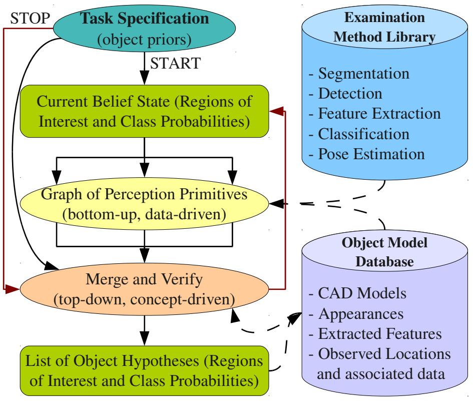

STOP
Task Specification
(object priors)
START
Current Belief State (Regions of Interest and Class Probabilities)
Graph of Perception Primitives
(bottom-up, data-driven)
Merge and Verify
(top-down, concept-driven)
List of Object Hypotheses (Regions of Interest and Class Probabilities)
Examination Method Library
- Segmentation
- Detection
- Feature Extraction
- Classification
- Pose Estimation
Object Model Database
- CAD Models
- Appearances
- Extracted Features
- Observed Locations and associated data

Figure 2: *Basic setup of the proposed system for iterative refinement of object hypotheses by multiple methods according to a task specification.*

for detecting 6 commonly occurring shapes of household objects (a grouping of data from [16] based on [9]), where 3D volumes obtain votes from scans taken from different angles.

3.4. Class Hierarchies

In order to match the perception capabilities of humans, the authors in [20] advocate that searching for predefined templates is not enough, and that recognition of new exemplars of known categories have to be facilitated. On this premise, in [9] we used geometric cues for categorization and visual cues for instance classification. We also reported on the improvement in accuracy of over 10% when the geometric categorization is allowed to work with "internal" categories. This suggests that an unsupervised classification level followed by a mapping to human-defined labels, as in [11], enables the classifiers to tune themselves to the specific feature space used.

4. Proposed Solution

Our proposed solution to integrate the approaches supported in Section 3 for a modular, multi-cue perception system that takes advantage of the robot's exploration capabilities is exemplified in Figure 2 (as a generalized extension of the system presented in [9]). It builds on the lessons learned form previous experiments by the authors and others, and on many discussions from people involved not only in perception, but also high-level planning, manipulation and knowledge engineering for example.

4.1. Regions of Interest and Poses

Most related systems from literature are either doing segmentation or classification (or both at once), but in both cases a region of space is observed, and hypotheses are given about what objects it, or parts of it, contains. A segmentation routine for example breaks large regions up into smaller ones, and assigns to each of them a non-informative prior, i.e. from the point of view of the method each segment can contain anything. Subsequent processing (classification) steps then refine these possibilities. Template matching methods for example do both steps at once, by returning possible (scored) positions in which an object could be in the scene.

Therefore, we propose the use of volumes of space, or *regions of interest* (ROIs) as the basic input *and* output data for object perception methods. These can be for example the hulls of clusters for 3D data, or the estimated volumes of image pixels. These, and the associate probabilities of given objects being contained in it, are received and updated by the perception methods, and can be used to merge information coming from different sensors, and different views.

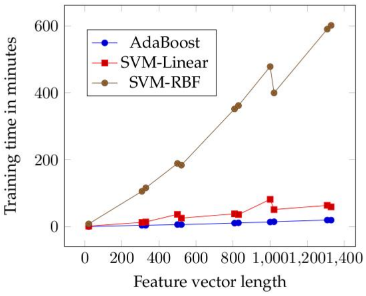


Figure 3: *Feature vector length vs. Training Time (20 classes)*

4.2. Task-adapting Perception Primitives

Initially, the system would start off with the complete workspace of the robot as its region of interest, with the different priors for the occurrence of the possible objects assigned to it. The list of these objects and their prior probability can then be considered by the different methods, and when summarizing their results.

We call all the segmentation, detection, fitting and classification methods *perception primitives*, as they are the different modules the system is build of. They can use different sensors, extract various features, apply different recognition methods, and have only to respect the aforementioned input and output in order to be part of the "ensemble". Classification methods such as those based on nearest neighbors, are easy to be re-trained, and this allows simple integration of new data as well. However, with the addition of more and more classes, the accuracy can drop – this can be avoided by taking advantage of the known *task specification* (i.e. list of possible objects and list of sought objects). Similarly, the accuracy of other classifiers deteriorates with the increase in the number of classes (see Tables 4,5 and those in Section 3.2), something that can be alleviated by task and environment specialization.

Table 4: *Error rate for single/concatenated features, 20 classes.*

Classifier	VFH	GRSD-	SURF	O.SURF	All
SVM-linear	0.081	0.270	0.154	0.163	0.0188
SVM-RBF	0.050	0.202	0.098	0.105	0.0172
AdaBoost	0.087	0.293	0.254	0.202	0.0544

Table 5: *Stacking with classifiers trained on single + double features, for 20 classes.*

L-SVM	RBF-SVM	AdaB	LogitB	GentleB
0.013	0.013	0.014	0.014	0.012

Not all classes are as fast to be re-trained as nearest neighbors though, as shown in Figure 3, but methods like locally weighted logistic regression [21] could be used to avoid retraining by adjusting only the weighting of the examples.

4.3. Combining Cues

Since each perception primitive refines the result of its input, the ROIs are trimmed down (if necessary) and the class probabilities accumulated. In the *merging* step all the results can be united through ROI unification, and a decision can be made by an ensemble method. Subsequent sensor readings can be accumulated using the same procedure, and the object hypotheses and their poses can then be verified if they match the data as in [11]. Accumulating or comparing object poses is more complicated, but a scored list of poses can also be maintained, and checked against the accumulated data in the given volume. Another approach to obtain 6DOF pose directly from camera images is to project CAD models of objects to the image and search for good edge responses. However it is unclear how these methods scale to handling very large number of objects.

5. Initial Demos and Discussion

Proof-of-concept demonstrations of the presented approach were made during the 2nd BRICS Research Camp "From 3D sensing to 3D models" (`www.best-of-robotics.org/2nd\_researchcamp/MainPage`) and the public 2011 CoTeSys Fall Workshop (`www.youtube.com/watch?v=DTaeWITW1kI`). Here, a region of interest is provided by the task executive using the known environment model along with the list of possible objects to be detected. The different detection, classification and model fitting methods decide for each request to activate or not based on the objects to be detected and if they have models for those. Different 2D and 3D methods are chained in order to produce the final result, i.e. list of object locations and locations/poses/models. The task executive then interprets the results, decides on the next action to be taken (which could be repeating a failed procedure) and triggers a new task if necessary (e.g. using the bounds of the detected cutter board to detect the slice of bread). As the number of integrated perception primitives increases, and multiple routines for performing the same task become available, the theoretical consideration presented in this paper become more and more important. The presented approach for taking advantage of multiple sources of information by a modular system proved to be useful and scalable in our initial experiments implemented in ROS (`ros.org`). We are confident that the robustness suggested by the supporting experiments will be of great use for integrating our perception system into a cognitive architecture with similar design philosophy, e.g. based on [22].

6. Acknowledgements

This work was supported by the DFG excellence initiative research cluster CoTeSys (www.cotesys.org). The author would like to thank Team2 from BRICS2: Leif Jentoft, Andre Ückermann, Kyle Strabala and Moritz Tenorth; and the IAS team, especially Nico Blodow, David Gossow, Ingo Kresse, Alexis Maldonado, and Dejan Pangercic, as well as Asako Kanezaki and Ferenc Balint-Benczedi.

7. References

[1] D. Kragic and M. Vincze, "Vision for robotics," *Foundations and Trends in Robotics*, vol. 1, no. 1, pp. 1–78, 2009.

[2] D. Vernon, "Cognitive vision: The case for embodied perception," in *Image and Vision Computing*. Elsevier, 2005.

[3] I. Horswill, "Integrating vision and natural language without central models," in *In Proceedings of the AAAI Fall Symposium on Embodied Language and Action*, 1995.

[4] J. Sinapov and A. Stoytchev, "Object category recognition by a humanoid robot using behavior-grounded relational learning," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[5] D. Lynott and L. Connell, "Modality exclusivity norms for 423 object properties," *Behavior Research Methods*, vol. 41, no. 2, pp. 558–564, 2009.

[6] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information," in *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[7] H. Nakayama, T. Harada, and Y. Kuniyoshi, "AI Goggles: Real-time Description and Retrieval in the Real World with Online Learning," *Computer and Robot Vision, Canadian Conference*, vol. 0, pp. 184–191, 2009.

[8] J. F. Georg Arbeiter and A. Verl, "3D Perception and Modeling for Manipulation on Care-O-bot 3," 2010.

[9] Z. C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D-3D Categorization and Classification for Multimodal Perception Systems," *The International Journal of Robotics Research*, 2011.

[10] M. M. Torres, A. C. Romea, and S. Srinivasa, "MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System," in *Proceedings of ICRA 2010*, May 2010.

[11] O. M. Mozos, Z. C. Marton, and M. Beetz, "Furniture Models Learned from the WWW – Using Web Catalogs to Locate and Categorize Unknown Furniture Pieces in 3D Laser Scans," *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 22–32, 2011.

[12] J. P. Frisby and J. V. Stone, *Seeing: The Computational Approach to Biological Vision*. MIT Press, 2010, ch. 8: Seeing Objects, pp. 178–179.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[15] A. Kanezaki, Z.-C. Marton, D. Pangercic, T. Harada, Y. Kuniyoshi, and M. Beetz, "Voxelized Shape and Color Histograms for RGB-D," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, San Francisco, CA, USA, September, 25–30 2011.

[16] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[17] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In CGIV*, 2008, pp. 404–417.

[18] L. Lam and C. Y. Suen, "Optimal combinations of pattern classifiers," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 945–954, 1995.

[19] J. Sill, G. Takács, L. Mackey, and D. Lin, "Feature-weighted linear stacking," *CoRR*, vol. abs/0911.0460, 2009.

[20] S. Dickinson, "The evolution of object categorization and the challenge of image abstraction," in *Object Categorization: Computer and Human Vision Perspectives*, S. Dickinson, A. Leonardis, B. Schiele, and M. Tarr, Eds. Cambridge University Press, 2009, pp. pp 1–37.

[21] K. Deng, "Omega: On-line memory-based general purpose system classifier," PhD Thesis, Carnegie Mellon University, Tech. Rep., 1997.

[22] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, N. Schlaefer, and C. A. Welty, "Building Watson: An Overview of the DeepQA Project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.

Whom Will an Intrinsically Motivated Robot Learner Choose to Imitate from?

Sao Mai Nguyen, Pierre-Yves Oudeyer

[1]Flowers Team, INRIA and ENSTA ParisTech, France.

nguyensmai at gmail.com, pierre-yves.oudeyer at inria.fr

Abstract

This paper studies an interactive learning system that couples internally guided learning and social interaction in the case it can interact with several teachers. Socially Guided Intrinsic Motivation with Interactive learning at the Meta level (**SGIM-IM**) is an algorithm for robot learning of motor skills in high-dimensional, continuous and non-preset environments, with two levels of active learning: **SGIM-IM** actively decides at a meta-level when and to whom to ask for help; and an active choice of goals in autonomous exploration. We illustrate through an air hockey game that SGIM-IM efficiently chooses the best strategy.

Index Terms: Active Learning, Intrinsic Motivation, Social Learning, Programming by Demonstration, Imitation.

1. Introduction

In initial work to address multi-task learning, we proposed the Socially Guided Intrinsic Motivation by Demonstration (SGIM-D) algorithm which merges socially guided exploration as defined in[1, 2, 3, 4] and intrinsic motivation [5, 6, 7, 8] based on SAGG-RIAC algorithm [9], to reach goals in a continuous task space, in the case of a complex, high-dimensional and continuous environment [10]. Nevertheless, the SGIM-D learner uses demonstrations given by a teacher at regular frequency. It is passive with respect to the social interaction and the teacher, and does not optimise the timing of the interactions with the teacher, not to mention that it did not consider the everyday situation where it has several human teachers around him, to whom it can ask for help. Some works have considered the choice among the different teachers that are available to be observed [11] where some of them might not even be cooperative [12], but have then overlooked autonomous exploration. Our new **SGIM-IM** (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level) learner is able to choose between active autonomous and social learning strategies, and in the case of social learning, whom to imitate from.

2. General Framework

2.1. Formalisation

In this subsection, we describe the learning problem that we consider. Csibra's theory of human action finds that infants connect actions to both their antecedents and their consequents [13, 14]. Thus, every episode would be described as [context][action][effect].

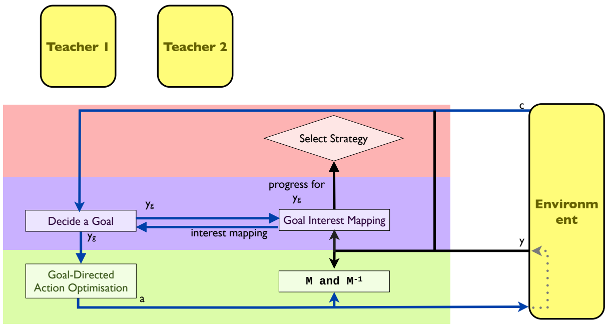


Figure 1: Data Flow under the Intrinsic Motivation strategy

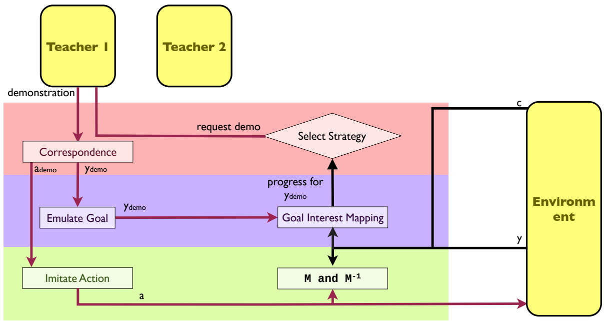


Figure 2: Data Flow under the Social Learning strategy with teacher 1

Let us describe different aspects of the states of a robotic system and its environment by both a state/context space C, and an effect/task space Y (an effect/task can be considered as restricted to the changes caused by the agent's actions). For given contexts $c \in C$, actions $act \in ACT$ allow a transition towards new states $y \in Y$ (fig. 1 and 2). We define the actions act as parameterised dynamic motor primitives, i.e. temporally extended macro-actions controlled by parameters a in the action parameters space A. The association (c, a, y) corresponds to a learning exemplar that will be memorised. Our agent learns a policy through an inverse model $\mathbf{M^{-1}} : (c, y) \mapsto a$ by building local mappings of $\mathbf{M} : (c, a) \mapsto y$, so that from a context c and for any achievable effect y, the robot can produce y with an action a. We can also describe the learning in terms of tasks, and consider y as a desired task or *goal* which the system reaches through the *means* a in a given *context* c. In the following, both descriptions will be used interchangeably.

2.2. SGIM-IM Overview

SGIM-IM learns by episodes during which it chooses actively its learning strategy between intrinsically motivated exploration or social interaction with each of the existing teachers.

In an episode under the intrinsic motivation strategy (fig. 1), it actively generates a goal $y_g \in Y$ of maximal competence improvement, then explores which actions a can achieve the goal y_g in context c, following the SAGG-RIAC algorithm [9]. The exploration of the action space gives a local forward model $\mathbf{M} : (c, a) \mapsto y$ and inverse model $\mathbf{M^{-1}} : (c, y) \mapsto a$, that it can use later on to reach other goals. The SGIM-IM learner explores preferentially goals where it makes progress the fastest. It tries different actions to approach the self-determined goal, re-using and optimising the action repertoire of its past au-

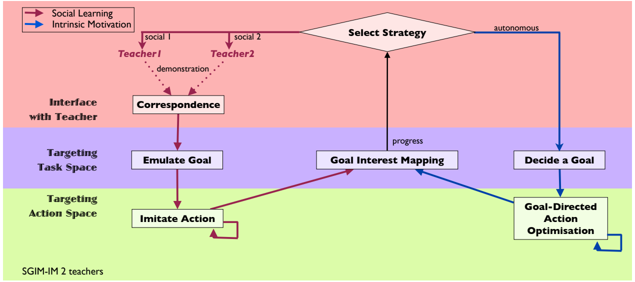


Figure 3: Time flow chart of SGIM-IM

Algorithm 2.1 SGIM-IM

```
Initialization: $\mathcal{R} \leftarrow$ singleton $C \times Y$
Initialization: $flagInteraction \leftarrow false$
Initialization: $Memo \leftarrow$ empty episodic memory
Initialization: $\Delta_0, ...\Delta_i, ...$ : progress values made by strategy i among: autonomous exploration or social learning with either teacher
loop
    $strategy \leftarrow$ Select Strategy$(pref_S, pref_A)$
    if Social Learning Strategy then
        demo $\leftarrow$ ask & perceive demo to the selected teacher i
        $(c_{demo}, a_{demo}, y_{demo}) \leftarrow$ Correspondence (demo)
        Emulate Goal: $y_g \leftarrow y_{demo}$
        $\gamma_s \leftarrow$ Competence for $y_g$
        Memo $\leftarrow$ Imitate Action$(a_{demo}, c)$
        $\gamma \leftarrow$ Competence for $y_g$
        Add $\gamma - \gamma_s$ to stack $\Delta_i$
    else
        Intrinsic Motivation Strategy
        Measure current context $c$
        $y_g \leftarrow$ Decide a goal$(c, \mathcal{R})$
        $\gamma_s \leftarrow$ Competence for $y_g$
        repeat
            Memo $\leftarrow$ Goal-Directed Action Optimisation$(c, y_g)$
        until Terminate reaching of $y_g$
        $\gamma \leftarrow$ Competence for $y_g$
        Add $\gamma - \gamma_s$ to stack $\Delta_0$
    end if
    $\mathcal{R} \leftarrow$ Update Goal Interest Mapping$(\mathcal{R}, Memo, c, y_g)$
end loop
```

tonomous exploration or the actions suggested by the teacher's demonstrations of the social learning strategy. The episode ends after a fixed duration.

In an episode under the social learning strategy with teacher i (fig. 2), our SGIM-IM learner observes the demonstration $[c_{demo}, act_{demo}, y_{demo}]$, memorise this effect y_{demo} as a possible goal, and imitates the demonstrated action act_{demo} for a fixed duration.

The SGIM-IM learner actively decides on a meta level which strategy to choose according to the recent learning progress enabled by each strategy. If it has recently made the most progress in the intrinsic motivation strategy, it prefers exploring autonomously. Conversely, if the demonstrations does not enable him to make progresses higher than by autonomous learning (limited teacher, or inappropriate teacher) it would prefer autonomous exploration.

3. SGIM-IM Architecture

3.1. A Hierarchical Architecture

SGIM-IM (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level) is an algorithm that merges interactive learning as social interaction, with the SAGG-RIAC algorithm of intrinsic motivation [9], to learn local inverse and forward models in complex, redundant, high-dimensional and continuous spaces and with several teachers. Its architecture (alg. 2.1) is separated into three layers (fig. 3) :

Algorithm 3.2 [strategy] = SelectStrategy(Δ_S, Δ_A)

```
input: $\Delta_0, ...\Delta_i, ...$ : progress values made by strategy i among: autonomous exploration or social learning with either teacher
output: $flagInter$ : chosen strategy
parameter: $nbMin$ : duration of the initiation phase
parameter: $ns$ : window frame for monitoring progress
parameter: $cost_i$ : cost of each strategy
Initiation phase
if Social Learning and Intrinsic Motivation Regimes have not been chosen each $nbMin$ times yet then
    $p_i \leftarrow 0.5$
else
    Permanent phase
    for all strategies do
        $w_i \leftarrow$ average(last $ns$ elements of $\Delta_i$)
    end for
    $p_i \leftarrow min(0.9, max(0.1, \frac{cost_i \times w_i}{\sum cost_j \times w_j}))$
end if
strategy $\leftarrow$ i with probability $p_i$
return strategy
```

- An interface with the teacher, which manages the interaction with the teacher. It decides in an active manner whether to request a demonstration and to whom (*Select Strategy*) and interpreting his actions or his intent and translates the demonstrations into the robot's representation system (*Correspondence*, which is an important issue [15] but will not be addressed in this study.
- The *Task Space Exploration*, a level of active learning which drives the exploration of the task space. With the autonomous learning strategy, it sets goals y_g depending on the interest level of previous goals, by stochastically choosing the ones for which its empirical evaluation of learning progress is maximal (*Decide a Goal*). With the social learning strategy, it retrieves from the teacher information about demonstrated effects y_{demo} (*Emulate a Goal*). Then, it maps $C \times Y$ in terms of interest level (*Goal Interest Mapping*).
- The *Action Space Exploration*, a lower level of learning that explores the action space A to build an action repertoire and local models. With the social learning strategy, it imitates the demonstrated actions act_{demo}, by repeating it with small variations (*Imitate an Action*). During self-exploration, the *Goal-Directed Action Optimisation* function attempts to reach the goals y_g set by the *Task Space Exploration* level, 1) by building local models during exploration that can be re-used for later goals and 2) by optimising actions to reach y_g. Then, the *Action Space Exploration* returns the measure of competence at reaching y_{demo} or y_g.

The active choice of learning strategy will be described hereafter. For the other parts of the architecture, which are common to SGIM-D, please refer to [10] for more details.

