A Model of the Visual Dorsal Pathway for Computing Coordinate Transformations: An Unsupervised Approach

Flavio Mutti[1], Hugo Gravato Marques[2], and Giuseppina Gini[3]

[1] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy
mutti@elet.polimi.it
[2] University of Zurich, Institute for Informatics, AI Lab, Zurich 8050, Switzerland
hgmarques@gmail.com
[3] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy
gini@elet.polimi.it

Abstract. In humans, the problem of coordinate transformations is far from being completely understood. The problem is often addressed using a mix of supervised and unsupervised learning techniques. In this paper, we propose a novel learning framework which requires only unsupervised learning. We design a neural architecture that models the visual dorsal pathway and learns coordinate transformations in a computer simulation comprising an eye, a head and an arm (each entailing one degree of freedom). The learning is carried out in two stages. First, we train a posterior parietal cortex (PPC) model to learn different frames of reference transformations. And second, we train a head-centered neural layer to compute the position of an arm with respect to the head. Our results show the self-organization of the receptive fields (gain fields) in the PPC model and the self-tuning of the response of the head-centered population of neurons.

1 Introduction

A coordinate transformation (CT) is the capability to compute the position of a point in space with respect to a specific frame of reference (FoR), given the position of the same point in another FoR. The way the mammal brain solves the problem of CTs has been largely studied. Nowadays it is fairly well known from lesion studies [10] that the main area involved in this type of computation is the Posterior Parietal Cortex [1] [6].

The computation of CT seems to exploit two widespread properties of the brain, namely, population coding [7], and gain modulation [2] [9]. Population coding is a general mechanism used by the brain to represent information both to encode sensory stimuli and to drive the body actuators. The responses of an ensemble of neurons encode both sensory or motor variables in such a way that can be further processed by the next cortical areas, e.g. motor cortex. There are at least two main advantages of using a population of neurons to encode information: robustness to noise [7] and the capability to approximate nonlinear

A. Chella et al. (Eds.): Biologically Inspired Cognitive Architectures 2012, AISC 196, pp. 239–246.
springerlink.com

transformations [8]. Gain modulation is an encoding strategy for the amplitude of the response of a single neuron that can be scaled without changing the response selectivity of the neuron. This modulation, also know as *gain field*, can arise from either multiplicative or nonlinear additive responses [2] [3].

Several computational models of the PPC address the problem of CTs using three-layer feed-forward neural networks (FNNs) [11], recurrent neural networks (RNNs) [9], or basis functions (BFs) [8]. The FNNs and the BFs models are trained with supervised learning technique whereas the RRNs model uses a mix of supervised and unsupervised approaches to train the neural connections, encoding multiple FoRs transformation in the output responses.

It is worth noting that gain modulation plays an important role in the computation of the coordinate transformations but it is still unclear if this property emerges in the cortex from statistical properties of the afferent (visual) information. Recently, [5] shows evidence to support that gain fields can arise through the self-organization of an underling cortical model called Predictive Coding/Biased Competition (PC/BC) . It demonstrates that the gain modulation mechanism arises through the competition of the neurons inside the PC/BC model, and comments on the feasibility of such system to compute CTs.

These computational models of the PPC could be particularly suitable for the robotics community to solve the well-known problem of CT. In the recent past, an architecture was proposed that explicitly includes a PPC model composed by a set of radial basis functions trained with supervised learning techniques [4]. However, most of the approaches in robotics address the problem of FoR transformation inside the more general sensorimotor mapping approach, without explicitly exploit the features of PPC models [6].

Following these ideas, we present a biologically inspired model for CTs. First we describe the training of a PPC model with an unsupervised learning approach; and second we introduce the computation of the arm position with respect to the head position. We hypothesise that gain modulation mechanisms can emerge in the PPC neurons, and that basis functions, encoding parallel CTs, can emerge after the training phase. The main contributions of this paper are: first to show an unsupervised approach to the learning of sensorimotor mapping; second to exploit the synergy between a biologically inspired neural network and the population coding paradigm; and third to introduce quantitative evaluation of the sensorimotor mapping performance.