3.2. Select Strategy

Based on the recent progress made by each of them, a meta level chooses the best strategy among autonomous exploration and social learning with each of the teachers. For each episode,

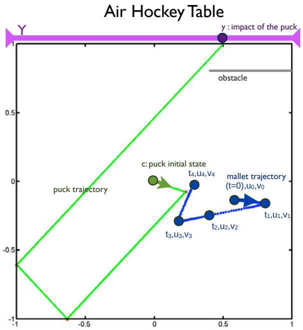


Figure 4: Air Hockey Table

the learner measures its progress as the difference of competence at the beginning and the end of the exploration for the self-determined or the emulated goal, and adds this progress value to stacks Δ_i, where i is the current strategy ($i = 0$ for autonomous exploration, $i = 1$ for social learning with teacher 1, $i = 2$ with teacher 2,...). The preference for each strategy is computed as the average on a window frame of the last ns progress values of Δ_i. Setting the value of n_s does not depend on the complexity of the tasks but more on the size of the task space. It needs to allow appropriate sampling of Y by each method. In our simulations, $n_s = 20$. Besides, to limit the reliance on the teacher and take into account the availability of each teacher, we penalise the preference for social learning with a $cost_i$ factor ($cost_i = 100$). For the autonomous exploration strategy, $cost_0 = 1$. The strategies are selected stochastically with a probability proportional to their preference (alg 3.2).

We applied our hierarchical **SGIM-IM** algorithm with 2 layers of active learning to an illustration experiment.

4. AirHockey Experiment

4.1. Description of the Experimental Setup

Our first experimental setup is a simulated square air hockey table that contains an obstacle (fig. 4). Starting with a fixed position and velocity (1 single context), the puck moves in straight line without friction. The effect is the position of the impact when the puck collides with the top border of the table. Y is thus the top border of the table, mapped into the $[-1, 1]$ segment, which highlights the subregion hidden by the obstacle as difficult to reach.

We control our mallet with a parameterised trajectory determined by 5 key positions $u_0, u_1, u_2, u_3, u_4 \in [-1, 1]^2$ (10 scalar parameters) at times $t_0 = 0 < t_1 < t_2 < t_3 < t_4$ (4 parameters). The trajectory in time is generated by Gaussian distance weighting:

$$u(\mathbf{t}) = \sum_{i=0}^{5} \frac{w_i(\mathbf{t}) u_i}{\sum_{j=0}^{5} w_j(\mathbf{t})} \text{ with } w_i(\mathbf{t}) = e^{\sigma * |\mathbf{t} - \mathbf{t_i}|^2}, \sigma > 0 \quad (1)$$

Therefore, A is of dimension 14 and Y of dimension 1. The learner maps which trajectory of the mallet a induces a collision with the top border at position y. This is an inverse model of a highly redundant mapping, which is all the more interesting than the obstacle introduces discontinuities in the model.

4.2. Experimental Protocol

We detail in this subsection the experiments we carry with our air hockey table, how we processed to evaluate SGIM-IM and provide our learner with demonstrations.

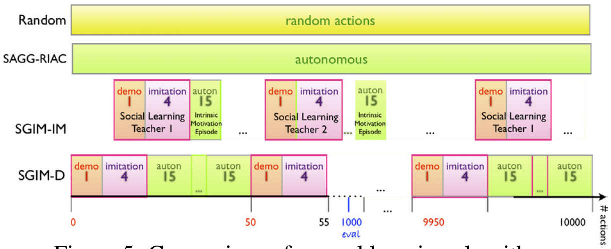


Figure 5: Comparison of several learning algorithms

4.2.1. *Comparison of Learning Algorithms*

To assess the efficiency of SGIM-IM, we decide to compare the performance of several learning algorithms (fig. 5):

- Random exploration: throughout the experiment, the robot picks actions randomly in the action space A.
- SAGG-RIAC: throughout the experiment, the robot explores autonomously driven by intrinsic motivation. It ignores any demonstration by the teacher.
- SGIM-IM: interactive learning where the robot learns by actively choosing between social learning strategy or intrinsic motivation strategy, and who to imitate from.
- SGIM-D: the robot's behaviour is a mixture between Imitation learning and SAGG-RIAC. When the robot sees a new demonstration, it imitates the action for a short while. Then, it resumes its autonomous exploration, until it sees a new demonstration by the teacher. Demonstrations occur every T actions of the robot.

For each experiment in our air hockey setup, we let the robot perform 10000 actions in total, and evaluate its performance every 1000 actions. For the air hockey experiment, we set the parameters of SGIM-IM to: $cost = 10$ and $ns = 20$, and those of SGIM-D to T= 50.

4.2.2. *Demonstrations and Evaluation*

We simulate 2 teachers by using the learning exemplars taken from Random and SAGG-RIAC learners. For teacher 1, we choose demonstrations in $[-1, 0.5]$ with each $y_{demo_k} \in [-1 + k \times 0.01, -1 + (k+1) \times 0.01]$. For teacher 2, we likewise choose demonstrations in $[0.5, 1]$, that manage to place the puck behind the obstacle.

We assess the algorithms by measuring how close they can reach a benchmark set distributed over $Y = [-1, 1]$ and placed every 0.05, with the mean error at reaching the benchmark points.

4.3. Results

Fig.6 plots the mean distance error of the attempts to hit the border at the benchmark points, with respect to the number of actions performed by the mallet. It shows that while Random exploration and SAGG-RIAC error decrease, SGIM-IM performs significantly better, and faster. It almost divided by a factor of 10 the final error value compared to SAGG-RIAC. Its error rate is always smaller than for the other algorithms since the very beginning. SGIM-IM has taken advantage of the demonstrations very fast to be able to hit the puck and place it on the top border, instead of making random movements which would have little probability of hitting the puck, let alone placing it at the benchmark position. Its performance is comparable with SGIM-D. This shows that its active choice of strategy was able to choose social learning over autonomous learning to bootstrap its progress, and to vary its choice of teacher to overcome the limited subspaces of the demonstrations of each teacher .

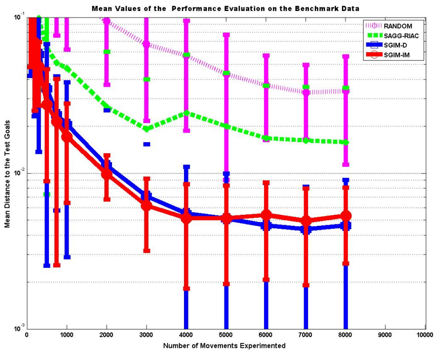


Figure 6: Evaluation of the performance of the robot with respect to the number of actions performed, under different learning algorithms. We plotted the mean distance to the benchmark set with its variance errorbar.

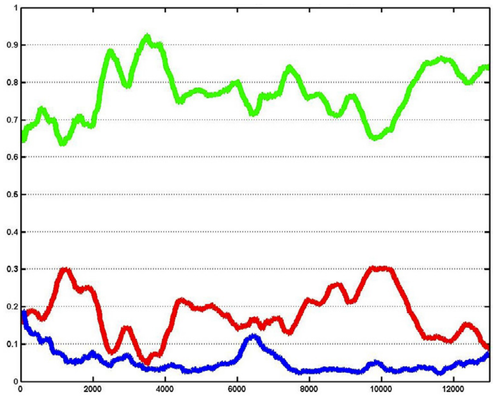


Figure 7: Percentage of times each strategy is chosen by SGIM-IM with respect to the number of actions performed: intrinsic motivation (green), social learning with teacher 1 (red) and with teacher 2 (blue).

4.4. Active Choice of Strategy

As for the strategy adopted, fig.7 shows that total number of demonstration requests increases in the very beginning, as they are most useful in the beginning, as each indicate to the learner which kind of actions can make the mallet hit the puck whereas random movements have low probability of hitting the puck. After this first phase, the learner prefers autonomous learning because of the cost of asking for teachers' help. It then increases again in the second half of the experiment when the progress made by autonomous exploration decreases. Demonstrations then help the learner improve in precision.

Furthermore, requests were asked more often to the teacher 1 as he covers a more important subspace of Y. This indicates that the learner could detect the difference in teaching capabilities of the 2 teachers. We would also like to point out that the number of demonstrations of teacher 2 made a small peak around 6500 when the error curve stops decreasing, showing that his help was most useful once the learner has managed to reach the subspace of Y that is easy to reach before getting interested in the subspace behind the obstacle. This slight peak effect can be more visible with more experiments to improve our statistics, and by complementary figures to analyse this effect.

5. Conclusion

We presented **SGIM-IM** (Socially Guided Intrinsic Motivation with Interactive learning at the Meta level), an algorithm that combines intrinsically motivated exploration and interactive learning with demonstrations. With an architecture organised into 3 layers, it actively decides when and to whom to ask for demonstrations. Through an air hockey experimental setup, we showed that SGIM-IM efficiently learns inverse models in high-dimensional, continuous and non-preset environment despite high redundancy. Its active choice of strategy was able to choose social learning over autonomous learning to bootstrap its progress, and to choose the right teacher to overcome the limited subspaces of the demonstrations of each teacher. It thus offers a framework for more flexible interaction between an autonomous learner and its users.

6. Acknowledgements

This research was partially funded by ERC Grant EXPLORERS 240007 and ANR MACSi.

7. References

[1] A. Whiten, "Primate culture and social learning," *Cognitive Science*, vol. 24, no. 3, pp. 477–508, 2000.

[2] M. Tomasello and M. Carpenter, "Shared intentionality," *Developmental Science*, vol. 10, no. 1, pp. 121–125, 2007.

[3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Handbook of Robotics*. MIT Press, 2007, no. 59, ch. Robot Programming by Demonstration.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009.

[5] E. Deci and R. M. Ryan, *Intrinsic Motivation and self-determination in human behavior*. New York: Plenum Press, 1985.

[6] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 599-600, 2001.

[7] M. Lopes and P.-Y. Oudeyer, "Active learning and intrinsically motivated exploration in robots: Advances and challenges (guest editorial)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 65–69, 2010.

[8] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[9] A. Baranes and P.-Y. Oudeyer, "Intrinsically motivated goal exploration for active motor learning in robots: A case study," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 1766 –1773.

[10] S. M. Nguyen, A. Baranes, and P.-Y. Oudeyer, "Bootstrapping intrinsically motivated learning with human demonstrations," in *Proceedings of the IEEE International Conference on Development and Learning*, Frankfurt, Germany, 2011.

[11] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitation," *J. Artificial Intelligence Research*, vol. 19, no. 569-629, 2003.

[12] A. Shon, D. Verma, and R. Rao, "Active imitation learning," in *American Association for Artificial Intelligence*, vol. 22, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 756.

[13] G. Csibra, "Teleological and referential understanding of action in infancy," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, p. 447, 2003.

[14] G. Csibra and G. Gergely, "Obsessed with goals: Functions and mechanisms of teleological interpretation of actions in humans," *Acta Psychologica*, vol. 124, no. 1, pp. 60 – 78, 2007.

[15] C. L. Nehaniv and K. Dautenhahn, *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge: Cambridge Univ. Press, March 2007.

Emergence of Leadership in a Group of Autonomous Robots

Francesco Pugliese[1], *Alberto Acerbi*[3] , *Orazio Miglino*[1], *Davide Marocco*[2]

[1] Natural and Artificial Cognition Laboratory, University of Naples, Italy
[2] Centre for Robotics and Neural Systems, Plymouth University, United Kingdom
[3] Centre for the study of cultural evolution, Stockholm University, Sweden

francesco.pugliese@unina.it, alberto.acerbi@gmail.com, orazio.miglino@unina.it, davide.marocco@plymouth.ac.uk

Abstract

For modern biology and ethology, the reason for the emergence of leaders-followers patterns in groups of living organisms, is the need of social coordination. In this paper we attempt to examine factors contributing to the emergence of leadership, trying to understand the relation between leader role and behavioral capabilities. In order to achieve this goal, we use a simulation technique where a group of foraging robots has to choose between two identical food zones. Thus, robots must coordinate in some way in order to select the same food zone and collectively gathering food. Behavioral and quantitative analysis indicate that a form of leadership emerges and the emergence of leadership relates with high level of fitness. Moreover, we show that more skilled individuals in a group tend to assume a leadership role, in agreement with literature.

Index Terms: Leadership, Evolutionary Robotics, Flocking

1. Introduction

Many animal species, including humans, live in groups **[1]**. The advantages of living in groups have been extensively explored in ethology and robotics, and they are related to (a) protection from predators **[2]**, (b) feeding efficiency **[3]**, (c) competition with other groups of conspecifics **[4]**, and (d) possibility of information sharing **[5]**.

Living in groups poses a fundamental problem of social coordination. Researches in robotics and agent-based modeling have usually focused on homogeneous groups, in which social coordination emerges from local rules followed in the same way by all individuals **[6,7]**,

Anyway, in real animals, especially in mammals and virtually always in primates, whenever there are groups, there is a leadership / followership pattern emergence. Evolutionary biologists use the term leadership for behaviors that influence the type, timing and duration of group activity **[8]** and generally argue that the reason for the emergence of leadership / followership patterns is the need to coordinate **[9]**. It has been proposed, for example, **[10]** that personality differences may represent a prerequisite for the emergence of leadership, where individuals more prone to environmental exploration tend to assume the role of leaders.

Game-theoretical analysis has shown how, in some situations, leadership is almost inevitable. In a simple two-player "coordination game", a pair of individuals has to reach two simple goals: one individual must stay near the partner for protection, and the other needs to seek resources such as food patches and waterholes. In this situation, any trait (physical or behavioral) that increases the likelihood of one individual moving first will make him more likely to emerge as the leader, and the other player is left with no option but to follow. Furthermore, if this trait difference between players is stable (i.e. if the first individual is always hungry first) then leadership-follower patterns will be stable over time **[11]**. Therefore, it seems that individuals are more likely to emerge as leaders if they have a particular physiological or behavioral trait increasing their propensity to act first to solve coordination problems. In the human case, social environment may have increased the conditions for the emergence of sophisticated leadership / followership patterns **[12]**.

Biological and ethological experiments are often difficult to be performed in laboratory and it is hard to get experimental evidences of theories about leadership and grouping emergence using experimental animal or human subjects.

In this work we propose an alternative and original approach based on a collective robotics experimental setup. We have simulated a group of artificially evolving robots (kepera-like) situated in an environment where they must coordinate in order to forage. We conceived the evolutionary process in order to maintain genetic (and behavioral) diversity whithin the groups, so to reproduce conditions which can lead to leadership emergence according to the literature previously provided. We tried to answer to some fundamental questions, such as: Does leadership arise in a group of genetically *heterogeneous* robots? Who is the leader? What are characteristics and skills of leaders?

The originality of our approach comes from the implementation of an evolutionary robotics model in order to study decision making in a social group. These kind of simulations are been performed, in the past, but with a merely agent-based approach (e.g. **[13]**).

2. Experimental Setup

2.1. The Task

A group of four simulated robots live in an environment consisting of a 110cm x 110cm squared arena surrounded by walls. When a robot bumps against environment's wall or against another robot, it bounces back in the neighborhood of the contact point, with a new random direction.

The food source is located in two target areas placed in a fixed position of the environment. Each robot is made of a circular chassis with a radius of 11 cm and it is equipped with two motors controlling the movements of two wheels, respectively (Fig. 1). Moreover, the robot is geared with two sensors which "smell" the relative position of the food zone in respect to the position of the robot body, as illustrated in Fig.2. According to the position of the food zone with respect to a fixed sector of the robot, smell sensors will be activated with a two digits binary code.

Each robot is characterized by a color of the body: green, blue, light blue and yellow and it is equipped with a linear retina system in order to see the position and the color of the other group members. The linear retina is made of five RGB photoreceptors that manage a portion of the robot field of view.

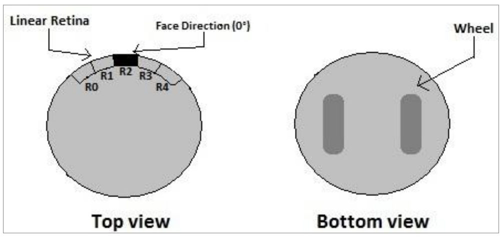


Figure 1: Schematisation of top and bottom view of the robot chassis.

The field of view (FOV) of each robot is 90 degrees wide, and represents the extent of the observable world that the robot can see at any moment. The FOV ranges from -45 degrees to +45 degrees with respect to the direction of movement (0°). In this way, each photoreceptor manages a 18 degree wide portion of the FOV, the first one is associated to a range of [-45°,-27°] respect to the face direction, the second one to [-27°,-9°], and so on.

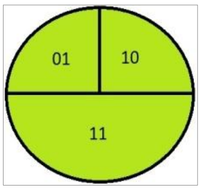


Figure 2: Representation the activation patterns of the robot smell system.

Each photoreceptor consists of 3 colour sensitive components, respectively Red, Green, and Blue. When an object (such as a robot) is located in the front of a photoreceptor, within its vision angle, the sensor is activated to the corresponding RGB value for that object. The maximum vision distance of receptors is the environment size. The setup is illustrated in Fig. 3.

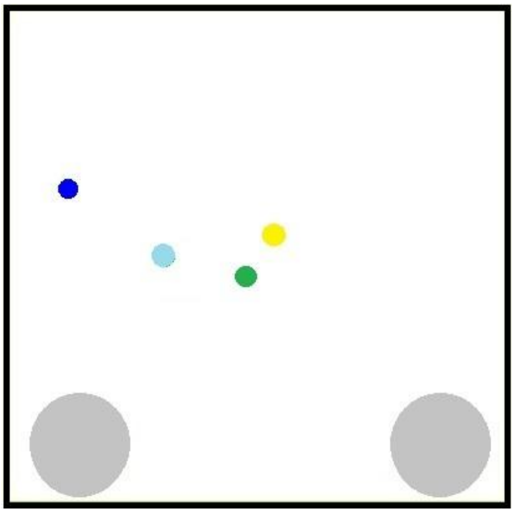


Figure 3: The environment and the robots.

2.2. Neural Controller

The control system (Fig. 4) of each robot consists of a feed-forward neural network with 18 input neurons, 2 hiddens, and 2 output neurons. Each layer of neurons is connected to the next layer with a pattern of synaptic weights representing the strength of the connections. The input layer contains 15 neurons encoding the activation state of the corresponding photoreceptors RGB components, 2 neurons that receive smell signals and 1 neuron that receives output from ground sensor. The output layer is made of 2 neurons which control the speed of two motors, respectively.

2.3. Artificial evolution

The evolutionary process for the robots is based on a ranking type genetic algorithms (e.g. **[14]**). Each individual is represented by a genotype that encodes the sequence of synaptic weights and biases of a neural network controller. Each parameter is encoded with 8 bits. In order to provide robots with different behaviours, each of the four robots belongs to a different population of 100 individuals. Thus, the evolution starts with 4 populations of completely “naive” robots (i.e. with randomly generated genomes) with no skills about how to move and identify the food sources.

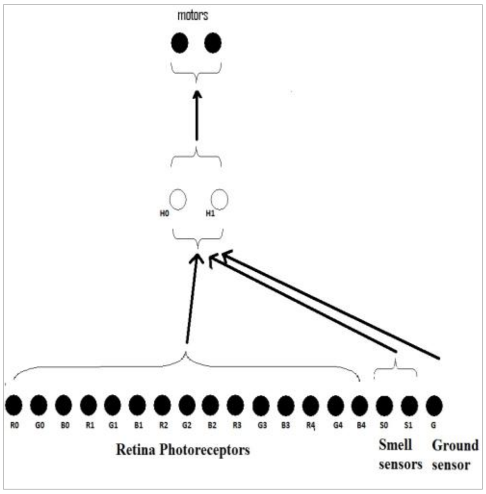


Figure 4: Neural network architecture.

Genotypes are randomly selected within each population: for each generation, individuals of each population is numbered by an index (0-99) and a sequence of indexes is chosen (i.e 3-4-5-4) from the four populations in order to extract the genotype that will control the robots. The first genotype (3), from the first population, controls the green robot, the second genotype, from the second population (4) controls the blue robot and so on. For 100 trials, a new different sequence of individuals is compared in the environment, and robots fitness is calculated at the end of life. If the same individual is extracted in more trials, in different sequences, (i.e 4-6-7-2 for a trial and 3-6-3-1 for another trial), the fitness score of that individual will be averaged over all trials. The same index sequence never will be extracted twice. The extraction of sequences is depicted in Fig. 5.

Each robot is rewarded with +1.0 at a given time step in which the entire group stays in the same food zone. Life time consists of 3000 cycles of neural network activation.

At the end of 100 trials (end of one generation), each individual (neural controller) is separately ranked according to the fitness score. The 20 higher-ranked individuals are selected from the list of genotypes for each population. Each best individual generates 5 offspring individuals which inherit its genotype . The first offspring individual preserves entirely the genotype of the father (elitism) while the other four ones receive a random mutation with a probability of 2%. The total number of new individuals 20(bests) x 5(off) x 4(pop), will populate the next generation. Since, each population evolves separately: this mechanism fosters the genetic differentiation between the four robots and allows the robots to evolve distinctly their behavioral skills.

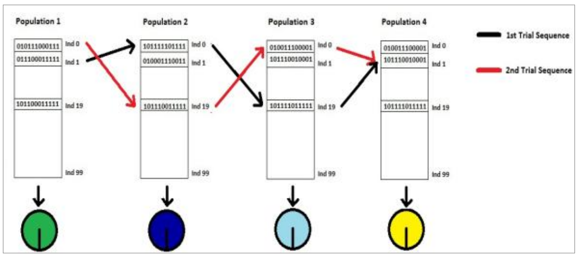


Figure 5: Schematization example of index sequence selection.

3. Results

By evolving the control systems robots for 10 replications with different initial populations and for a total of 300 generations, we observe the emergence of a grouping behavior. For a better understanding of the behavioral observations, we performed some detailed analysis. For each replication (seed) we calculated the average fitness over the last 20 generations, plotted in Fig.6. The variation between seeds suggests that in some replications there could be a stronger grouping and following pattern with respect to others
.

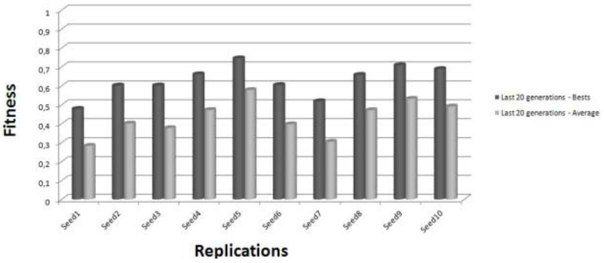


Figure 6: Schematization example of index sequence selecting.

This variation is also confirmed by running tests where we measured which robot in a group, is the closest to the group "center of mass". For each generation, 4 tests are performed by stopping one robot of the group in a fixed position of the environment. Then, the average distance between the fixed individual and the other robots is calculated. In this way, we obtained 4 curves that show the distance of each robot from the group "center of mass" (and example is the plot in Fig. 7).
We can observe that the yellow robot has the minimum average distance from the "center of mass", especially in the last generations. It means that it is always near the centre of the group and the other robots surround or follow it. This fact suggests the emergence of a leader/followers pattern, where the yellow robot is the leader.

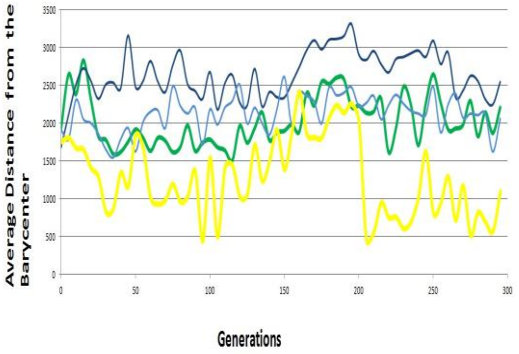


Figure 7: Evolution of distance from group centre of mass over generations. Replication n.9.

It is also interesting to measure the "quality" of the leadership within a group. This measure is obtained by a Leadership Measure (LM) calculated for each replication (Fig. 8). The LM is obtained by calculating the difference between the minimum distance from the group "centre of mass" (Leader) and the average distance of the other 3 individuals (Followers). High differences imply a good extent of a clear leader/followers relationship. Comparing Fig.6 and Fig.8 we can notice that in a replication where there is a high LM, it is also present an high level of fitness. This fact indicates that leadership is a successful strategy in these simulations.

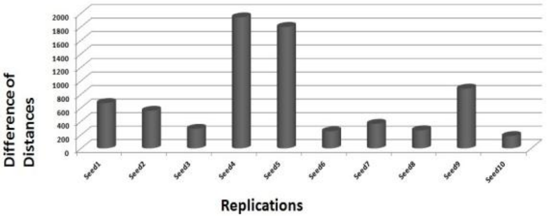


Figure 8: Leadership Measure.

Moreover the second important information emerges when we ran another test in which the fitness of the group and of the individuals is calculated. This test fitness is calculated by testing in the environment only the 4 best individuals for each generation (sampled with a step of 5 generations). Thus, a group fitness and individual fitnesses of each robot are calculated for each generation. The individual fitness, in this test, is taken by summing the times in which a given individual is located in the food zone, independently of the behavior of other robots. It should be noted that this is a virtual fitness, since it is not employed in evolution and it is only used in testing, so to understand the skill of each individual. We hypothesize that those fitnesses should be different, as the robots belong to different populations and play different roles in the group. By plotting individual fitness values for replication n.9, for example, (Fig.9), it is possible to observe how the skills of each group member evolve throughout generations. In this case, after an initial phase (of about 30 generations), where the robots have almost the same individual abilities, the yellow robot consistently reaches better performances. This data prove, in a preliminary way, that whenever there is a strong presence of leadership in one replication, the most skilled individual (i.e. the fastest in approaching the food zone, the one that shows a better exploratory behavior) tend to be the leader of the group. This relation has been also observed in replication 4 and 5, that also show a consistent level of leadership (see figure 8) .

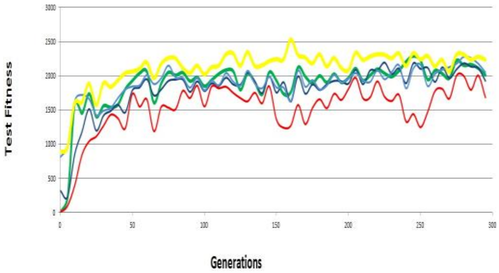


Figure 9: Evolution of abilities in reaching the food zone over generations. The group fitness is represented in red. Replication n.9.

The mechanisms underpinning the emergence of leadership are essentially based on the decision making process within the group. As we can see in figure 8, the yellow robot is the one that shows the best performance within the group. That is, it is the robot that reaches the food zone before the others, and this is true throughout all the evolutionary time. It should be noted that the better performance of the yellow robot, for example, depends entirely on the initial conditions of the population genotypes. In other replications of the same experiment, we observed different populations, i.e. different colours, as emergent leaders of the group.

The fact that the best robot is also the leader can be explained by the fact that it can reaches the food zone faster and more frequently during the different tests. Therefore, during the evolutionary process the other robots of the group can use the best robot within the food zone as a landmark, which helps them to reach and remain within the same area and gain fitness. Thanks to this process, which facilitates the decision of the group towards one of the two areas, the best robot is elected as the leader of the group.

4. Conclusions

Although preliminary, these results show that in a group of robots, with variable distribution of skills (due to different genetic characters), leadership is often observed. In particular, the result of our simulation suggests that the stronger the leadership and the higher the level of the group coordination, the higher the overall fitness of the group. Interestingly, we observed that the robot which emerges as leader is also the best in reaching the food zone and foraging on it. This fact suggests similarities on what is reported in biological literature.

However, more analysis is needed to better understand the process that leads to the emergence of those types of social behaviors and many other questions can be addressed with this kind of simulation, such as, what happens when robots are clones: do leadership/followership patterns emerge? What happens in a condition where the leader is not the individual with the best behavior? How does selective pressure on individual robots favor or inhibit the emergence of leadership? What happens when populations are not segregated and genotypes can mix and compete?

We believe that these kind of questions could be investigated in the future by following and extending the approach preliminarily presented here.

5. References

[1] Terborgh, J. and Janson, C.H., “The socioecology of primate groups”, Annu Rev, Ecol Syst, 17:111-135, 1986.

[2] Alexander, R. D., “The evolution of social behavior. Annual Review of Ecology and Systematics”, 5, 326–383, 1974.

[3] Wrangham, R. W., “An ecological model of female-bonded primate groups. Behaviour”, 75, 262–300.,1980.

[4] Van Schaik, C. P., “Why are diurnal primates living in groups?”, Behaviour, 87, 120–144.,1983

[5] Acerbi, A., Marocco, D., “Orienting Learning by Exploiting Sociality: an Evolutionary Robotics Simulation”, Proceeedings of IJCNN 09, 2009

[6] Gigliotta, O. and Miglino, O., “Groups of Agents with a Leader”, Journal of Artificial Societies and Social Simulation vol. 10, no. 4 1, 2007

[7] Gigliotta, O. and Mirolli, M., “Who Is the Leader? Dynamic Role Allocation Through Communication in a Population of Homogeneous Robots” In Serra R., Villani M., Poli I. (Eds.): Artificial Life and Evolutionary Computation. Proceedings of Wivace 2008. Singapore, World Scientific: 167-177,2009

[8] Guilford, T. and Chappell, J., “When pigeons home alone does flocking have a navigational function ?”, Proceedings of the Royal Society B, 263, 153e156, 1996.

[9] Flack, A., Pettit, B., Freeman, R., Guilford, T. and Biro, D., “What are leaders made of ? The role of individual experience in determining leader-follower relations in homing pigeons”, Animal Behaviour, 2012.

[10] Krause, J., and Ruxton, G. “Living in Groups”, Oxford: Oxford University Press., 2002

[11] King, A. J., Johnson D.P. D., Dominic D.P. Johnson and Van Vugt, M., “The Origins an Evolution of Leadership. Current Biology”, 2009.

[12] Dall, S.R.X., Houston, A.I. and McNamara, J.M., “The behavioural ecology of personality: consistent individual differences from an adaptive perspective”., Ecol. Lett. 7, 734–739.,2004.

[13] Couzin ID, Krause J, Franks NR, Levin SA., “Effective leadership and decision-making in animal groups on the move”. Nature. 2005 Feb 3;433(7025):513-6.,2005.

[14] Nolfi, S., Gigliotta, O., “Evorobot: A tool for running experiments on the evolution of communication and language in embodied agents (pp. 297-301)”., Berlin: Springer.,2010.

Developing Motor Skills for Reaching by Progressively Unlocking Degrees of Freedom on the iCub Humanoid Robot

Salomón Ramírez-Contla[1], *Angelo Cangelosi*[1], *Davide Marocco*[1]

[1]University of Plymouth, Adaptive Behaviour and Cognition Lab
Drake Circus, Plymouth, England

salomon.ramirez-contla@plymouth.ac.uk, angelo.cangelosi@plymouth.ac.uk,
davide-marocco@plymouth.ac.uk

Abstract

To explore development of motor skills for reaching in the iCub robot, we test the capabilities for a neural network controller to learn progressively by locking some degrees of freedom (DOF) of the robot's arm before allowing it to explore the space with more DOF's. We consider exploration and bio-inspired mechanisms can aid in the development of control of the iCub robot arm. Results suggest the advantage of progressive development over an initial full training, also, these pointed out the importance of interaction with the world and the necessity of trial and error occurring in a time lapse for developing of reaching skills.

Index Terms: degrees of freedom, motor skills, development, epigenetic robotics

1. Introduction

Proposed by Bernstein, the degrees of freedom problem [1] poses difficulties for autonomous skills learning and has drawn attention recently in the psychology field [2, 3]. Recent research on robotics [5, 4] has addressed this problem as well and tried to implement some of the ideas proposed by Bernstein due to the nature of recent advances in robotics and the need of developing controllers for redundant robot arms, specially of those of humanoids. Current cognitive robotics research has focused on the importance of the embodiment of an agent in order to richly interact within a world plenty of stimuli and cues that can aid in processes and reduce workload for a central controller such as the brain. The body plays an important role for this interaction and roboticists constantly look for new and better ways to control it.

Studies with evolutionary robotics approaches have been carried out with success for reaching and maipulation tasks. Massera et al. [6, 7] successfully evolved networks capable of fine-grained interaction with objects by exploiting the morphological contraints of a robotic arm. In this work, however, we are interested on the epigenetic development of such tasks.

Development of the human body flows from the top and centre of the body to the limbs. The spinal cord is the starting point, arms, legs, hands and toes take longer to develop. It is said that it follows a proximo-distal and cephalo-caudal direction [8] and this can be appreciated in infants: younger infants move their limbs in broad uncontrolled movements because only the most proximal joints of the limbs have been developed, like the shoulders. Later on, it can be seen that the elbow and wrist also come into play. Also, control of the lower part of the body comes after that of the upper part. In experimental psychology and motor development of humans there is evidence indicating that for learning new skills, adults freeze some of the distal joints involved in the new task until some degree of performance has been achieved, then some more degrees of freedom are used for achieving better performance [9, 10]. In the present work we test if that interaction with the world along with experience limited by constraints imposed by the physical characteristics of the arm, can help the learning process if this is segmented. We use a simulated iCub robot with neural controllers for the arm.

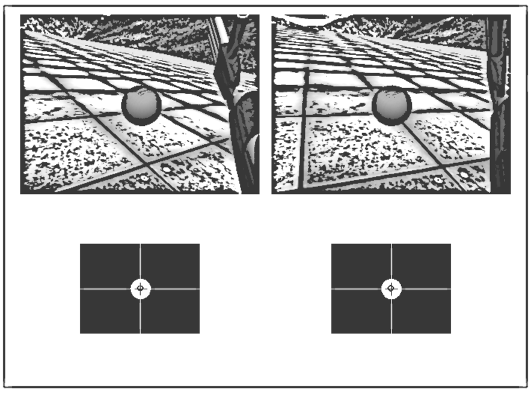


Figure 1: *Images from the robot once the controller has foveated the target. Above, the original images. Below, the low-res colour-segmented images.*

2. Methods

For testing the hypothesis, experiments were planned and carried out on the iCub robot simulator [11]. The iCub robot [12] is a humanoid robot about the size of a four years old child with 53 degrees of freedom designed for cognitive development research. The iCub's head subsystem consists of six degrees of freedom and is capable of vergence (the oculomotor adjustment needed to foveate the same point in space with both eyes). Three degrees of freedom in the head (tilt, pan and eyes' vergence) and four degrees on the arm (two from the shoulder, two from the elbow) were used. The robot head was provided with a visual tracking controller that locates and gazes at a specific target. For the experiments the target was a red ball. The gazing controller performed colour segmenting for the target's colour on the images coming from both eyes. This processing allowed it to track the centre of the target and adjust the position of three joints in the head in order to have the target in the centre of each eyes' field of view (Fig 1). By this mechanism, the robot

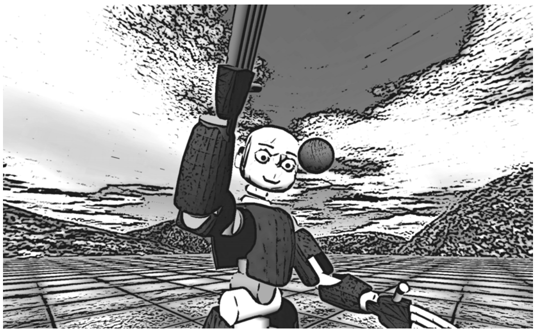


Figure 2: *The iCub performing the reaching task once it has foveated.*

gets information about the depth or distance at which the target is and together with the tilt and pan joints positions, it encodes the positions of the target in space. We use vergence as a depth measure following recent findings [13, 14] that indicate that vergence is in fact one strong signal for depth estimation and programming of prehension movements of humans.

Three different learning conditions (with three networks each) were tested on the robot to test our hypothesis. The first two were: staged learning or development, involving learning control of two DOF's and then the two other (DEV condition), and learning the head-hand associations involving four DOF's from the beginning (NO-DEV condition). For the last condition (NO-TRAIN), a group of three randomly initialized networks were created. These did not go through any learning process and are the control group.

With the help of the gazing mechanism a dataset was captured consisting of joint values of the head and eyes and the joint values of an arm position suitable for locating the end effector (the hand) in the point where the target was. This process can be considered a tutoring stage where the ball was put in the hand every time the robot executed random babbling [15] with the arm, then the gaze controller moved the head for foveating the target. For the cases the head was not able to move to a position were the target could be gazed no data was captured. This train set is equivalent of one acquired by performing random babbling while foveating the target. Reduction of the time required by this process is of course reduced when this kind of tutoring is present, as it happens with infants helped by parents when they start trying to reach objects that are usually out of reach or the baby simply fails to reach.

The controller for the robot arm was a feed-forward network with three inputs (one for each joint of the head controlled by the gazing controller), forty hidden units and four outputs, each of these output units controlled one joint of the arm. Two of these joints are in the shoulder and two in the elbow of the robot. During the initial training set creation, random babbling only occurred for the two most proximal joints of the arm, that is, for the ones in the shoulder. The other two joints were kept in constant values, in positions that we considered natural for an extended, similar to those when performing reaching for objects not very close to the body. Therefore, the positions that can be reached after the initial training are determined by the physical characteristics of the arm and by the generalization capabilities of the network. All learning for the networks was with the back-propagation of error algorithm using a learning rate of 0.01 and a momentum of 0.1.

For the development condition (DEV), each of the three networks was trained using the set acquired via tutoring until the mean square error (MSE) became stable. For the three networks this was around the three hundred epochs. Figure 3 shows the training error for this stage of learning. After the initial training for reaching using two degrees of freedom, a test phase was carried out in ecological conditions. For an extended period of time, the robot was presented with the target in different locations, each time, the robot gazing mechanism was used for gazing the target, then the arm neural controller was activated with the inputs coming from the position of the head and eyes. When the robot successfully reached the target, that is, it touched it, the arm went to it's resting position and the next test target position was presented. Otherwise, the two degrees of freedom that were initially locked (remember their values were constant for the first phase of learning) were randomly moved while the two most proximal degrees of freedom were kept constant with the values the neural controller produced. With this movement the robot was sometimes able to reach the target. When that was the case, the position that enabled it to achieve reaching was stored in a new set that was used for later training. This phase will be called from now on ”experience phase”. Figure 4 shows the training error during the second phase of learning for the three networks and the mean of the three of them.

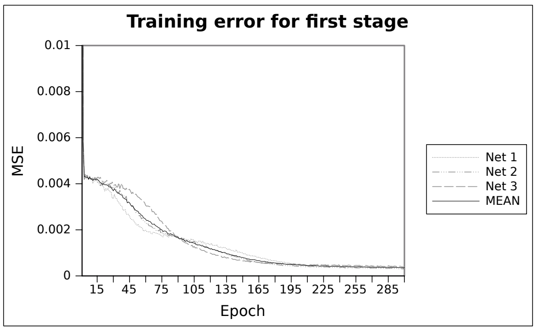


Figure 3: *Error during training of the first stage of development.*

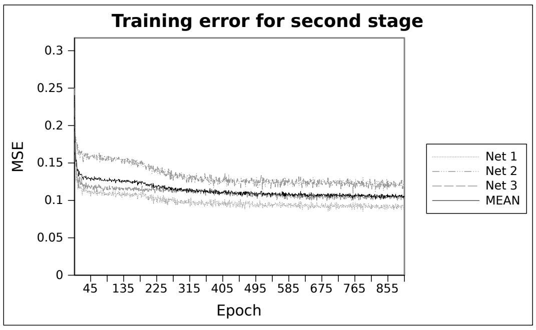


Figure 4: *Error during training of the second stage of development.*

The training using the new set generated in the experience consisted of 900 epochs. Figure 4 shows the error during training of this second stage of the learning.

The second condition (NO-DEV) consisted of using the training set generated during the experience phase on randomly initialised networks without going through an initial, partial, learning phase nor an experience phase. That is, these controllers were trained with the set that uses four degrees of freedom from the beginning. The training was for nine hundred

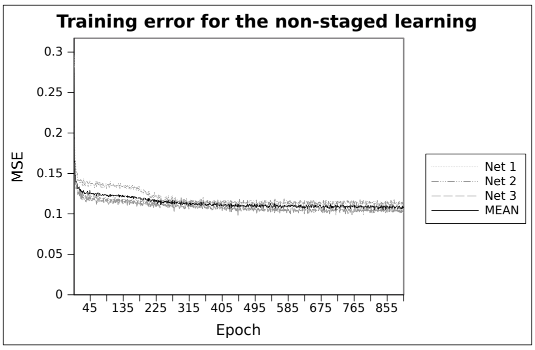


Figure 5: *Error during training of the non-developed controllers.*

epochs, at that point the MSE was stable. Figure 5 shows the error for this no-development learning.

Measurements for comparing the two conditions were performed during the execution of a reaching task similar to the task executed in the experience phase. Figure 2 shows the iCub executing the task once it has foveated the target. Final distance from hand to target was saved for each of the trails of the three controllers. Also, the number of times the controller successfully reached was recorded for having a percentage of success for each of the networks.

3. Results and discussion

Analysis on the output data indicates the controllers belonging to the staged or developmental training performed better in terms of final distance to the target as well as in the percentage of success (Figs 6 and 7). An analysis of variance test was performed to check for statistical difference between conditions (including the non-trained condition). This test reported statistical difference: current effect $F_{2,897}$=850.45, *p*=0.0000.

This can be due to various factors: following a developmental training, consisting of tutoring, experience during operation in it's environment and learning based on that experience could have shaped the weights of the controller's networks to a stage that was able to find a solution for the second training set. Even when the training error of the final training in both conditions is very similar, in test conditions an advantage of the developed can be appreciated.

Because reaching is an important step in the development of motor and cognitive skills, it is also a skill explored to get an insight of the series of processes emerging in infants [16]. Our work on development of reaching tries also to consider the fact that for acquiring a skill it is necessary to has trial-and-error processes where time constraints cannot be avoided. In our experiments, the generation of the second training set for the staged learning condition, the ”experience phase”, took considerably longer than any other part of the experiment. But we believe this was a very important step due to the fact that each network will generate different outputs for the same inputs so the set is particular to each of them.