This paper is organized as follows. In Section 2 we design the neural network model that performs the implicit sensorimotor mapping, in Section 3 we present the performed experiments and in Section 4 we derive our conclusions.

2 Model

In this section we present the neural model used for computing CTs between an arm and the head FoR. We define a simple mechanical structure composed by an eye, a head and an arm with the same origin. We assume the same origin because the fixed translations among these FoRs can be neglected due to their

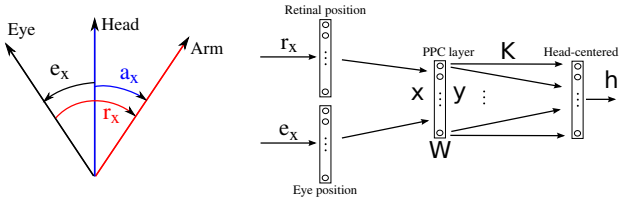

Fig. 1. (Left pane) Body definition composed by an eye, a head and an arm with the same origin. (Right pane) Neural Network model. The first layer encodes the sensory information into a neural code, the second layer models the posterior parietal cortex and it performs the multi sensory fusion and the third layer encodes the arm position with respect to the head frame of reference.

known contribution in the computation of the CTs (Figure 1, left pane). The eye position is defined by the angle e_x with respect to the head FoR, the retinal stimuli position of the arm is defined by the angle r_x with respect to the eye FoR; the head-centered position of the arm is defined by $a_x = r_x + e_x$ angle (see Figure 1, left pane). The neural architecture is divided in three layers: the first is composed by two populations of neurons which represent the information of the retinal position of the arm, r_x, and the eye position with respect to the head, e_x. The second is composed by PPC population of neurons that encode the position of the arm in different FoRs. The third is a population of neurons that encodes the arm position with respect to the head FoR.

The first layer of the network model receives as input the eye position with respect to the head FoR (e_x) as well as the arm position with respect to the retinal FoR (r_x). We define the eye angle e_x in degrees and the retinal position of the target r_x both in degrees and in pixels (see Section 3). These *numeric* values are encoded in a *population coding* paradigm, where a given sensor value is represented as a population of neural responses [8]. The response of a population neuron is defined as a Gaussian as follows:

$$n_i = A \exp\left(-\frac{(v-\mu_i)^2}{2\sigma^2}\right) \tag{1}$$

where n_i is the response of neuron i, μ_i is the neuron preferred sensor value, v is the *numeric* input angle (in degrees), and σ is the standard deviation of the Gaussian.

The PPC layer is based on the Predictive Coding/Biased Competition model (PC/BC) proposed in [5]. The model is trained with a unsupervised approach that is based on Hebbian learning. The system equations are:

$$s = x \oslash (\epsilon_2 + \hat{W}^T y) \qquad y = (\epsilon_1 + y) \otimes W\, s \tag{2}$$

where s is the internal state of the PC/BC model, $x = [n_0, \ldots, n_{L+M}]$ is the neural population input vector defined by M retinal neurons and L neurons encoding the eye position, W is the weight matrix, $\hat{W}$ is the normalized W,

y is the output vector of the PPC layer, and ϵ_1, ϵ_2 are constant parameters; $\oslash$ and $\otimes$ indicate element-wise division and multiplication respectively. These equations are evaluated iteratively for a certain number of time steps; after a certain period of time, y and e values reach a steady state. The internal state s is self-tuned and represents the similarity between the input vector x and the reconstruction of the input $\hat{W}^T y$ ($s \approx 1$ indicates an almost perfect reconstruction).