We have tried to implement what Bernestein [1] suggested for simplifying the degrees-of-freedom problem: in our experiments the robot arm controller goes through a developmental progression in order to find a first but simpler solution to the problem an later on, increasing the complexity of the problem. Other roboticists have implemented similar ideas in recent years. Ivanchenko and Jacobs [17] simulated a three degrees of freedom robot arm that tries to learn the dynamics of the arm while moving on trajectories on two dimensional space. The difference with our approach is that in our case the architecture of the networks is the same for every condition, it is the presence or absence of experience what shapes the performance at their final stages. Ivanchenko has a special architecture, devised from the idea that this decouples dynamic interactions among the joints and therefore allows to separately train the joints. Unlike Ivanchenko, for our experiments we decided to keep the same architecture. We want to explore uncoupling of dynamics without changing the internal (not directly exposed to the environment) characteristics of the system. In Ivanchecnko's, results indicate that a developmentally trained controllers only outperformed the non-developmentally ones when the developmental path matched the nature of the task executed. In the case of our experiments, training as well as the ”experience” phase matched the final task. This could explain the obtained results.

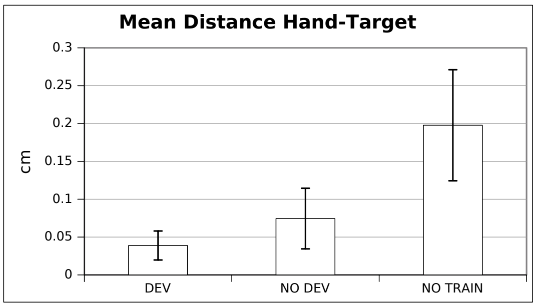


Figure 6: *Mean distance from the center of the palm fo the hand to the center of the target. Each controller was tested with 60 different target locations, none of them belonging to any of the sets used for training. Bard indicate standar deviation for all tests on each condition.*

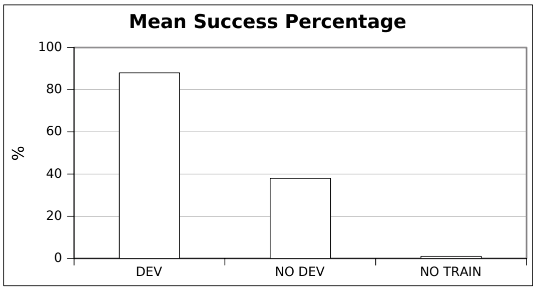


Figure 7: *Results show the percentage of times the end efector touched the target. Each controller was tested with 60 different target locations, none of them belonging to any of the sets used for training.*

As Massera et al. in [6], we have started this exploration on a robotic arm with just four degrees of freedom. Our approach contrasts with that one in that we are interested on the epigenetic development of the skill instead of an evolutionary one. Moreover, in our case, experiments look to include vision into the development of the task instead of direct pass of coordinates or distances to the system without visual processing, as we consider that working towards the implementation of this type of skill development will require real-life sensory capabilities. The head controller for our experiments employs vision as a simple processing but action-involving task. Schlesinger et al. [18]

have also explored with the freezing of DOF's but again, using a non-realistic vision mechanism and a 2D environment and using evolutionary algorithms. Our work has pushed this type of exploration to a more realistic environment and explores the interaction on fixed architecture systems. We showed that even in this circumstances, a developmental approach can lead to better performance. Using the iCub simulator has proven to be a good test-bed for this type of research, as it allowed to implement and test controllers and visual sensors and explore performance in a controlled environment and free of mechanical strain issues.

3.1. Future work

In this study we have investigated the advantages of a progressive unlocking of joints to achieve better reaching performance. Our system has used two and then four degrees of freedom to explore and then improve a motor skill. However, limbs of natural systems, such a humans, display the property of overcompleteness. Overcompleteness implies that even though only 4 degrees of freedom are required for navigating a limb through three dimensional space [19], limbs on many vertebrates usually exhibit more than 4 degrees of freedom. This property turns the problem of controlling a limb more complex in computational terms (at least for traditional control) but also can represent and advantage in terms of the possibility of finding solutions that allow to reach a target at the same time that an obstacle is avoided. This could keep a relation with the representation of the reachable space. Also, constraints in other sensory or mechanical parts will be explored in further work.

4. Acknowledgements

This research has been funded by the National Council on Science and Technology of Mexico (CONACyT) and EU FP7 Project ITALK.

5. References

[1] Bernstein, N., “The coordination and regulation of movements”, Pergamon, 1967.

[2] Karl M. Newell, David E. Vaillancourt, “Dimensional change in motor learning”, Human Movement Science, 20(4–5), pp. 695-715, 2001.

[3] Rohde, M., and Di Paolo, E., “t for Two: Linear Synergy Advances the Evolution of Directional Pointing Behaviour”, Proc. Advances in Artificial Life: 8th European Conference, ECAL 2005, pp.262-271, Springer, 2005.

[4] Berthouze, L., and Lungarella, M., “Motor skill acquisition under environmental perturbations: On the necessity of alternate freezing and freeing of degrees of freedom” Adaptive Behavior, 12(1), pp. 47-64, 2004.

[5] Gómez, G., Lungarella, M., Eggenberger Hotz, P., Matsushita, K., and Pfeifer, R., “Simulating development in a real robot: on the concurrent increase of sensory, motor, and neural complexity”, Proc. of 4th Int. Workshop on Epigenetic Robotics, pp. 119–122, 2004.

[6] G. Massera, A. Cangelosi, S. Nolfi, “Developing a Reaching Behaviour in a Simulated Anthromorphic Robotic Arm Through an Evolutionary Technique”, in Artificial Life X: Proceeding of the Tenth International Conference on the simulation and synthesis of living systems MIT Press, 2006.

[7] Massera G., Cangelosi A., Nolfi S., “Evolution of prehension ability in an anthropomorphic neurorobotic arm”, Frontiers in Neurorobotics, 1(4), 2007.

[8] Sharma, S.K., “Principles Of Growth And Development”, Isha Books, 2005.

[9] Vereijken, B., Van Emmerik, R., Whiting, H.T.A. and Newell, K., “Free(z)ing degrees of freedom in skill acquisition”, Journal of Motor Behavior 24, pp. 133-142, 1992.

[10] McDonald, P. V., Van Emmerik, R. E., Newell, K. M., “The effects of practice on limb kinematics in a throwing task”, Journal of Motor Behavior, 21(3), pp. 245-264, 1989.

[11] Tikhanoff, V., Cangelosi A., Fitzpatrick, P., Metta, G., Natale, L. and Nori, F., “An open-source simulator for cognitive robotics research: The prototype of the iCub humanoid robot simulator”, in Proc. IEEE Workshop Perform. Metrics Intell. Syst., Washington, DC, 2008.

[12] Metta G., Natale L., Nori F., Sandini G., Vernon D., Fadiga L., von Hofsten C., Rosander K., Santos-Victor J., Bernardino A., Montesano L., “The iCub Humanoid Robot: An Open-Systems Platform for Research in Cognitive Development”, Neural Networks, special issue on Social Cognition: From Babies to Robots. 23(8-9), 2010.

[13] Mon-Williams, M. and Dijkerman, H., “The use of vergence information in the programming of prehension”, Experimental Brain Research, 128(4), pp. 578–582, 1999.

[14] Wismeijer, D., van Ee, R., and Erkelens, C., “Depth cues, rather than perceived depth, govern vergence” Experimental Brain Research, 184(1), pp. 61–70, 2008.

[15] Von Hofsten, C., “Eye-hand coordination in the newborn”, Developmental Psychology, 18(3), pp. 450-461, 1982.

[16] Corbetta, D., Thelen, E. and Johnson, K., “Motor constraints on the development of perception-action matching in infant reaching”, Infant Behavior and Development, 23, pp. 351-374, 2000.

[17] Ivanchenko, V. and Jacobs, R. A., “A developmental approach aids motor learning”, Neural Computation, 15(9), Pages 2051-2065, 2003.

[18] Schlesinger, M., Parisi, D. and Langer, J., “Learning to Reach by Constraining the Movement Search Space”, Developmental Science, 3, pp. 67-80, 2000.

[19] Shadmehr, R. and Wise, S. P., “The Computational Neurobiology Of Reaching And Pointing: A Foundation for Motor Learning”, Computational Neuroscience,MIT Press,2005.

Morphology Dependent Distributed Controller for Locomotion in Modular Robots

Avinash Ranganath [1], Juan Gonzalez-Gomez [2], Luis Moreno Lorente [1]

[1] Robotics Lab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Madrid, Spain
[2] Robotics and Cybernetics Group, Departamento de Automática, Ingeniería Electrónica e Informática Industrial, Universidad Politécnica de Madrid, Madrid, Spain
arangana@uc3m.es, juan@iearobotics.com, moreno@ing.uc3m.es

Abstract

Stigmergy is defined as a mechanism of coordination through indirect communication among agents, which can be commonly observed in social insects such as ants. In this work we investigate the emergence of coordination for locomotion in modular robots through indirect communication among modules. We demonstrate how intra-configuration forces that exist between physically connected modules can be used for self-organization in modular robots, and how the emerging global behavior is a result of the morphology of the robotic configuration.

Index Terms: modular robot, locomotion, distributed controller, self-organization, embodiment

1. Introduction

Modular robots are systems composed of several individual unit modules, which with self-reconfigurable capability can autonomously change their morphology. Modular robots can be broadly classified into lattice-type and chain-type systems. Lattice-type systems achieve locomotion through continuous self-reconfiguration, where each module has the ability to move independently in the configuration, giving the notion of modules flowing on the ground and around obstacles. Locomotion in a chain-type system is achieved by controlling the actuator of individual modules in a fixed configuration.

One of the earliest demonstrations of locomotion in chain-type reconfigurable modular robots was provided by Mark Yim in [1], which included several locomotion modes such as walking, crawling, rolling, climbing etc. Distributed controllers for locomotion in chain-type modular robots have been researched in [2], [3], [4], [5] and [6]. Shen et al. have used a biologically inspired method called Digital Hormone Method (DHM) [2], [3], [4] for adaptive communication of state information between modules, based on which a module can decide an action from the gait table, resulting in the emergence of locomotion. Gonzalez-Gomez et al. have demonstrated in [5] how simple sinusoidal oscillators can be used on minimal configuration modular robots with two and three modules to generate locomotion in one and two dimensions respectively. In [6] Ijspreet et al. at the Biorobotics Laboratory, EPFL, have used Central Pattern Generators (CPG) [7] for producing locomotion oscillations on their modular robotic platform called YaMoR. CPGs are specialized neurons found in the spinal cord of vertebrate animals, which have the capability of producing rhythmic output without rhythmic sensory or central input. The mathematical model of CPGs used for controlling locomotion in modular robots are usually one or two CPG neurons per module, which are coupled in different ways with CPGs of other modules based on the configuration.

Though DHM and CPGs are distributed control methodologies, they rely on explicit inter-modular communication. The simple oscillators for locomotion in modular robots demonstrated by Gonzalez-Gomez et al. in [5] is a distributed controller as well, but the phase relation between modules are predetermined, making the controller heterogeneous. We have, in this work, attempted to develop a locomotion controller for chain-type modular robots that is distributed, homogeneous and which does not rely on explicit communication between modules.

2. Simulation and robotic platform

In this work, we test our locomotion controller on modular robotic configurations built using the simulated model of the *Y1* modular robot modules, developed by Juan Gonzalez-Gomez. OpenRAVE is the simulation environment used for experiments in this work. OpenRAVE is physics based, open-source, robotics simulator that has Open Dynamic Engine as its core. The *Y1* modules are an open source, low cost, flexible, and easy to build modular robotic platform, which have been used as a research platform in several research projects. The*Y1*s, as could be seen in Fig.1, are open-ended cube shaped modules, which have a single degree of freedom, with a rotation range of 180º. The dimensions of these modules are 72x52x52 mm. The simulated modules are kept consistent with the real modules, both structure wise, and with respect to actuator features.

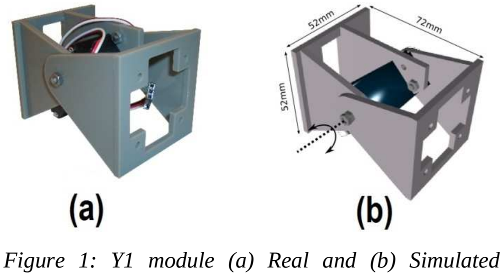


Figure 1: Y1 module (a) Real and (b) Simulated versions.

2.1. Modular robot configurations

We have tested our locomotion controller on three different modular robotic configurations, as could be seen in Fig.2. Each configuration is explained in the following subsections.

2.1.1. Minimal configuration

The Minimal configuration is a two module, one-dimensional configuration, and according to [5], this is the smallest possible configuration for producing locomotion in one-dimension. When both the modules are actuated with simple sinusoidal oscillators with predefined phase difference, they produce a caterpillar gait, which resembles a travelling sine wave, with the phase value determining the direction of locomotion.

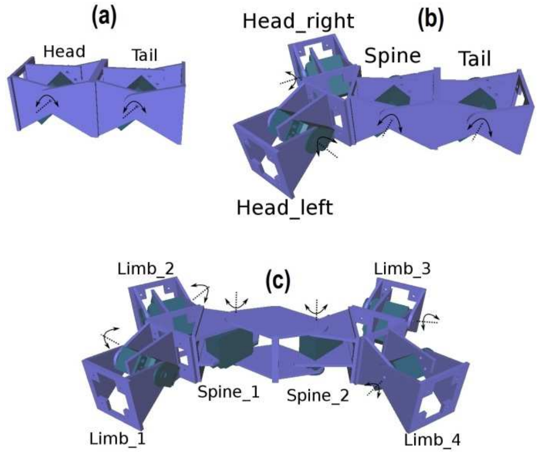


Figure 2: (a) Minimal configuration (b) Y-bot and (c) Lizard.

2.1.2. Y-bot

Y-bot is a four module configuration that can be seen as an extension of the Minimal configuration when two more modules (*Head_left* and *Head_right*) are connected to the *Spine* module at an angle of +/-60º. Locomotion in two-dimension is possible with this configuration, although we focus only on one-dimensional locomotion gait in this work. Again, with simple phase-differed sinusoidal oscillators, this configuration produces a caterpillar like gait, when modules *Head_left* and *Head_right* remain in phase.

2.1.3. Lizard

Lizard is a six module configuration that has four *Limb* modules, and two *Spine* modules. The *Spine* modules are rotated by +/-90º along the pitch axis, in relation with the rest of the configuration. When modules in this configuration are actuated with phase-controlled sinusoidal oscillators, as shown in Table 1 (derived empirically), the result is a quadruped walking gait, resembling that of a reptile.

Table 1. Phase relation between modules in a Lizard configuration with respect to the module 'Limb_1'.

Module	Phase Angle
Limb_1	0º
Limb_2	160º
Spine_1	80º
Spine_2	-80º
Limb_3	160º
Limb_4	0º

3. Controller

Locomotion in general, whether a gallop of a horse, flapping of a bird, or walking of a human, can be seen as repetitive and coordinated movement of limbs, through which the locomotion gait emerges. Looking at locomotion as a collection of oscillators, the phase relation between these oscillators determines the generated gait. This phase relation can be brought about by sharing actuation information among modules through explicit inter-module communication in a modular robotic system. But since a modular robot is an embodied system comprising of physically connected robot modules, our controller relies on the intra-configuration forces that exist among modules for coordination.

3.1. Intra-configuration forces

In a simulated Minimal configuration, when one module is actuated with a sinusoidal oscillator, with amplitude of 60º, and the other module is made to remain at a constant 0º, the oscillating module is seen to affect the other module. As could be seen in Fig.3, the unactuated module oscillates as well with low amplitude and an offset, due to the force exerted on it by the oscillating module. This is because a robot is an embodied system, where physically connected modules exert force on each other when actuated, which can be seen as an implicit communication among modules. Since the simulation tool used here is based on physics, similar (if not exactly the same) results can be expected in the real system.

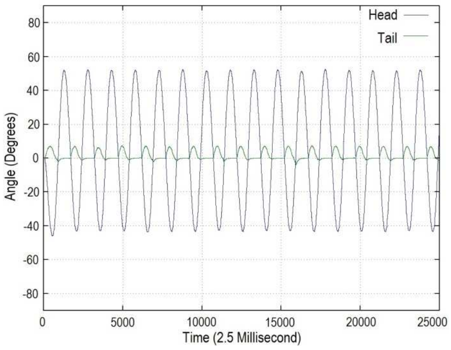


Figure 3: Plot of actuator values in a Minimal configuration, demonstrating the effects of the oscillating module over an unactuated module.

3.2. Simple controller

Since oscillation is fundamental to all locomotion gaits, we made the modules oscillate independently with fixed amplitude and an offset as defined in (1). Conditions (2) and (3) are used to determine if the module's actuator has reached the desired oscillation angle, and if either of the two conditions satisfies, then the direction of rotation of the module's actuator is switched by obtaining the next oscillatory angle from (1). Fig.4(a) depicts the control strategy. Condition (2) checks if the actuator is within a range of $+\alpha$ and $-\alpha$ of the desired position determined by (1). Condition (3) checks if the rate of actuation is above a certain limit specified by β. The value of the parameters A, o, α and β are determined empirically.

$$y_i := (-1)^i A + o, \forall i \in \mathrm{N} \tag{1}$$

$$\left| y - \theta_t \right| \le \alpha \tag{2}$$

$$\Delta\theta_t \le \beta \tag{3}$$

Where y_i is the i^{th} input to the module's actuator, A is the amplitude, o is the offset, θ_t is the positional feedback from the module's actuator at time instance t. Parameters α and β are constants.

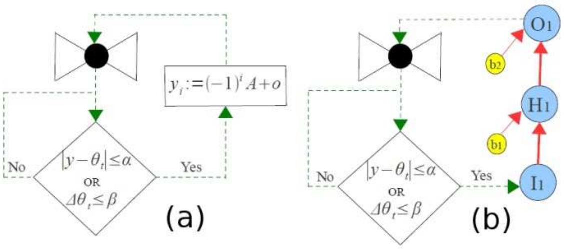


Figure 4: Control flow of (a) Simple controller (b) Neural controller.

3.3. Neural controller

Extending the previous model to include adaptive oscillation rather than a fixed-amplitude-offset oscillator, (1) is replaced with a fully connected feed-forward multilayer perceptron Artificial Neural Network [ANN], as shown in Fig.4(b). The ANN has one input neuron, one hidden layer with a single hidden neuron, and one output neuron. The input to the neural network is the positional feedback from the module's actuator, and the output is the control signal for the same. The lone hidden neuron and the output neuron have one bias node each. *Flood*, an open source ANN library, is used for implementing the ANN. The parameters of this controller are optimized using Genetic Algorithm [GA].

4. Experiment and results

4.1. Evolution

The parameter β and the synaptic weights of the ANN in the neural controller are optimized using GA, individually for each of the three configurations. A robotic configuration is set up in the simulation environment, with each module controlled independently with the neural controller, starting with random initial parameters. The evaluation criteria for evolving optimal parameters, is the distance travelled at the end of the simulation cycle. Each individual in the population is evaluated for 50 seconds in the simulation environment. A fairly standard GA approach is followed, with Roulette Wheel selection method and Intermediate Recombination method for reproducing new offspring. Table 2 contains the GA parameters employed.

Table 2. GA Parameter values used for evolution.

Parameters	Value
Population Size	50
Evolution length	50 generations
Crossover percentage	50.0%
Elite population percentage	12.5%
Mutation rate	1/Size of genome

4.2. Evaluation

The resulting neural controller was evaluated by controlling modules in a given configuration with the most optimal control parameters evolved for that configuration. When actuated, the modules in the Minimal configuration started oscillating in phase, but quickly develop and maintain a steady phase difference, and resulted in a caterpillar locomotion gait. The frequency of oscillation is not predefined in the controller, but intrinsic to the system, and it is inversely-proportional to the amplitude. The amount and stability of phase relation between modules is a result of the morphology. A plot of the oscillation, frequency and phase values of the emerged locomotion gait in this configuration is as shown in Fig.5, Fig.6 and Fig.7 respectively.

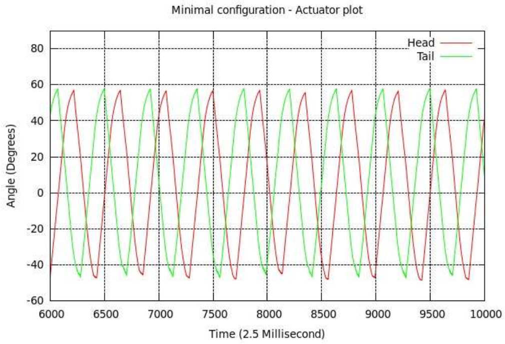


Figure 5: Plot of actuator values in the Minimal configuration actuated with the neural controller.

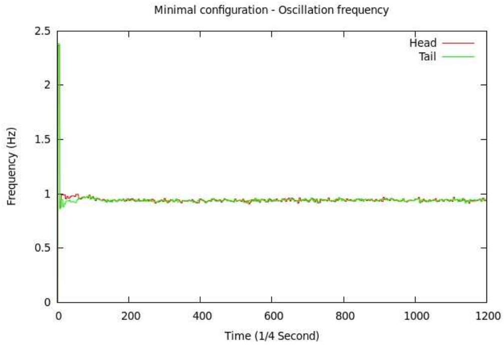


Figure 6: Oscillation frequency graph of modules in the Minimal configuration when actuated with the neural controller.

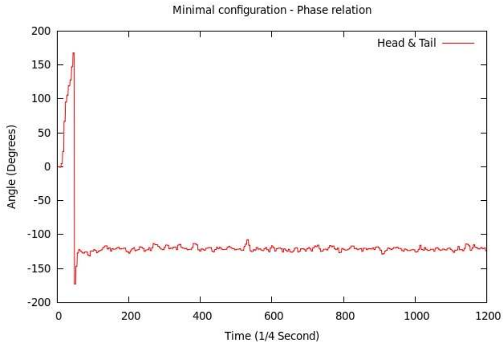


Figure 7: Graph containing phase relation between modules in the Minimal configuration when actuated with the neural controller.

When modules in the Y-bot configuration were actuated with the best evolved neural controller, a similar caterpillar gait emerged and the phase relation graph is as shown in Fig.8. In the Lizard configuration, the neural controller produced a quadruped walking gait, similar to that of a reptile. Each configuration with its respective neural controller was evaluated for a period of 300 seconds. Table 3 contains the speed of locomotion, averaged over 10 evaluations. Fig.9 and Fig.10 contains the phase relation graph of the emerged locomotion gait in the Lizard configuration. The graphs in Fig.9 and Fig.10 are from a single evaluation, but presented separately as two different conventions are used with respect to the Y-axis range for better visualization.

Table 3. Speed of locomotion averaged over 10 evaluations.

Configuration	Speed (Cms/Sec)
Minimal configuration	3.35
Y-bot	4.18
Lizard	2.09

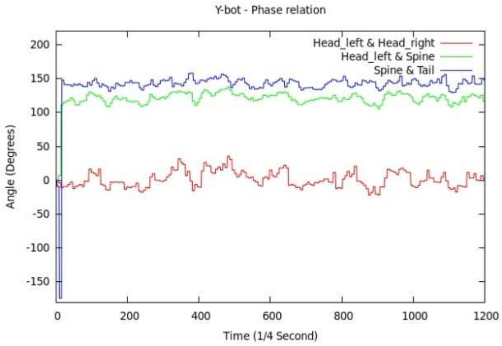


Figure 8: Graph containing phase relation between modules in the Y-bot configuration when actuated with the neural controller.

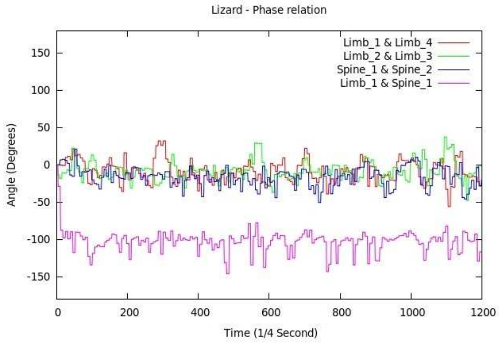


Figure 9: Graph containing phase relation between some pairs of modules in the Lizard configuration when actuated with the neural controller. The phase angle is represented as a value between -180º and +180º for better visualization.

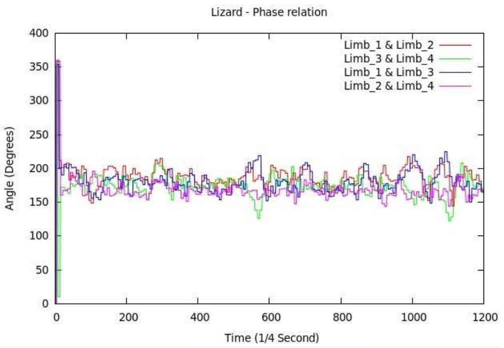


Figure 10: Graph containing phase relation between a few other pairs of modules in the Lizard configuration when actuated with the neural controller. The phase angle is represented as a value between 0º and 359º for better visualization.

4.3. Cross-evaluation

Considering both, the difference in morphology and the dynamics of the emerged locomotion gait in the Y-bot and the Lizard configurations, the required coordination among modules of both the configurations must be very different. To test how a controller evolved for a particular configuration would fair when applied on a different configuration, we cross-evaluated the neural controller evolved for the Y-bot configuration on the Lizard configuration, and vice versa. The emerged locomotion gait when cross-evaluated was virtually similar to the configuration's original locomotion gait in both the cases, implying that the controller is able to adapt its behavior based on the change in morphology.

5. Conclusions

In a multi-robot system like modular robots, coordination among modules is required to produce a stable locomotion gait, and with our controller we have been able to demonstrate how such coordination among modules can emerge based only on indirect local interaction among connected modules, without the need for any direct communication between them. Furthermore, by cross-evaluating the controller, we have been able to demonstrate the dependency of the emerged gait on the morphology of the robot, supporting the notion of embodiment in a robot.

Moving forward, we would like to first evaluate the proposed controller on configurations with real *Y1* modules. In the current model, although the parameter β which determines the actuation rate threshold is optimized using GA, it is a constant during the control phase. We would like to extend our model in such a way that the activation rate threshold value is adaptive during the control phase.

6. References

[1] Yim, M. 1994. Locomotion with a unit-modular reconfigurable robot. Ph.D dissertation, in Department of Mechanical Engineering, Stanford University.

[2] Shen, W.-M., Salemi, B., Will, P. 2000. Hormones for Self-Reconfigurable Robots. in Intelligent Autonomous Systems. Venice, Italy. pp. 918–925.

[3] Salemi, B., Shen, W.-M., Will, P. 2001. Hormone-Controlled Metamorphic Robots. in International Conference on Robotics and Automation. Seoul, Korea.

[4] Hou, F., Shen, W.-M. March 2006. Hormone-inspired Adaptive Distributed Synchronization of Reconfigurable Robots. In The 9th Intl. Conf. Intelligent and Autonomous Systems (IAS-9), Tokyo, Japan.

[5] Gonzalez-Gomez, J., Boemo, E. September 2005. Motion of Minimal Configurations of a Modular Robot: Sinusoidal, Lateral Rolling and Lateral Shift. Proc. of the 8th International Conference on Climbing and Walking Robots, CLAWAR, London. pp. 667-674.

[6] Sproewitz, A., Moeckel, R., Maye, J., Ijspeert, A.J. 2008. Learning to move in modular robots using central pattern generators and online optimization, Int. J. Rob. Res., vol. 27, num. 3-4, p. 423--443.

[7] Ijspeert, A.J. 2008. "Central pattern generators for locomotion control in animals and robots: A review," Neural Networks, vol. 21, no. 4, pp. 642 – 653.

On a design of a torque sensor for the iCub humanoid robot

Wiktor Sieklicki, Francesco Becchi

Telerobot, Genova, Italy
wiktor.sieklicki@gmail.com, becchi@telerobot.it

Abstract

The paper presents the evaluation process of a first version of the one axis torque sensor designed for the iCub humanoid robot. Newly designed strain gauges equipped sensor was found to show a significant readouts hysteresis, therefore several tests were run to define the reason of the hysteresis. Some of the design issues met while testing the new sensor are discussed including the screws connection and relative rigidity of the sensor's elements analyses. Verification of the assembly procedure is also included. Tests revealed several problems on both design stage and exploitation of the sensor. Possible solutions to the encountered problems are further proposed.

Index Terms: torque sensor, strain gauge, friction, fatigue analysis

1. Introduction

A humanoid robot low-level understanding of the environment is provided by the sensors in which it is equipped. Interaction with the real objects and moving within the unspecified environment is only possible, when adequate amount of information is provided. This include external and internal forces and torques applied to the robot's end-effectors which in case of the iCub robot have been sensed with use of four 6-axis Force/Torque sensors placed along the kinematic chain of legs and arms – one sensor in each [1]. Information from those sensors, together with on-fly motor current measurements has been insufficient though for the robot to walk. More detailed information about the torques applied by each motor of the robot can be provided by a joint level torque sensors what shall result in better understanding of the internal and external forces of the manipulators and shall widen the robot's possibilities to interact with the environment [2]. The joint torque level control offers moreover possibility to compensate effects of the robot dynamics without real time computation of the robot dynamics and the control schemes may be robust in respect to parameter variations [3,4]

A single-axis torque sensors were considered to be placed in each powering unit of the robot's lower-body (Fig.1, left). In powering units B to F one torque sensor was considered, whereas since powering unit A consists of four motors, this part was assigned with four torque sensors. Because the iCub robot is very compact, the torque sensor had to be designed specifically for this application with several design limitations imposed. [5]. The torque sensor (Fig.1, e) was considered to be placed in the kinematics chain of each powering unit of the robot after the harmonic drive CSD-17-100 flexspline (Fig.1, c) and before the unit output (Fig.1, a).

Readouts of the newly designed sensor appeared to have a hysteresis though and for this reason a series of sensor's tests were commissioned. This paper presents the evaluation process of a first version of the sensor and discusses some of the sensor's design issues found while testing the sensor. Possible solutions to the encountered problems are also proposed.

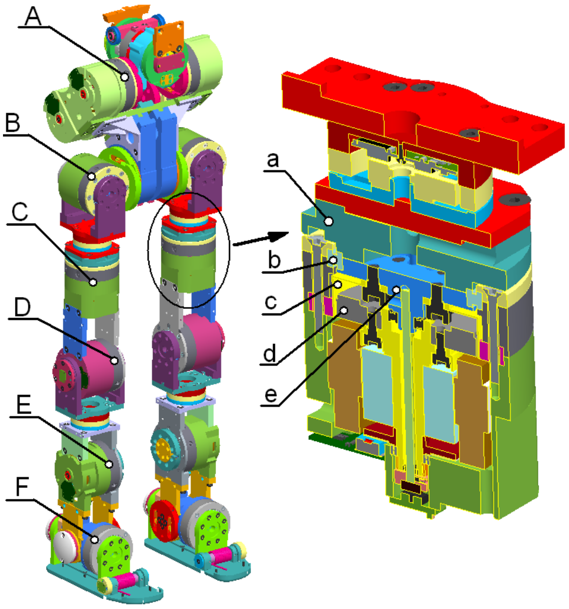


Figure 1: Placement of the 1-axis torque sensors in the iCub robot's structure (left), assembly of a torque sensor (right) – output flange (a), bearing (b), flex spline (c), wave generator (d), sensor (e).

2. The sensor

Tested sensor structure was made of 17-4 ph stainless steel characterized by 1100MPa ultimate tensile stress and 200GPa Young modulus[6]. The sensor design involved an inter mounting part constrained to the harmonic drive with use of 6xM4 8.8 screws on a radius of 7.29mm each, an outer mounting hoop constrained with use of 8xM3 8.8 screws on a radius of 17.5mm and four beams out of which two had strain gauges glued onto (Fig.2). All mounting holes in the sensor were threaded with ISO thread. The sensor in the robot assembly was also supported by KAA15XLO bearing on an outer hoop (Fig.1, b).

The sensor was equipped with SS-060-033-500P Micron Instruments P-doped silicon semiconductor bar-type strain gauges of 500Ohms nominal resistance and 0.84mm of active length [2]. Strain gauges were connected in Wheatstone's bridge design employing four strain gauges arranged in a two half-bridges configuration.

Readings of the sensors were acquired via the CAN bus by the STRAIN board [3]. The utilized board was designed for a six axis Force-Torque sensor used in the earlier versions of iCub. It operates six Wheatstone half-bridges equipped with very same strain gauges as in discussed application, on six independent channels. For sake of this study, only two channels were utilized. To acquire data from the STRAIN board a Canreal software ver. 4.33 was used. Data was displayed with use of Gulp! software ver.0.22 alpha. Offset was set with help of the Gulp! software.

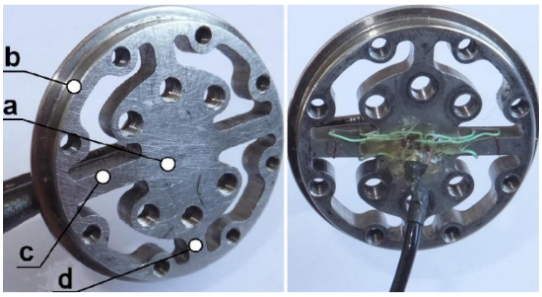


Figure 2: Sensor structure (left) - inner mounting part (a), outer mounting hoop (b), sensor beams (c), short beams (d); strain gauges glued onto the sensor (right).

3. Tests setup

In order to provide most uniform testing conditions special mounting flanges were manufactured. Inner mounting flange was simulating connection with a harmonic drive and was constrained to the table, whereas an outer flange was designed in the way to enable application of a constant torque to the sensor. Sensor was then fixed to the mounting flanges with screws. Torque was applied by loading a rod having 287mm of length in the way, that the load vector was in the plane of the sensor's face surface and perpendicular to the radius of the sensor body. To achieve this, weights were hung on a cord at the end of the rod. The sensor was also tested under equal loads applied in opposite directions (creating a pair of forces), but results were comparable to the previously discussed load case, whereas the test setup was far more complicated. Since the radial deflection of the loaded sensor was noticed to be approx. 0.5˚ it was decided, that the load applied to one of the rods should have given sufficient approximation. Sensor was loaded with loads varying from 1 to 11kg. Maximum torque that the sensor was designed to withstand is 40Nm, what is adequate to the load of 14kg applied on a 287mm long lever. Tests were mostly carried out in the load range of up to 20 Nm because they involved mounting the sensor with some screws loosen what entailed much more stressful load case than provisioned for the sensor. Tests were done in a steady temperature conditions of approximately 20˚C (±5˚C). Hysteresis of the sensor's readouts was observed to be independent from the environment temperature.

4. Tests

Obvious reason for hysteresis in case of strain gauges is gluing to the bending surface, what may result in some amount of permanent deformation of the loaded glue. Whether this was the reason, the sensor was loaded when not constrained rigidly to anything (with loosen all the screws mounting the sensor). Such a test setup resulted in lack of hysteresis. This proved, that the strain gauges are glued correctly to the sensor body.

Verification procedure was begun with checking the signal conditioning system. Alternative acquisition module (ADT4U-RS232, WoBit production) was used together with dedicated software (ADT4U-PC ver.1.02) in order to verify the correctness of the STRAIN board and software functioning. The new acquisition module was earlier tested with other strain gauges presenting no problems. New setup showed the same drawbacks as the original one. Hence it was deduced, that the sensor readouts problems do not origin from electronics nor software errors. Further tests were carried out with use of the original acquisition module and software.

4.1. Screws connection verification

During initial tests it was observed, that the mounting screws tightening torque was significantly influencing the sensor's readouts hysteresis. Tests of friction based connection (with use of not shoulder screws) shown, that in best case of tightening torques (M4 and M3 screws tightened with 3.5 and 2Nm respectively) residuals of the sensor's readouts were varying from 15 up to 24% of the applied load (Fig.2 - dark blue). Tightening screws with higher torques resulted in higher values of residuals, whereas tightening screws with lower torques improved results for low load values but for high load values it made residuals inacceptable high. Following tests were meant to provide information of which connection element causes most of the problems and if use of shoulder screws does alter the results. Testing was divided in two cases - first involving only inner mounting flange connected with the sensor with tightened screws, whereas the outer hoop was attached to the mounting flange with loosen screws, and the later one involving the external mounting flange connected to the sensor's outer hoop with tightened screws, whereas the inner mounting flange was attached with loosen screws.

Tests involving only outer hoop connected with use of tightened, not shoulder screws (tightening torque: 1 - 2.5Nm) revealed that residuals for different tightening torques and different loads applied varied from 5 to 15% of applied load (Fig.2 - dark green). Best results were achieved for the sensor connected by the outer mounting flange with M3 screws tightened with 2Nm torque. Further tests including only inner mounting flange connected with use of tightened, not shoulder screws (tight. torque:1-4Nm) revealed that the residuals varied for different tightening torques and different loads between 0.2 and 3.4% of applied load (Fig.2 - light red). Best results were achieved for the sensor connected by inner mounting flange with M4 screws tightened with 3.5Nm torque.

At this point it is important to note, that M4 screw 8.8 is able to withstand 6.1kN of axial force before elongating plastically (what is not acceptable). The axial force was therefore considered acceptable if the stress in the screws does not exceed 520 MPa, which is 0.65*Ultimate tensile stress. In this case axial force evoked according to Eq.1. in a single screw is 4kN. Tightening torque applied to each screw shall be then 3.48Nm.

$$\mathrm{F}_{\mathrm{axial}}=\frac{\mathrm{M}}{0.5*d2*\left(\tan\left(\mathrm{arctg}\left(\frac{\mathrm{\mu g}}{\cos(\alpha)}\right)+arctg\left(\frac{P}{\pi*d2}\right)\right)\right)+0.5*dh*\mu} \tag{1}$$

,where M is a tightening torque, d_2 is a middle diameter of a screw (=3.545mm), μg is a friction coeff. between surfaces of the thread (=0.15), α is a tread's lead angle (=0.5236rad), P is a tread's pitch (=0.7), dh is a screw's head middle diameter (=5.9mm) and μ is a friction between screw's head and a reciprocal surface (=0.15). Since the friction between the sensor and the inner mounting flange is directly dependent from the axial force evoked by screws, value of the friction force (T) evoked with tightening all six (n) screws is calculated according to Eq.2 to be 3.6kN (assuming μ, as well as μ_g equal 0.15).

$$\mathrm{T}=\mathrm{F}_{\mathrm{axial}}*\mathrm{n}*\mu_{\mathrm{g}} \tag{2}$$

$$\mathrm{M}_{\mathrm{friction}}=\mathrm{T}*\frac{d}{2}*\mu \tag{3}$$

The momentum of friction ($M_{friction}$) between the sensor and the rear mounting flange's surfaces - necessary to keep the sensor on its original position - is according to Eq.3 26Nm, where d is a middle diameter of the surfaces being in contact

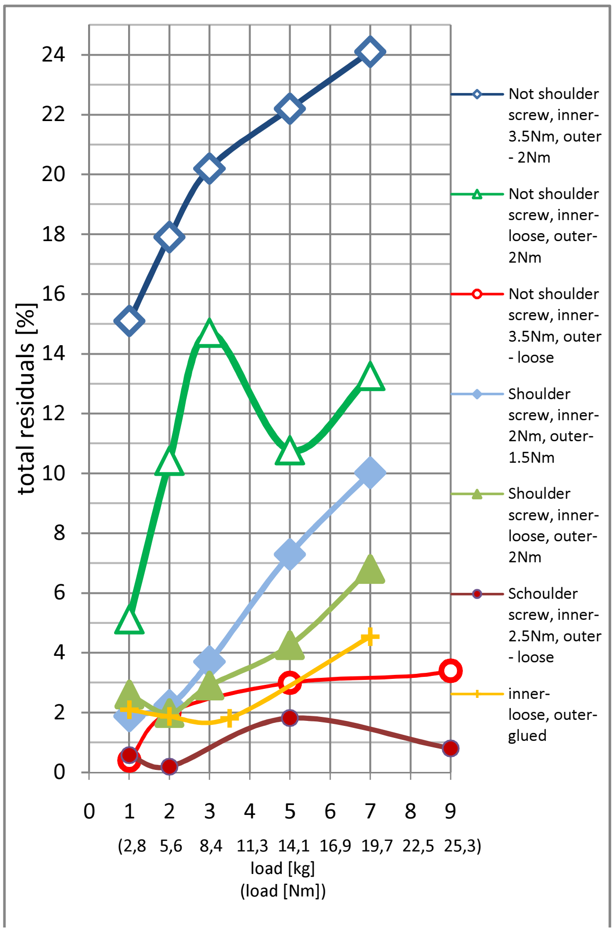


Figure 3: *Residuals of the sensor's readouts in percentage of total load applied to the sensor after the load is removed.*

(=14.58mm). Calculated friction momentum is less than expected 40Nm,thus relative movement between parts appear.

Similar calculations were done for M3 screws. Axial force in every single screw was assumed to be F_{axial}=2.3kN, screws tightening torque M=1.6Nm, middle diameter d_2=2.675mm, friction coeff. μ_g=0.15, thread's lead angle α=0.5236rad, thread's pitch P=0.5, screw's head middle diameter d_h=5mm,friction coeff. μ=0.15, number of screws n=8, middle diameter of surfaces in contact d=17.5mm. In was calculated, that eight M3 screws provide a 48Nm of friction momentum.

Above numbers does not correspond though with the fact, that tightening M3 screws resulted in much higher hysteresis during tests, what was shown in Fig.2. Screws diameter must had been therefore not the only reason for hysteresis. Shoulder screws were introduced in place of the regular ones in order to minimize the relative movement of sensor and mounting flanges. In this case screws' shafts was supposed to keep the sensor on its position rather than friction between surfaces.

Tests involving shoulder screws shown, that when screws at both inner and outer mounting flanges were tightened, best results were obtained for M3 and M4 screws tightened all with torque of 2 and 1.5Nm respectively. Hysteresis was in such case varying from 2.7 to 7.4% of applied load for different torques applied (Fig.2 - light blue).

Similarly to the previous tests, inner and outer mounting flanges were further investigated separately. Outer loop screwed with eight M3 shoulder screws (tight. torques: 1 - 2.5Nm) resulted in the residuals varying from 1.8 to 7.0% of applied load for different torques applied to the sensor (Fig.2 - light green). Tests including only inner mounting flange connected with use of six M4 shoulder screws (tight. torques: 1 - 4Nm) revealed that the residuals varied for different tightening torques and different loads from 0.2 to 3.4% of applied load (Fig.2 - dark red).