The unsupervised training rule is given by:

$$W = W \otimes \{1 + \beta\, y(s^T - 1)\} \tag{3}$$

where β is the learning rate. This training rule minimizes the difference between the population responses x and the input reconstruction $W^T y$; the weights increase for $s > 1$ and decrease for $s < 1$.

Let's consider the output vector $y = [y_0, \ldots, y_T]$ as the population responses of the PPC model. Each neuron response y_i should be compatible with the gain modulation paradigm, according to the experimental results of [5], in such a way that the response exhibit a multiplicative behaviour, as a function of both eye and retinal positions. The weight matrix, which encodes the response properties, is *internal* to the PPC model and the training phase is independent with respect to the unsupervised training phase that will involve the head-centered network layer.

The population of neurons associated to the head-centered frame of reference deals with the estimation of the arm position a_x given the eye angle e_x and the projection of the arm in the retina r_x. The synapses between the PPC layer and head-centered frame are trained with an Hebbian learning, taking into account the arm position, a_x. Estimating a_x means identifying the maximum response inside a population of neuron that encodes a_x with the population coding paradigm. The head-centered population responses are given by $h = K\, y$, where y is the output vector of the PPC model, K is the weight matrix representing the fully-connected synapses between the PPC model and the head-centered layer and h is a vector that contains the population responses, encoding the estimated a_x. The dimension of h depends on the granularity of the a_x encoding. The training phase is performed using Hebbian learning:

$$K = K + \delta\, h\, p_a^T \qquad \delta = \frac{1}{N} \tag{4}$$

where p_a is the vector that contains the proprioceptive population responses encoding a_x, and δ is the learning rate depending by N, the number of samples.

3 Experimental Results

In this section we present the results obtained in two experiments; in the first experiment we train and analyse the network where either the eye angle and the retinal position are encoded in degrees and in the second experiment we introduce a simple camera model to encode the retinal information in pixels.

The training phase is carried out in two steps: (1) train the PPC layer and (2) train the head-centered layer. The PPC layer is trained following the method described in Section 2 (Equation 3) and the synapses between the PPC and head-centered layer are trained using Hebbian learning as described in Section 2 (Equation 4).

In the first experiment, we encode both r_x and e_x in degrees and for the PPC layer, we use the same parameter values as in [5]. The y consists of a 64-element vector with a range for the sensors values defined as follows: $r_x \in [-30°, 30°]$, $e_x \in [-30°, 30°]$, $a_x \in [-60°, 60°]$. We encode the sensory input with a population of 61 neurons with a gaussian response and with a standard deviation $\sigma = 6°$. The σ value is chosen taking into account the experiment described in [5] whereas the neuron preferred values are equally distributed inside the range value.

After the training of the PPC layer, we train the head-centered layer with a population of 121 neurons, defining h as a 121-elements vector. With 121 neurons representing a_x the coding resolution (1°) can be analytically derived. The standard deviation of the neuron responses associated to the arm position a_x is equal to 6°. The population of neurons, encoding the proprioceptive position of the arm, has the same number of neurons of the head-centered layer (121) and each neuron has the same standard deviation (6°). The proprioceptive responses vector p_a drives the Hebbian learning for the head-centered neural layer (Equation 4).

Figure 2 shows the analysis of the trained network: top left pane shows the responses of the trained network that represents the arm position a_x with respect to the head frame of reference. The red solid line represent the response for $a_x = 20°$, the green dashed-dot line represent the response for $a_x = 0°$ and the blue dash line represent the response for $a_x = -20°$. Top right pane shows the error distribution (in degrees) of the estimated a_x with respect to the arm position, the eye position and the retinal position respectively. The solid lines represent the mean error and the dashed lines represent the standard deviation limits. The error distributions are quite similar and, in general, the error is quite low with a global mean error equal to 1.93° with a global standard deviation equal to 1.89°. Bottom left pane shows the receptive field after the training phase of the PPC layer: it is shown the global shape of the gain modulation. As expected, the curves shapes are compatible with the gain modulation paradigm, supporting the evidence that an unsupervised method can effectively learn a multiplicative behaviour. Bottom right shows the contours at half the maximum response strength for the 64 PPC neurons: it is worth noting the different color of the contours that represent different level of activations. A qualitative analysis points out that the population responses are stronger where the correspondent neuron receptive fields are slightly overlapped. Moreover, the PPC neurons receptive fields almost cover the whole subspace in the e_x-r_x plane, indicating that there is at least a neuron firing for each combination of e_x and r_x.

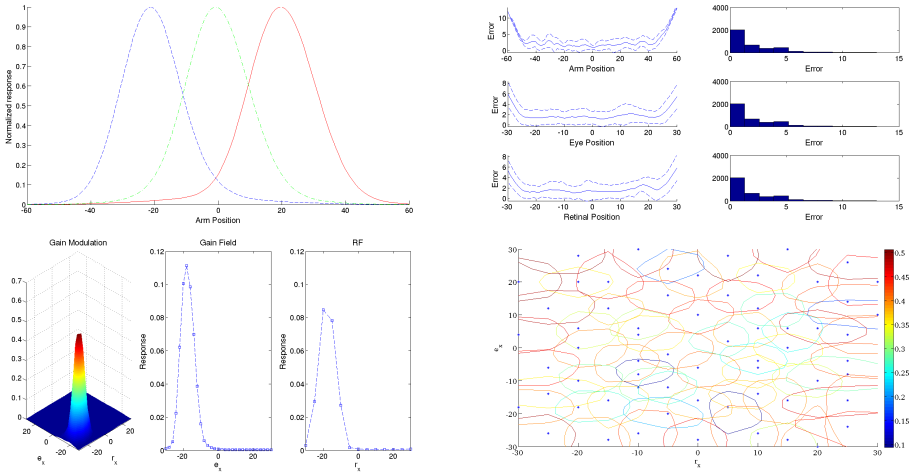

Fig. 2. Experimental results with r_x in degrees.(Top left) It shows the responses of the trained network that represents the arm position a_x with respect to the head frame of reference for $-20°$,$0°$ and $20°$, respectively. (Top right) The error distribution (in degrees) of the estimated a_x with respect to the arm position, the eye position and the retinal position respectively. The solid lines represent the mean error and the dashed lines represent the standard deviation limits. (Bottom left) It represents a receptive field after the training phase of the PPC layer. (Bottom right) Contours at half the maximum response strength for the 64 PPC neurons.

In the second experiment we investigate a more realistic scenario where the retinal position is a pixel position in the image plane. We just consider only the horizontal component of the image position of the arm. To compute the real a_x value we exploit some geometrical constraints, given by the camera model. In the specific:

$$a_x = e_x + \tan^{-1}\left(\frac{r_x}{f}\right) \qquad [°] \tag{5}$$

where r_x is the retinal position in pixels of the arm and f is the focal length of the camera. For our purposes, we choose a focal length equal to 120 pixels that represents a camera with a open lens of about 140°.

The PPC layer contains 64 neurons but the input range are $r_x \in [-320, 320]$, $e_x \in [-25°, 25°]$, $a_x \in [-94°, 94°]$ where r_x is defined in pixels; it follows that we suppose to have a image plane with an horizontal component that has a size equal to 641 pixels. The range of a_x follows the maximum value that the a_x can reach. We use 101 and 51 neurons to represent r_x and e_x, respectively. We use the standard deviation σ of gaussian representing r_x equal to 60 pixels. Also in this case the standard deviation of the proprioceptive neurons encoding a_x is equal to 6°.