Another important aspect of the threaded connection is that there should be enough material left around threaded holes to withstand pressure caused by the screw head when tightening the screws. This apply particularly to the M4 screws, since M3 screws' heads do not come into contact with the sensor in the assembly. From the condition for surface pressures (Eq.4), the bulk material around the threaded hole should have a minimal diameter of (d_m). Applying axial force of 4kN by each the M4 screw implies having a minimal diameter of 6.25mm of material that has to surround the hole.

$$d_m=\sqrt{\frac{4*Faxial}{\pi*\sigma k}+D^2} \qquad (4)$$

, where σ_k is an ultimate tensile stress (1100MPa) multiplied by a safety coeff. (0.65), *D* is an external diameter of a threaded hole. The sensor has bulk material around M4 threaded holes of 5.5mm in diameter, what is not sufficient for the hole to stay undeformed after tightening the screws.

In order to further minimize the hysteresis, the sensor was glued with outer mounting flange (sensor was left attached to the inner m.f with loosen screws). For this reason Epoxy Loctite 9497 A&B Hysol was used. This mounting scheme resulted in significant improvement. In this case hysteresis was varying from 1 to 4.1% of applied load with much less steep characteristics (Fig.2 - orange).

Next candidate to cause the hysteresis was the mismatched relative rigidity of sensor's elements. Finite element analysis was done using ANSYS software in order to evaluate correctness of the sensor's shape design. Particular attention was paid to the external mounting hoop (Fig., b) of the sensor. The reason for investigating this element was that only a deflection of intended elements should be measured by the strain gauges, whereas in this design the rigidity ratio between elements which are supposed to deflect (Fig.2, *c* and *d*) and an element which is supposed to stay rigid (Fig.2, *a* and *b*) seemed to be too small. For the purpose of this analysis a radial displ. of 0.0189rad was applied to each of the mounting holes of the outer mounting hoop, while inner mounting part was constrained (forces and displacements were applied to all nodes of holes surfaces). Such load case resulted in 637MPa of stress (Fig.4, right). To simplify the representation of displacements, the design was transformed into a Cartesian coordinate system (Fig.5) as an infinite subsequent series of two beams (representing entities *c* and *d* from Fig.2) coupled by a thick part from one side and a thin part from the other (which correspond to entities a and b from a Fig.2).

Applying a displacement of f=0.032mm in Y direction to the same holes as in the case of circular design (what represents a displacement of 0.0189rad applied to the circular design) resulted in maximum stress of 640MPa (Fig.4, bottom). Circular and serial designs shows therefore a good approximation. As an outcome of applied translation along Y axis, a lateral translation of 0.023mm along X axis occurred next to the short beams forcing elements to rotate. Unwilled rotation is a result of too rigid short beam in respect to a coupler. The rotation is caused by internal forces which are shown in Fig.6 – in this case a sensor was constrained by holes of the outer mounting hoop, whereas force was applied to the thick part of the sensor.

In this load case reaction forces in X direction evoked at the holes next to the short beam were 678N and -678N, what was in each case approximately one third of a total force applied in Y direction to the sensor. At the same time reaction forces in Y direction evoked at the holes next to the short beam were 839N. Internal forces in this element behave as a pair of forces applied to the coupler rotating it and causing this way a respective movement between the sensor's external hoop and the mounting flange. Therefore it can be stated, that the external hoop of the sensor was not rigid enough to keep the mounting holes in fixed position in respect to each other.

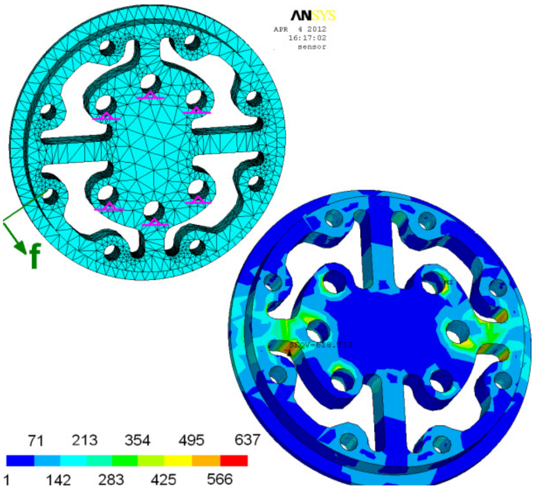


Figure 4: Sensor initial conditions (top) and a stress distribution under f=0.0189rad displacement applied to the mounting holes of outer hoop (bottom).

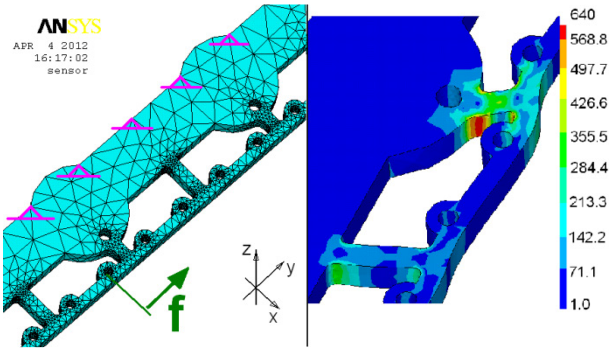


Figure 5: Serial representation of the sensor with analysis initial conditions (left), stress distribution under f=0.034mm displacement applied-displacement representation is scaled up 100 times (right).

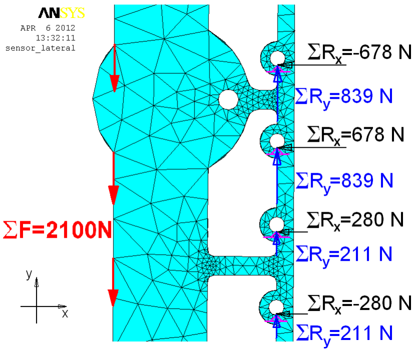


Figure 6: Serial representation of the sensor with internal loads shown.

5. Conclusions

Tests of the sensor revealed, that the hysteresis of the sensor's readouts in its original connection scheme was significant and residuals varied from 10 to 24% of the applied load for different torques applied to the sensor. The sensor's connection to the output/input flanges scheme, relaying on friction was found to be incorrect - sensor was moving in respect to the mounting flanges when load was applied. The connection should have been designed in the way to provide enough friction between contact surfaces or it should rely on pin connection rather than on the friction. Most important result of presented study was that applying torque to the sensor resulted in deformation of the sensor's outer mounting hoop, what caused relative motion of parts hence the friction. Revision of the sensor's design revealed also that amount of bulk material left around the threaded holes was not sufficient - the sensor deflected upon tightening the mounting screws.

Sensor's tests gave several important hints to minimize the readouts' hysteresis. At the design stage of the sensor it is important to decide on a proper mechanical sensor's interface. For friction based interface friction between elements should be enough to keep them with no respective movement. Screws with calibrated shaft gave most repeatable results with less hysteresis in case of discussed case (residuals varying from 2 to 7.2% of applied load for different torques). It should be kept in mind, that this problem arise in case when some of the mechanical interface elements are not rigid enough to stay undeformed. Kind of solution may be gluing parts together (in this case it resulted in hysteresis 5 to 10 times smaller than in the initial tests with friction based connection). It was also shown, that for a given connection scheme – screw based – it is important to maintain a proper tightening torques of screws. With hand screwdriver human can apply much higher tightening torques than M3 or M4 screws can withstand. This may result in plastic elongation of screws and uncertain behavior of the sensor's mechanical interface.

6. Acknowledgements

This research has been supported by the EU FP7 Project RobotDoC ITN (235065 Marie Curie Actions).

7. References

[1] Fumagalli M., Gijsberts A., Ivaldi S., Jamone L., Metta G., Natale L. & Nori F., Learning To Exploit Proximal Force Sensing: a Comparison Approach. In: O. Sigaud & J. Peters (Eds). From Motor to Interaction Learning in Robots, 2010, pp. 159-177

[2] Stokic D. and Vukobratovic M., Historical perspectives and state of the art in joint force sensory feedback control of manipulation robots, 1993, Robotica, 11:149–157

[3] Asada H., Youcef-Toumi K., Direct-Drive Robots. 1987, Cambridge, MA: MIT Press

[4] Kosuge K., Takeuchi H. and Furuta K., Motion control of a robot arm using joint torque sensors. IEEE Transactions on Robotics and Automation, 1990, 6(2):258–263

[5] Tsagarakis N., Vanderborght B., Laffranchi M., Caldwell D. The mechanical design of the new lower body for the child humanoid robot iCub, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, St. Louis, USA, pp. 4962-4968

[6] http://www.aksteel.com/pdf/markets_products/stainless/precipitation/17-4_PH_Data_Bulletin.pdf

[7] http://www.microninstruments.com/datasheets/DS-Bar%20Strain%20Gage.pdf

[8] http://eris.liralab.it/wiki/FT_sensor

Emergent Spontaneous Movements Based on Embodiment: Toward a General Principle for Early Development

Yasunori Yamada [1,2], *Yasuo Kuniyoshi* [1]

[1] Graduate School of Information Science and Technology, The University of Tokyo, Japan
[2] Research fellow of the Japan Society for the Promotion of Science
{y-yamada, kuniyosh}@isi.imi.i.u-tokyo.ac.jp

Abstract

We investigate whether spontaneous movements, which initiate and guide early development in animals, can be accounted for by the properties underlying embodiment. We constructed computer and robotic models of several biological species with biologically plausible musculoskeletal bodies and nervous systems, and extracted the embodied and motor networks based on inter-muscle connectivities. In computer simulations and robot experiments, we found that the embodied and motor networks had similar global and local topologies, suggesting the key role of embodiment in generating spontaneous movements in animals.

Index Terms: embodiment, developmental model, network analysis

1. Introduction

Through evolutionary processes, the animal body and nervous system have mutually adapted in order to achieve efficient sensorimotor integration within the environment. As a result, various adaptive behaviors can emerge from dynamical interactions between the body, nervous system and the environment. This is possible because the neural system exploits the physics of the body on the one hand, while on the other hand, the body dynamics structures the neural dynamics via sensory information. This constitutes a fundamental property of embodied intelligence [1].

Converging developmental studies have emphasized the significance of learning from as early as the fetal period for motor and cognitive development [2]. In particular, these studies have emphasized the importance of spontaneous movements for early development. Recent detailed ultrasound studies on the emergence of fetal motility revealed that spontaneous behaviors start prior to the completion of the spinal reflex arc [3]. Further, these spontaneous movements play an important role in shaping reflex movements and organizing the nervous system in the spinal cord and brain during development [4][5].

Several researchers have suggested Central Pattern Generators (CPGs) as the neural basis for spontaneous movement [3], but how these spontaneous movements emerge in animals with complex and redundant musculoskeletal systems is still not completely understood. Understanding the neural and biomechanical basis of this underlying mechanism can be useful for understanding how spontaneous movements guide early development.

Further, accumulating evidence from developmental research has revealed species generality in the early developmental stage, for example, the dorso-ventral patterning program that characterizes motor neuron and interneuron generation in the spinal cord, progressive phases of limb motor development and motor primitives for locomotion [6][7]. These studies raise the question of whether and how a general mechanism guides early development beyond the difference of body, nervous system and their environment. Yet, few studies have answered this question and constructed a theoretical model for early development.

Our aim was to deepen our understanding of general mechanisms of early development in natural organisms by focusing on spontaneous movements. In this paper, we argue that embodiment generates spontaneous movements and guides early development. Since spontaneous movements precede the development of spinal reflex arc and affects the formation of anatomical and functional neural circuits, it is not necessarily reasonable to assume that innate muscle coordination circuits are required for the generation of the spontaneous movements. We thus predicted that embodiment, which structures sensory-motor interactions, intrinsically contains enough information to generate spontaneous movements. To test our hypothesis, we constructed biological models of several animal species, and investigated their movements in both computer simulations and robot experiments. In a series of experiments, we showed that spontaneous movements can emerge according to their species-specific embodiment from the common neural circuit without any pre-defined muscle coordination circuits in all tested species, and suggesting that this principle can apply to a wide range of species.

2. Materials and methods

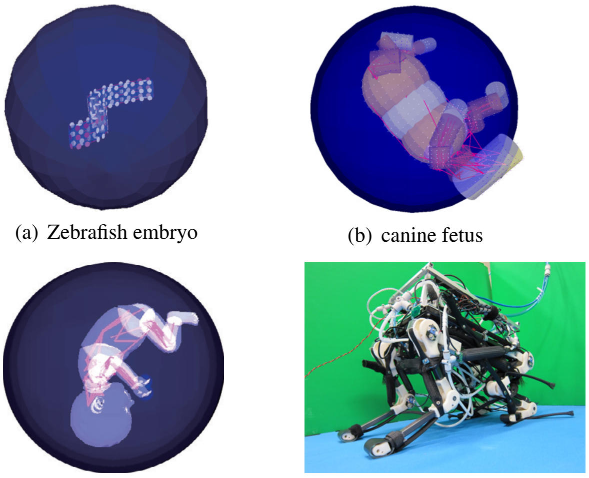
2.1. Biological systems

We constructed three musculoskeletal vertebrate models in computer simulation: the Zebrafish embryo, canine and human fetus models (Fig. 1(a) $\sim$ Fig. 1(c)). Each of these models had parameters that changed with developmental stage.

In the Zebrafish embryo model, the key parameters that we manipulated were size, muscle configuration and the number of somites during the embryonic stage [8]. The number of somites, that is the number of muscles, increased with development.

We constructed the human fetus model based on previous work by Mori and Kuniyoshi [9]. In the human and canine fetus models, the size, mass, moment of inertia of each body part, joint angle limits, muscle configuration and force were manipulated to match those of the fetuses at a gestational age [10][11]. The canine and human fetus models had 170 and 198 muscles, respectively, in the whole body excluding the finger and face muscles.

For the embryonic and fetal environment, we used the amniotic fluid and uterine wall models produced by Mori and Kuniyoshi [9].



(a) Zebrafish embryo

(b) canine fetus

(c) human fetus

(d) quadruped robot

Figure 1: Biological systems. (a)-(c) Blue circle is egg or uterine wall, white and red circle is contact point, and red string is muscle.

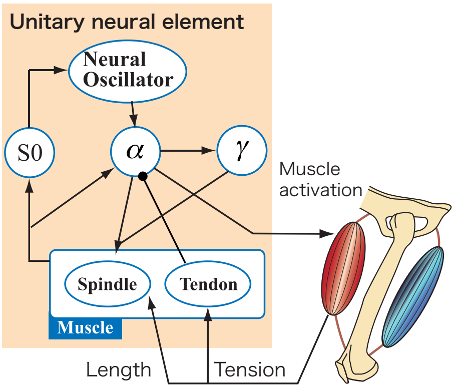


Figure 2: Spinobulbar model. Neural oscillator : neural oscillator neuron model, S0 : afferent sensory interneuron model, α : alpha motor neuron model, γ : gamma motor neuron model, Spindle : muscular sensory organ model, Tendon : Golgi tendon organ model. Arrow and filled circle represent excitatory and inhibitory connections, respectively.

Further, to examine the hypothesis in a real-world environment, we designed a simple quadruped robot that captured important features of the animal musculoskeletal system (Fig. 1(d)). For actuators, we employed McKibben-type pneumatic artificial muscles that reproduced some of the non-linear properties of biological muscles in terms of damping and elasticity. The muscle configuration and sensory feedback of our robot were compatible with the mono- and bi-articular muscles of a quadruped animal, its muscle spindles and Golgi tendon organs.

For the nervous system, we employed the spinobulbar model developed by Kuniyoshi and Sangawa [12]. This model receives muscle length and tension as sensory input, and then outputs the degree of muscle activation as motor command. Each muscle is independently controlled by a single unit within the spinobulbar model (Fig. 2). These muscles are coupled to each other so that if one muscle moves (i.e. contracts), the other muscles change configurations due to the physical constraint of the body (i.e. elongates). Therefore, although individual units of the spinobulbar model are not linked by pre-defined motor coordination circuits, this model can serves to dynamically couple different muscles, and then generate various whole-body movements. In this paper, we refer to *embodied coupling* as such dynamic coupling through the body. This model allows us to investigate whether and how embodiment shapes spontaneous movements through embodied coupling.

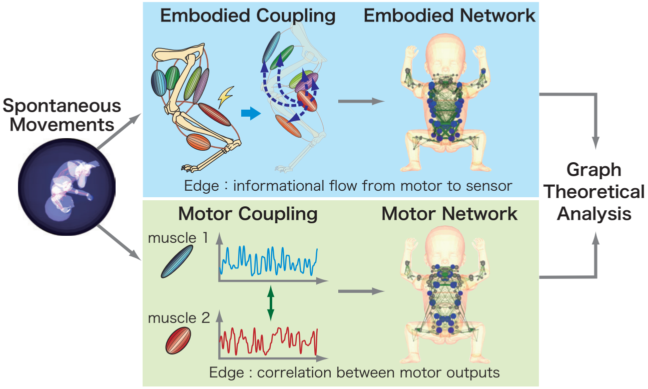

Spontaneous Movements
Embodied Coupling
Embodied Network
Edge : informational flow from motor to sensor
Motor Coupling
Motor Network
muscle 1
muscle 2
Edge : correlation between motor outputs
Graph Theoretical Analysis

Figure 3: Diagram of measuring and analyzing embodied and motor networks based on inter-muscle connectivities.

2.2. Embodied and motor network analysis

To quantitatively characterize embodiment and spontaneous movements in our biological models, we built networks of muscles with inter-muscle connections defined by sensory and motor activations (Fig. 3).

To characterize embodiment, we extracted the *embodied network* which represents patterns of embodied coupling, that is, how much the motor output of one muscle influences the sensory information of other muscles. We computed this embodied coupling with transfer entropy, which captures patterns of directed interaction and information flow [13].

To characterize spontaneous movements, we extracted the *motor network* by measuring the dynamic motor coupling between muscles. The dynamic coordination of motor commands between muscles was quantified by measuring the Pearson correlation coefficient between motor outputs.

We used standard graph measures to analyze and compare the local and global network properties of both the embodied and motor networks, as well as muscle-specific properties within each network.

3. Experiments

We conducted simulations with the Zebrafish embryo, canine and human fetus models, using the open dynamics engine for simulating rigid body dynamics [14]. We set the time step of the simulation to 1 ms, and ran each simulation for $1,000$ s.

We also did several experiments with the quadruped musculoskeletal robot. The robot was mounted with a CPU board running a real-time OS that sent pressure values as motor command and received the length and tension of each pneumatic muscle as sensory feedback every 7.5 ms. One external PC communicated with the CPU board every 100 ms and computed the

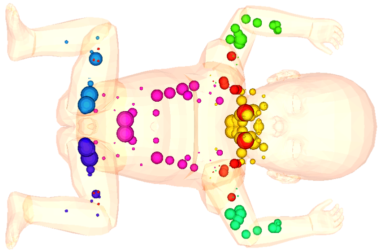


Figure 4: Module decomposition of the motor network in the human fetus model. Circle is muscle, size of circle is within-module degree and colors depict community assignments. The number of modules is 7.

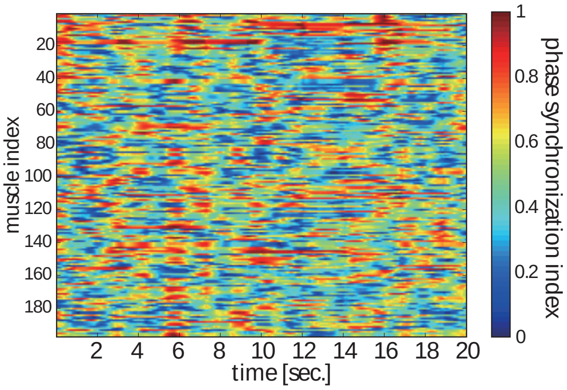


Figure 5: Time series of the instantaneous phase synchronization index (for 200 ms) between one muscle and the others.

neural dynamics every 1 ms.

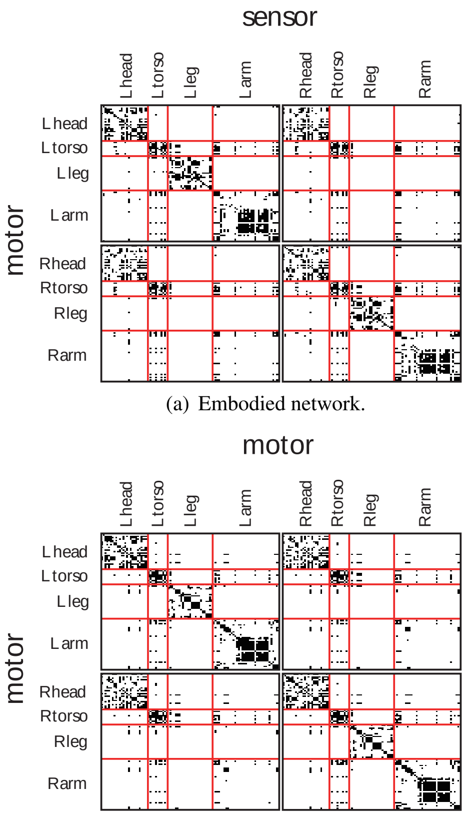
When we constructed the embodied and motor networks, we used muscle length from the muscle spindle model as sensory information and motor commands from the alpha motor neuron model as motor information.

3.1. Emergent movements

Throughout the simulation experiments, the Zebrafish embryo, canine and human fetus models exhibited mixtures of periodic and aperiodic complex movements. In robot experiments, we also observed the robot transited from forward to backward movements, and after several steps it regenerated forward movements. Among other behaviors, there were forward-only and backward-only movements as well as jumping-like motion. Movie of the experiments is available on the first author website[1].

We also investigated the modular architecture in the motor networks to reveal movement modules. We found that modular partition of the canine and human fetus models corresponded to groups of the body parts, such as each leg and arm (Fig. 4).

To reveal dynamic coordination relationship at a time scale of movement unit, we calculated the phase locking value [15], which can identify transient synchrony between muscle pairs on a millisecond scale. Figure 5 shows temporal evolution of the instantaneous synchrony between one muscle and the others.



(a) Embodied network.

(b) Motor network.

Figure 6: Binary networks of the human fetus model. Black squares represent existing connections.

Table 1: Network properties in the human fetus model.

	Embodied	Motor
clustering coefficient	5.63	7.13
characteristic path length	1.74	1.69
modularity	0.69	0.70
small worldness	3.24	4.22
assortativity	0.44	0.33

3.2. Relationship between embodied and motor networks

We carried out a detailed analysis of the embodied-motor networks in all four biological systems (Fig. 6, Table 1). We computed characteristic measures of network organization, including the (normalized) clustering coefficient, (normalized) characteristic path length, modularity, small worldness and assortativity. Normalized measures were computed relative to a set of 100 comparable random graphs.

Both networks showed a high level of clustering coefficient (>1) and a high level of characteristic path length (~ 1), confirming a small-world organization of networks. Modularity of both networks was more than 0.3, suggesting the presence of a significant modular architecture in the networks. Assortativity characterizes network resilience properties against node removal. Both networks showed positive assortativity coefficients, indicating that properties of these networks are relatively robust against the removal of nodes.

[1] http://www.isi.imi.i.u-tokyo.ac.jp/%7Ey-yamada

Further, we compared the node-specific degree, strength and betweenness centrality of the embodied and motor networks. These measures identify node centrality, which is useful diagnostic for comparing topologies. As the results, these measures within embodied networks were significantly correlated with those within the motor network (e.g. degree $r = 0.76$, strength $r = 0.77$ and betweenness centrality $r = 0.60$ for the human fetus model; Student's t-test $p < 0.001$, n = 198).

We note a high level of consistency of these graph measures, suggesting substantial agreement in the topological organization between the embodied and motor networks.

3.3. Relationship between canine and human fetus models

Dominici et al. showed the similarity of motor primitives in locomotor system between humans, other mammals and birds [7].

So, we compared embodied and motor networks with canine and fetus models in the muscles shared by two species according to anatomical knowledge. As the above global graph measures, two models have similar topology both in embodied and motor networks. Node-specific measures also were significantly correlated between emergent motor networks in the canine and human fetus models (degree $r = 0.45$, strength $r = 0.46$ and betweenness centrality $r = 0.38$; Student's t-test $p < 0.001$, n = 170).

4. Discussion

Animals are dynamically coupled to their environments, with embodiment shaping the structure of sensory input, and sensory information determining neural dynamics. In this paper, we explained how such mechanisms occur in neural-body coupling using the biological systems. We also explained how embodiment produces spontaneous movements and attempted to characterize this as a general phenomenon transgressing differences in embodiment. To examine our hypothesis, we introduced a set of quantitative network analysis aimed at capturing the relationship between embodiment and spontaneous movements. We exemplified their use by running computer simulations and robot experiments which produced spontaneous movements.

In the computer simulation and robot experiments, we found the embodied and motor networks share similar topologies of global and node-specific graph metrics. These results suggest spontaneous movements can emerge according to their species-specific embodiment without any pre-defined innate muscle coordination circuit.

Further, we observed the neural-body coupling based on the biological body and nervous system was capable of producing transient synchronization between localized body parts, and resulted in complex and interrelated spatiotemporal behaviors. Fast motor dynamics exhibit intermittent synchronization and desynchronization on a time scale of hundreds of milliseconds, enabling the system to continually explore a repertoire of functional motor coupling.

Experimental studies on animal movements have suggested the existence of motor primitives, or motor synergies, and have studied their organization as a result of learning [4]. However, their neural basis and mechanisms of organization during development remain poorly understood. Our results suggest the embodiment possesses lots of regularities that restrict the number of coordination and allows the animal to explore a variety of embodied dynamics via neural-body coupling. We believe that these explorations via spontaneous movements guided by embodiment could bring about organization of motor synergies as a result of modulation and selection of the motor repertories based on sensory information and motives.

In comparing the canine and human fetus models, we discovered similarities in embodiment and spontaneous movements. Although further research is needed to understand the precise cause and potential implication of the above findings, it is reasonable to assume that similarities in the embodiment and spontaneous movements create, through developmental processes, similarities in motor development and motor synergy for locomotion [6] [7].

In this paper, we quantified the contribution of embodiment in shaping spontaneous movements with biologically realistic systems. We discussed the possibility that the spontaneous movement based on embodiment is a key principle for the early development of natural organisms. In future works, we will investigate and compare how the spontaneous movement shapes coordination neural circuits and guides motor and cognitive development using biological systems of various species at various developmental stages.

5. References

[1] R. Pfeifer and J. C. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, 2006.

[2] M. H. Johnson, *Developmental Cognitive Neuroscience*. Blackwell Publishing, 2005.

[3] M. Hadders-Algra, "Putative neural substrate of normal and abnormal general movements," *Neuroscience & Biobehavioral Reviews*, vol. 31, no. 8, pp. 1181–1190, 2007.

[4] M. Granmo, P. Petersson, and J. Schouenborg, "Action-based body maps in the spinal cord emerge from a transitory floating organization," *The Journal of Neuroscience*, vol. 28, no. 21, pp. 5494–5503, 2008.

[5] M. Milh, A. Kaminska, C. Huon, A. Lapillonne, Y. Ben-Ari, and R. Khazipov, "Rapid cortical oscillations and early motor activity in premature human neonate," *Cerebral Cortex*, vol. 17, no. 7, pp. 1582–1594, 2007.

[6] S. Grillner and T. Jessell, "Measured motion: searching for simplicity in spinal locomotor networks," *Current opinion in neurobiology*, vol. 19, no. 6, pp. 572–586, 2009.

[7] N. Dominici, Y. Ivanenko, G. Cappellini, *et al.*, "Locomotor primitives in newborn babies and their development," *Science*, vol. 334, no. 6058, pp. 997–999, 2011.

[8] C. Kimmel, W. Ballard, S. Kimmel, B. Ullmann, and T. Schilling, "Stages of embryonic development of the zebrafish," *American Journal of Anatomy*, vol. 203, no. 3, pp. 253–310, 1995.

[9] H. Mori and Y. Kuniyoshi, "A human fetus development simulation: Self-organization of behaviors through tactile sensation," in *IEEE 9th International Conference on Development and Learning*, 2010, pp. 82–87.

[10] A. C. Andersen, *The Beagle as an Experimental Dog*. The Iowa State University Press, 1970.

[11] J. Archie, J. Collins, and R. Lebel, "Quantitative standards for fetal and neonatal autopsy," *American journal of clinical pathology*, vol. 126, no. 2, pp. 256–265, 2006.

[12] Y. Kuniyoshi and S. Sangawa, "Early motor development from partially ordered neural-body dynamics: experiments with a cortico-spinal-musculo-sleletal model," *Biological Cybernetics*, vol. 95, pp. 589–605, 2006.

[13] T. Schreiber, "Measuring information transfer," *Physical Review Letters*, vol. 85, no. 2, pp. 461–464, 2000.

[14] R. Smith, "Open Dynamics Engine - ODE," http://www.ode.org/.

[15] M. Rosenblum, A. Pikovsky, J. Kurths, C. Schäfer, and P. Tass, "Phase synchronization: from theory to data analysis," in *Neuro-Informatics and Neural Modelling*, ser. Handbook of Biological Physics, F. Moss and S. Gielen, Eds. Amsterdam, 2001, vol. 4, pp. 279–321.

Integrated Model for Generating Non Verbal Body Behavior Based on Psycholinguistic Analysis in Human-Robot Interaction

Amir Aly and Adriana Tapus

Cognitive Robotics Lab - ENSTA-ParisTech, France
{amir.aly, adriana.tapus}@ensta-paristech.fr

Abstract

Robots are more and more present in our daily lives. They have to move in human-centered environments, to interact with humans, and obey some social rules so as to produce an appropriate social behavior in accordance with human's profile (i.e., personality, emotional state, and preferences). The user's personal profile links between different ways of communication like the verbal, nonverbal, and para-verbal. Verbal and nonverbal communication play a major role in transferring and understanding messages in a social interaction between a human and a robot, because of their natural alignment and synchronization. The nonverbal behavior can be generated based on a linguistic and contextual analysis of the verbal language, relying on rules derived from extensive research into human conversational behavior.

This study defines a new mechanism of generating gestures, in parallel with generated natural language based on human personality. Our model contains the following steps:

- Speech recognition platform.
- Personality recognition analysis.
- Natural language generation.
- Gestures generation corresponding to the generated language.
- Transferring the data of the generated gestures in real time to the humanoid robot (in our case Nao robot).

In this work, we used Dragon (Dragon Naturally Speaking 11.5) speech recognition system that can dictate continuous speech into a text with high accuracy. This text is used for the user's personality recognition analysis. The personality of the user is expressed through the Big Five personality dimensions [1]: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, based on some psycholinguistic cues in the written text [2].The natural language generator (PERSONAGE) developed by Mairesse and Walker in [3], receives the description of the human's personality as input, and tries to generate a corresponding text to the personality dimension to be used as a verbal reaction by the robot, based on the traditional pipelined natural language generation (NLG) architecture [4]. PERSONAGE generator was mainly developed to produce personality based utterances for the restaurants' domain in New York City, but it can also be extended to other domains, and this point is still under development. The user's personality and the gestures are highly correlated. In [5] the authors discussed the effect of the personality traits on the characteristics of the performed gestures (e.g., amplitude, direction, rate, and speed). Similarly, they can influence the verbal content of speech in terms of (verbosity, repetitions, etc). Moreover, Nass et al. in [7] discuss the similarity-attraction principle (i.e., individuals are attracted by others with the same personality traits). All this work constituted the inspiration of this current work, where we try to use the user's personality traits as an intermediate step towards automatically generating robot gestures based on the generated text that matches the user's personality traits.

Moreover, we use (BEAT) toolkit for generating different kind of gestures (e.g., eyebrows, iconic, beat, and deictic) based on the generated utterances in the previous step, from which it extracts some linguistic features (e.g., theme/rheme) in order to generate a series of synchronized gestures [6]. BEAT is driven by hand made rules synchronizing gestures with linguistic cues obtained from live conversations. However, the existing system doesn't include a lot of gesture types. In this work, we try to extend the existing model by training the system over new rules characterizing new kind of gestures to increase the gestural expressiveness toolkit.

The last step concerns the modeling of both the generated gestures and the generated language on the robot's behavior. In this way, we expect to obtain a multi-modal customized robot behavior capable to interact with humans autonomously in different contexts and scenarios.

A video of our system with Nao robot is available at: http://www.ensta-paristech.fr/~tapus/HRIAA/media.

1. References

[1] W.T. Norman, "Toward an adequate taxonomy of personality attributes: replicated factor structure in peer nomination personality rating", Journal of Abnormal and Social Psychology, 66, 574-583, 1963.

[2] F. Mairesse, M. Walker, Matthias Mehl, and Roger Moore, "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text", Journal of Artificial Intelligence Research (JAIR), 30, pages 457-500, 2007.

[3] F. Mairesse, and M. Walker, "Controlling User Perceptions of Linguistic Style: Trainable Generation of Personality Traits", Computational Linguistics, 37(3), 2011.

[4] E. Reiter, and R. Dale, "Building Natural Language Generation Systems", Cambridge University Press, 2000.

[5] M. Neff, Y. Wang, R. Abbott, and M. Walker, "Evaluating the Effect of Gesture and Language on Personality Perception in Conversational Agents", Proceedings of Intelligent Virtual Agents (IVA), 2010.

[6] J. Cassell, H.H. Vilhjalmsson, and T. Bickmore, "BEAT: The Behavior Expression Animation Toolkit", Proceedings of SIGGRAPH, 477- 486, 2001.

[7] C. Nass and M. K. Lee, "Does Computer-Synthesized Speech Manifest Personality? Experimental tests of recognition, similarity-attraction, and consistency attraction." Journal of Experimental Psychology: Applied, 7(3), 171-181, 2001.

From Low Level Motor Control to High Level Interaction Skills

David Bailly, Pierre Andry, Philippe Gaussier

ETIS, ENSEA, University Cergy-Pontoise, CNRS UMR 8051, F95000 France
david.bailly@ensea.fr, pierre.andry@ensea.fr, philippe.gaussier@ensea.fr

1. abstract

The goal of this research is to create a non-verbal system able to interact safely and naturally with humans. The main hypothesis is that mechanisms of high level interactions such as cooperation and understanding intentions can be obtained from well designed low-level systems. For example, an effector device instrumented to detect force constraints applied by others allows to get easily the direction (opposing vs facilitating) and, at a higher level of interpretation, the intention of others concerning the device's movement. This is one of the reasons we preferred hydraulic technology which presents a potential of physical compliance. Moreover, pressure control in the pistons is closer to muscles control than the electric motors.

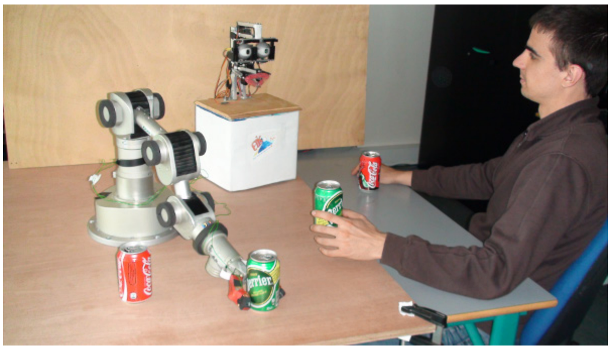


Figure 1: *Example of the robotic devices used in our experiments. A 6 DOF robotic arm is used in conjunction with a robotic head for visual perception and interaction.*

For the control architecture, we are interested in modeling the layers of motor command : low level force control, multimodal inputs (especially vision) leading to prediction and anticipation capabilities. To do so, this research includes the design of a bio-inspired neural network able to provide a force control of the hardware and merging inputs from different kind of sensors including vision and proprioception. The control has to be as close as possible to the hardware with the less layer possible. It is based on a control by activation of agonist and antagonist muscles. The position and torque sensor as well as short range proximity sensor are used to learn simple movements and their sensory outcome. The vision is also available through robotic eye mounted on a fast pan-tilt system allowing movement at human speed. High definition camera gives a video flow that can be used to analyze the scene. The neural network designed allows the system to analyze the scene using point of interest. By extracting local features around those points it is possible to construct a library of visual feature. Using this library objects can be recognize by learning simple associations between those local feature and sensorial context including supervision signals. Action can then be associated with the context or the presence of an object. Moreover sequences of simple actions can be learned through cognitive maps. For example the robot can learn from the human teacher to grasp, move and release an object. From then and with the recognition of object the robot is able to learn tasks such as sorting objects using their visual characteristic.

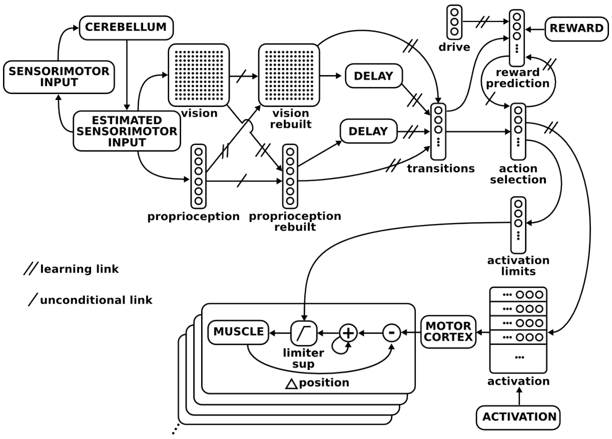


Figure 2: *Overview of the neural network model used to learn on-line sequences of gestures.*

As we construct this controller we hope to improve our knowledge of some structures of the brain such as the motor cortex, the pre-frontal cortex, the striatum or the cerebellum. Models of all these structures and other are used in the model here developed. The researches aim especially to better understand the influence of each structure on the global behavior of the robot as well as the synergies that emerge from the cooperation between structures and to create a new type of humanoid robot where all parts from the technology, through the low level control to the high level control is thought in the optic of realistic interactions with humans.

2. Acknowledgements

This work is part of the INTERACT ANR09-COORD014 project, part of the french ANR CONTINT research.

From typical neurocognitive development to neurorehabilitation of autistic children using mobile toy robots

Irini Giannopulu

University Pierre and Marie Curie, Paris VI
23, rue de Montparnasse & 4, Place Jussier
75006 Paris, France

Abstract

Typical neurocognitive development is based on multimodal interactions. One way to study multimodal interactions is to analyze how children learn language. The studies we conducted aim to understand the development of cognitive non verbal tasks including -attention, action/gesture imitation, haptic and visual perception tasks -and their relationship with the development of verbal tasks in children aged 5 to 6 years old. The above considerations relevant to the natural and neuronal environments were taken into account for the neurorehabilitation of autistic children using artificial environments rendered possible through the use of mobile toy robots. Autism which is a complex neurocognitive disorder is characterized by troubles in communication as well as deficits in the cognitive treatment of emotions. We designed four studies whose aim was to quantitatively and qualitatively evaluate the multimodal interaction between autistic children and a mobile toy robot during free spontaneous game play. A range of cognitive non verbal criteria including eye contact, touch, manipulation, and posture were analyzed, firstly in a dyadic interaction and secondly in a triadic interaction. The dyadic interaction of autistic children and a mobile toy robot suggests that the mobile toy robot in an ecological situation such as free, spontaneous game play could be used as a neural mediator in order to reduce skill impairment witnessed in autistic children. The analysis of the triadic interaction involving an autistic child, the robot and a therapist concludes that once the robot-child interaction has been established, the child can use the robot as a mediator to express positive emotion and play with a person. Therefore robot therapy could conceivably have a high potential to improve the condition of brain activity in autistic children.

Online Language Learning to Perform and Describe Actions for Human-Robot Interaction

Xavier Hinaut[1,2], *Maxime Petit*[1,2], *Peter F. Dominey*[1,2]

[1] Stem cell and Brain Research Institute, INSERM U846, 18 Avenue Doyen Lepine, 69500 Bron, France
[2] Université de Lyon, Université Lyon I, 69003, Lyon, France

{xavier.hinaut ; maxime.petit ; peter.dominey}@inserm.fr

Abstract

The goal of this research is to provide a real-time and adaptive spoken langue interface between humans and a humanoid robot. The system should be able to learn new grammatical constructions in real-time, and then use them immediately following or in a later interactive session. In order to achieve this we use a recurrent neural network of 500 neurons - echo state network with leaky neurons [1].

The model processes sentences as grammatical constructions, in which the semantic words (nouns and verbs) are extracted and stored in working memory, and the grammatical words (prepositions, auxiliary verbs, etc.) are inputs to the network. The trained network outputs code the role (predicate, agent, object/location) that each semantic word takes. In the final output, the stored semantic words are then mapped onto their respective roles. The model thus learns the mappings between the grammatical structure of sentences and their meanings.

The humanoid robot is an iCub [2] who interacts around a instrumented tactile table (ReacTable^{TM}) on which objects can be manipulated by both human and robot. A sensory system has been developed to extract spatial relations. A speech recognition and text to speech off-the-shelf tool allows spoken communication. In parallel, the robot has a small set of actions (put(object, location), grasp(object), point(object)). These spatial relations, and action definitions form the meanings that are to be linked to sentences in the learned grammatical constructions.

The target behavior of the system is to learn two conditions. In action performing (AP), the system should learn to generate the proper robot command, given a spoken input sentence. In scene description (SD), the system should learn to describe scenes given the extracted spatial relation.

Training corpus for the neural model can be generated by the interaction with the user teaching the robot by describing spatial relations or actions, creating <sentence, meaning> pairs. It could also be edited by hand to avoid speech recognition errors. These interactions between the different components of the system are shown in the Figure 1.

The neural model processes grammatical constructions where semantic words (*e.g. put*, *grasp*, *toy*, *left*, *right*) are replaced by a common marker. This is done with only a predefined set of grammatical words (*after*, *and*, *before*, *it*, *on*, *the*, *then*, *to*, *you*). Therefore the model is able to deal with sentences that have the same constructions than previously seen sentences.

In the AP condition, we demonstrate that the model can learn and generalize to complex sentences including "*Before you put the toy on the left point the drums.*"; the robot will first point the drums and then put the toy on the left: showing here that the network is able to establish the proper chronological order of actions.

Likewise, in the SD condition, the system can be exposed to a new scene and produce a description such as "To the left of the drums and to the right of the toy is the trumpet."

In future research we can exploit this learning system in the context of human language development. In addition, the neural model could enable errors recovery from speech to text recognition.

Index Terms: human-robot interaction, echo state network, online learning, iCub, language learning.

References

[1] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks", Tech. Rep. GMD Report 148, German National Research Center for Information Technology, 2001.

[2] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *PerMIS: Performance Metrics for Intelligent Systems Workshop*, Washington DC, USA, pp. 19-21, 2008.

Acknowledgements

This work has been financed by the FP7-ICT 231267 Project Organic and by the FP7-ICT-270490 Project EFAA. Neural model has been developed with Oger toolbox: http://reservoir-computing.org/organic/engine.

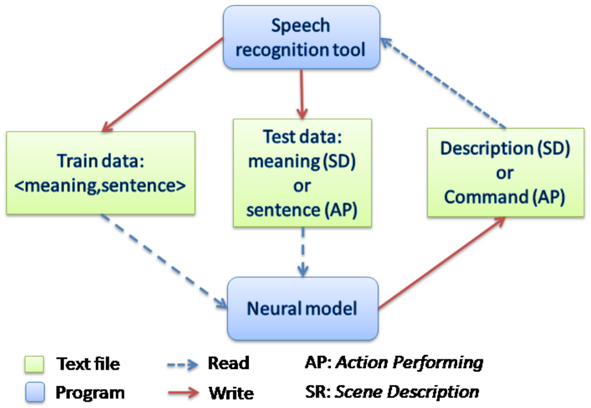


Figure 1: Communication between the speech recognition tool (that also controls the robotic platform) and the neural model.

Infants' perception from the physical relations between objects

Lauriane Rat-Fischer, Kevin J. O'Regan, Jacqueline Fagard

CNRS UMR 8158 – Université Paris Descartes, France

Abstract

Learning to use a tool is a critical step in human development. Recent work has identified the developmental steps leading to the emergence of tool-using in infants (e.g. [2], [4]). These longitudinal and cross-sectional studies show evidence for the beginning of tool-using in infants from the age of 18 months. The tool use studied in these studies refers to the retrieving of an out-of-reach toy with a rake-like tool, when there is a spatial gap between the toy and the tool.

It is surprising that the ability to use a tool to retrieve an out-of-reach toy appears so late in the development, whereas infants are able to combine two objects starting from the age of 10 months, and achieve more and more complex object combination during their second year of life (see for example [1]). Why does tool-use emerge so late? One possible explanation is the change in infants' ability to attend to more than one item in the environment at the same time [3]. This raises the question of what infants perceive in their environment when trying to solve a task like retrieving an out-of-reach object, and in particular what infants perceive from the physical relations between the toy and the tool.

We explored this question using the string paradigm: infants were presented with an out-of-reach object connected to a string that was within reach. Infants are known to be able to pull a string to retrieve an object attached to it starting from the age of 10 months [6]. However, when 16-month-olds are presented with four strings, only one of them connected to the toy, they often fail to pull the connected string and instead pull any string at random [5]. To check infants' attentional behaviour toward the connection, we used a Tobii eye tracker with a scene camera to see which string the infants looked at when they saw someone preparing to do the task. We tested infants aged 16, 20 and 24 months.

The preliminary results show that infants older than 16 months looked at the correct string prior to the adult's movement. Gaze analyses are still in progress, but we expect that infants who failed to pull the correct string (mostly 16 month-olds) will have different attentional strategies than successful infants. This attentional mechanism might be directly correlated with the strategies used for the tool task around the period where infants start to succeed in using a tool. The aim of the whole research program is to understand more deeply the mechanisms underlying tool-use learning in infants, in the view of implementing them in a computational model that might be of relevance to autonomous learning of tool-use in robots.

1. References

[1] Hayashi M and Matsuzawa T. (2003), “Cognitive development in object manipulation by infant chimpanzees”, Anim. Cogn. 6 (4)1225-1233.

[2] O'Regan K, Rat-Fischer L and Fagard J (2011), “Mechanisms leading to tool use: Alongitudinal study in human infants”, Front. Comput. Neurosci. Conference Abstract: IEEE ICDL-EPIROB 2011

[3] Rat-Fischer L, O'Regan J and Fagard J., “The Emergence of tool-using during the second year of life”, Subm. at Journ. of Exp. Child. Psych.

[4] Rat-Fischer L, O'Regan J and Fagard J (2011a), “Tool-use learning mechanisms at the end of the second year in human infants”, Front. Comput. Neurosci. Conference Abstract: IEEE ICDL-EPIROB 2011

[5] Rat-Fischer L, O'Regan J and Fagard J (2011b), “Using eye tracking to investigate the notion of connectedness in infancy: the example of the string paradigm”, Poster presented at Eyetrackids Conference 2011

[6] Richardson HM (1932), “The growth of adaptive behavior in infants: An experimental study of seven age levels”, Genet. Psych. Mono., 12, 195-359.

Hebb-like Learning for the Grounding of High-Level Symbols in Sensorimotor Trajectories

Martin F. Stoelen[1], *Davide Marocco*[2], *Angelo Cangelosi*[2], *Fabio Bonsignorio*[1,3], *Carlos Balaguer*[1]

[1]RoboticsLab, Universidad Carlos III de Madrid, Madrid, Spain
[2]Centre for Robotics and Neural Systems, University of Plymouth, Plymouth, UK
[3]Heron Robots, Genova, Italy
(mstoelen,fbonsign,balaguer)@ing.uc3m.es, (davide.marocco,acangelosi)@plymouth.ac.uk

Abstract

The symbol grounding problem [1] is currently an active topic in both cognitive modeling and robotics. It refers to the need for grounding the symbols used to represent thoughts and beliefs in something other than just more symbols. This paper describes work on an artificial neural network that grounds symbols in sensorimotor trajectories through a local Hebb-like learning performed online. This is of interest for exploring the development of higher-level cognitive abilities in humans through experiments with robots [2]. For example the grounding of action words in the sensorimotor interaction with the world [3]. It is also relevant for assistive robots. Here it may be desirable to learn online the correlation of multimodal inputs over time [4].

An architecture with a time-delayed input structure and no hidden layers was used. Each input for each time delay was represented by a set of neurons, the number of which depended on the discretization desired. A Gaussian distribution was used to distribute activation over neurons for a given input value. Each delayed input layer was given full connectivity with a layer representing the current time inputs. A Hebb-like [5] learning rule was then used to associate all inputs in the past with all inputs in the present online. No activation was propagated during training of the neural network. Causality was thus assumed to arise implicitly from the time-delayed input structure of the neural network and its embedding in the sensorimotor loop. The learned weights were then used to predict into the future by one time delay value, by propagating the activation resulting from inputs in the past and present. Predicting low-level actuation as well as the high-level descriptive labels grounded in the low-level actuation. The ability to learn online distinguishes the approach from recent work on Multiple Time-scales Recurrent Neural Networks (MTRNN) [6].

First results from benchmarking trials on a simulated iCub humanoid robot [7] are presented. A set of Cartesian trajectories were executed with one 7 Degree Of Freedom (DOF) arm, and 4 descriptive labels where given to different phases of the trajectories. Six DOF were actuated and used as low-level inputs. Two overlapping number 8 shape movements were learned, rotated 90 degrees with respect to one another. The labels indicated the top/bottom and left/right part of the trajectories, as well as the respective movement directions. During prediction, number 8 shapes that were 25% faster and 25% smaller were also attempted. For the worst case scenario used the root mean square error for the prediction of the joint angles was kept within 7 degrees and the labels were predicted correctly more than 80% of the time. This indicates that the approach is reasonably robust to the ambiguity introduced by partially overlapping and non-exact trajectories.

Index Terms: Symbol grounding, Hebb rule, sensorimotor coordination, developmental robotics, assistive robotics, benchmarking, online learning

1. Acknowledgements

This work was supported by the European Commission FP7 Project ITALK (ICT-214668) within the Cognitive Systems and Robotics unit (FP7 ICT Challenge 2).

2. References

[1] Harnad, S., “The Symbol Grounding Problem”, Physica D: Nonlinear Phenomena, 42:335-346, 1990.

[2] Asada, M., Macdorman, K.F., Ishiguro, H., and Kuniyoshi, Y., “Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots”, Robotics and Autonomous Systems, 37:185-193, 2001.

[3] Marocco, D., Cangelosi, A., Fischer, K., and Belpaeme, T., “Grounding Action Words in the Sensorimotor Interaction with the World: Experiments with a Simulated iCub Humanoid Robot”, Frontiers in neurorobotics, 4, 2010.

[4] Balaguer, C., Jardón, A., Monje, C.A., Bonsignorio, F., Stoelen, M.F., Martínez, S. and Victores, J.G., “SULTAN: Simultaneous User Learning and Task Execution, and its Application in Assistive Robotics”, in Workshop on New and Emerging Technologies in Assistive Robotics at IROS 2011, San Francisco, CA, 2011.

[5] Hebb, D.O., “The Organization of Behavior: A neuropsychological theory”. New York: Wiley, 1949.

[6] Yamashita, Y. and Tani, J., “Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment”, PLoS Computational Biology, vol. 4, no. 11, 2008.

[7] Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., and Nori, F., “An Open-Source Simulator for Cognitive Robotics Research: the Prototype of the iCub Humanoid Robot Simulator,” in Proceedings of IEEE Workshop on Performance Metrics for Intelligent Systems Workshop (PerMIS’08), eds R. Madhavan and E. R. Messina (Washington, DC), 2008.

An Embodied View on the Development of Symbolic Capabilities and Abstract Concepts

Marek Ruciński, Francesca Stramandinoli[*]

Centre for Robotics and Neural Systems, Plymouth University
Drake Circus, PL48AA Plymouth, United Kingdom
(marek.rucinski,francesca.stramandinoli)@plymouth.ac.uk

Abstract

Until recently, research studies about symbolic representations have mainly focused on concrete language; hence, very little is known about the symbolic/conceptual system governing abstract language. In contrast to concrete entities, which can be perceived through the senses, abstract language refers to things that are intangible and that are not physically defined nor spatially constrained [1, 2]. Nevertheless, according to the embodied view of cognition, representations of such concepts are also shaped by our sensorimotor interactions with the environment. We present two cognitive robotics experiments which look at the relations between motor actions and abstract symbol manipulation capabilities. Through the first study we want to address the question whether abstract concepts can be grounded on more concrete motor primitives, while in the second experiment we want to understand if motor activities can play a facilitating role in the acquisition of conceptual competences.

A recent body of work in the neuroscience [3, 4, 5, 6] and the behavioural communities [6, 7] has revealed that words are not arbitrarily linked to their referents but they are grounded in perception, action and sensorimotor knowledge. Furthermore, different theories proposed in psychology [8, 9] state that embodiment plays an important role even in representing abstract concepts. By exploiting this knowledge, we have developed a cognitive model for the learning of compositional actions from the combination of motor primitives. In this model, sequences of linguistic inputs lead to the development of new higher-order concepts by combining words grounded on basic actions and concepts. This mechanism allows to interpret linguistic commands in terms of internal language and motor repertoire. The developed model uses recurrent neural networks. Simulation results have shown that motor primitives have different activation patterns according to the action's sequence in which they are contained. This seems to be consistent with recent neurophysiological [10] and computational neuroscience results [11]. We argue that a hierarchical organisation of concepts can be a possible account for the acquisition of abstract words in cognitive robots.

Learning to count is an example of acquisition of a conceptual competence facilitated by a motor activity. It is well established that pointing or touching plays an important role in learning the counting procedure between 2 and 6 years of age [12, 13, 14]. Importantly, there are studies which suggest that active gesture provides a unique contribution not present when gesturing is performed by another person [15]. Up to day various, not mutually exclusive hypotheses about the role of gesture have been proposed. First, gesture may facilitate coordination of producing number words (temporal aspect) and matching them with items (spatial aspect) by naturally joining the two aspects in one activity [16]. Second, gesture may help overcome limitations in cognitive resources like reducing the working memory load [13]. Third, gesture may be seen as a social learning communication channel through which the child provides its tutor with feedback on the current learning state [17]. Due to its embodied character and connection with a concrete symbolic competence, counting is an attractive topic for robotics modelling. Using this approach we seek to validate aforementioned hypotheses.

Index Terms: symbolic representations, sensorimotor knowledge, embodiment, language acquisition

1. Acknowledgements

This research has been supported by the EU project RobotDoC (235065) from the FP7 Marie Curie Actions ITN.

2. References

[1] K. Wiemer-Hastings, J. Krug, X. Xu, "Imagery, context availability, contextual constraint, and abstractness", Proceedings of the 23rd annual conference of the cognitive science society, 1134–1139, 2001.

[2] L.W. Barsalou, K. Wiemer-Hastings, "Situating abstract concepts", Grounding cognition: The role of perception and action in memory, language, and thinking, 129–163, 2005.

[3] F. Pulvermüller, M. Härle, F. Hummel, "Walking or Talking?: Behavioral and Neurophysiological Correlates of Action Verb Processing", Brain and language, 78(2): 143–168, 2001.

[4] O. Hauk, I. Johnsrude, F. Pulvermuller, "Somatotopic Representation of Action Words in Human Motor and Premotor Cortex", Neuron, 78(2): 41, 301–307, 2004.

[5] M. Tettamanti, G. Buccino, M. C. Saccuman, V. Gallese, M. Danna, P. Scifo, F. Fazio, G. Rizzolatti, S. F. Cappa, D. Perani, "Listening to Action-related Sentences Activates Fronto-parietal Motor Circuits", Journal of Cognitive Neuroscience, 17, 273–281, 2005.

[6] G. Buccino, L. Riggio, G. Melli, F. Binkofski, V. Gallese, G. Rizzolatti, "Listening to action-related sentences modulates the activity of the motor system: A combined TMS and behavioral study", Cognitive Brain Research, 24, 355–363, 2005.

[7] C. Scorolli, A. M. Borghi, "Sentence comprehension and action: effector specific modulation of the motor system", Brain Research, 1130, 119–124, 2007.

[8] L. Barsalou "Processing abstract language modulates motor system activity", The Quarterly Journal of Experimental Psychology, 61(6): 905–919, 2008.

[*] Corresponding author. Tel.: +44 1752584908. E-mail address: francesca.stramandinoli@plymouth.ac.uk (F. Stramandinoli)

[9] M. Andrews, G. Vigliocco, D. Vinson, “Integrating experiential and distributional data to learn semantic representations”, Psychological review, 116(3): 463–498, 2009.

[10] L. Fogassi, P. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, G. Rizzolatti, “Parietal lobe: from action organization to intention understanding”, Science, 308, 662, 2005.

[11] F. Chersi, S. Thill, T. Ziemke, A. Borghi, “Sentence processing: linking language to motor chains”, Frontiers in Neurorobotics, 4, 2010.

[12] T. A. Graham, “The role of gesture in children’s learning to count”, Journal of Experimental Child Psychology, 74(4): 333–355, 1999.

[13] B. Schaeffer, V. H. Eggleston, J. L. Scott, “Number development in young children”, Cognitive Psychology, 6(3): 357–379, 1974.

[14] R. Gelman, “What young children know about numbers”, Educational Psychologist, 15(1): 54–68, 1980.

[15] M. W. Alibali, A. A. DiRusso, “The function of gesture in learning to count: More than keeping track”, Cognitive Development, 14(1): 37–56, 1999.

[16] K. C. Fuson, “Children’s counting and concepts of number”, New York, NY, US, Springer-Verlag Publishing, 1988.

[17] S. Goldin-Meadow, D. Wein, C. Chang, “Assessing knowledge through gesture: Using children’s hands to read their minds”, Cognition and Instruction, 9, 201-–219, 1992.

Active Learning in a Computational Model of Word Learning

Maarten Versteegh[1,2], *Christina Bergmann*[1,2], *Louis ten Bosch*[2], *Lou Boves*[2]

[1]International Max Planck Research School for Language Sciences, Nijmegen, The Netherlands
[2]Centre for Language Studies, Radboud University Nijmegen, The Netherlands
m.versteegh@let.ru.nl

1. Abstract

Infants learning the meaning of their first words are faced with a difficult task involving multiple sources of uncertainty. Both the speech they hear from their caretakers and their visual environment can be viewed as stochastic sources of information. This paper investigates a computational model of word learning that takes a closer look at these sources of uncertainty. We present an algorithm, based on exploiting the statistical regularities in the input, and show that this is sufficient to reliably learn words from speech and vision under highly adverse circumstances. The learning strategies the algorithm implements supports conjectures about infant learning, most notably the crucial roles of statistical learning and of taking an active approach to in processing the environmental input.

We identify two types of uncertainty. The first relates to the fact that the infant must discover recurrent acoustic stretches from a highly variable speech stream and eventually form meaningful combinations of these building blocks. Research has shown that the ability of young learners to detect patterns in spoken language is at least partly based on the use of the statistical properties of the speech signal.

The second type of uncertainty is due to possible inconsistencies between patterns in the speech stream and objects in the scene surrounding the learner. Any pairing of a spoken word and a communicative scene presents the learner with a large number of possible referents. When a caretaker utters a sentence pertaining to an object in the visual environment, it may thus be highly ambiguous from the viewpoint of the learner. Infants use statistical evidence across many individually ambiguous word-scene pairings to discover cooccurrences of auditory and visual information, thus resolving the ambiguity. They are also able to detect when form-referent pairings do not match their previous experience and will actively attempt to resolve these perceived mismatches by aligning a different visual referent with a given auditory form.

We present a computational model of word learning that incorporates both cross-situational statistical learning as well as the corrective process after detection of a mismatch, which has an analogy in active learning procedures in machine learning techniques. Learning in our model consists of multi-modal pattern finding, combining information extracted from speech utterances and images with the goal of both identifying the basic building blocks of language and forming word-referent pairings from situations of referential ambiguity.

Results from the model show the advantages of combining cross-situational statistics and active learning. We show that cross-situational statistics by itself provides the learner with a high level of robustness against referential ambiguity. Incorporating active learning, we then show that this improves the model's robustness even further, where the gain is proportional to the level of ambiguity in the input. Our model shows the importance of the active learning procedure attested in infants, in reliably learning words under highly uncertain conditions, such as the conditions an infant faces when learning his or her first words.

Innate Neonatal Face Preference - An Embodied Phenomenon?

Nick Wilkinson, Francesco Rea, Katrin Lohan, Giorgio Metta [1], Gustaf Gredeback [2]

[1] Robotics, Brain and Cognitive Sciences, Instituto Italiano di Tecnologia, Genova, Italy
[2] BabyLab, Dept. of Psychology, Uppsala University, Sweden
nicholas.wilkinson@iit.it

Abstract

Faces appear to attract the attention of humans in an extra-ordinary way from birth [1] and throughout the rest of life. The currently dominant view in developmental psychology is that infants have some form of innate internal representation which preferentially matches face like stimuli. The debate surrounding this issue has become a canonical stage for issues such as the nature of innateness and the development of specialisation in the brain. This poster/presentation will introduce initial results from modelling studies of innate face preference conducted on the iCub humanoid robot.

The hypothesis under investigation is that an innate internal representation is unnecessary, and possibly insufficient for basic face preference in real world conditions. Instead we exploit the fact that the visual system is mounted on a face to make faces "special" to the robot. This fact enables the emergence of a visual "affordance", or sensorimotor invariance in the relationship between the viewer and the world, which we recently described in [2]. On this view, the reason why pairs of eyes pop out in visual scenes is that we see things with a pair of eyes. Other facial texture can also play a role. Eyebrows are known to be very important in visual-facial perception. We are testing this hypothesis in both real interactions with humans and with stimuli used to study human neonates. Performance is compared to a model based on the CONSPEC hypothesis of an innate face template [3].

A well defined theoretical framework has allowed us to make predictions regarding neonate face preference which we hope will motivate studies on neonates [4]. In addition to offering a way out of a somewhat unproductive 20 year debate in the literature on newborn face preference as to the form of this representation, this theoretical framework and the results it generates will bear on issues of innateness and the nature of cognitive and behavioural inheritance.

This study also exemplifies wider debates in cognitive science. Does cognition occur primarily in internal models of self and world which are somehow matched to the sensorimotor flux and used to control that flux from outside it? Or is cognition largely embedded in the embodiment of the agent and the form of the sensorimotor flux itself? Can the causative structure of behavioural and cognitive events and processes be functionally localised in space, such within a brain region? Or is a systems approach required to understand what may be fundamentally distributed networks of causality without a central controller? What is the appropriate way to use abstraction and concreteness in cognitive modelling? We will present our results in the light of their relevance for these broader discussions within the field, in the hope of showing how bridges between developmental psychology and developmental robotics may be built, a process crucial to the ongoing success of developmental cognitive science as an inter-disciplinary project.

1. Acknowledgements

This research is being supported by the EU project Robot-Doc under 25065 from the 7th Framework Programme, Marie Curie Action ITN.

2. References

[1] Johnson, M., Dziurawiec, S., Ellis, H., and Morton, J. (1991). Newborns preferential tracking of faces and its subsequent decline.
Cognition, 40(1-2):1–19.

[2] Wilkinson, N., Metta, G., Rea, F. and Gredeback, G. (submitted). Visual morphological resonance: Direct, depth robust binocular sensing of real world spatial scale. Journal of Vision.

[3] J. Morton and M. Johnson. Conspec and conlern: A two-process theory of infant face recognition. Psychological Review, 98(2):164–181, April 1991.

[4] Wilkinson, N., Metta, G., and Gredeb¨ack, G. (2011a). Interfacial relations: Binocular geometry when eyes meet. Proceedings of 2nd International Conference on Morphological Computation, Venice, p.p 69–71.