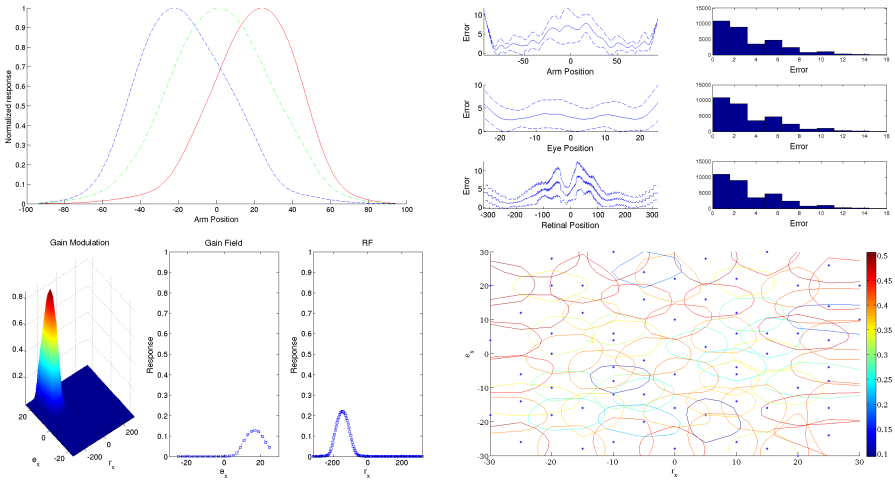

Fig. 3. Experimental results with r_x in pixels.(Top left) It shows the responses of the trained network that represents the arm position a_x with respect to the head frame of reference for $-20°$,$0°$ and $20°$, respectively. (Top right) The error distribution (in degrees) of the estimated a_x with respect to the arm position, the eye position and the retinal position respectively. The solid lines represent the mean error and the dashed lines represent the standard deviation limits. (Bottom left) It represents a receptive field after the training phase of the PPC layer. (Bottom right) Contours at half the maximum response strength for the 64 PPC neurons.

Figure 3 shows the results from the analysis of the trained network. The overall performance is lower than that obtained in the previous experiments: the top right pane shows the error distribution with respect to the arm, eye and retinal position, respectively. In this set of experiments, during the PPC learning, the system is able to learn PPC receptive fields that are compatible, in a qualitative way, with the gain modulation principle (see Figure 3, bottom left pane). The bottom right pane shows the receptive fields distribution in the space r_x-e_x where we have the same qualitative features of the previous experiment. The estimation of a_x has a global mean error equal to $3.36°$ with a global standard deviation equal to $2.90°$.

4 Conclusions

This work described an unsupervised approach to learn coordinate transformations. The results show how the system is able to correctly compute the position of a target with respect to the stable head frame of reference knowing only the projection of the target onto the image plane and the eye position with respect to the head. Further experiments are foreseen to validate the model for more realistic scenarios, trying the method on a real robotic system and extending the model for complex physical architectures.

Acknowledgments. This work has been possible thanks to the partial support provided by the 2010-2012 Italian-Korean bilateral project (ICT-CNR and KAIST). The authors gratefully acknowledge the contribution the NVIDIA Academic Partnership for providing GPU computing devices. The research leading to these results has received funding from the European Community's 7th Framework Programme FP7 no. 207212 - eSMCs.

References

1. Andersen, R.A., Cui, H.: Intention, action planning, and decision making in parietal-frontal circuits. Neuron 63, 568–583 (2009)
2. Andersen, R.A., Essick, G.K., Siegel, R.M.: Encoding of spatial location by posterior parietal neurons. Science 230, 456–458 (1985)
3. Brozovic, M., Abbott, L.F., Andersen, R.A.: Mechanism of gain modulation at single neuron and network levels. Journal of Computational Neuroscience 25, 158–168 (2008)
4. Chinellato, E., Antonelli, M., Grzyb, B.J., del Pobil, A.P.: Implicit sensorimotor mapping of the peripersonal space by gazing and reaching. IEEE Transactions on Autonomous Mental Development 3(1), 43–53 (2011)
5. De Meyer, K., Spratling, M.W.: Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function. Neural Computation 23, 1536–1567 (2011)
6. Hoffmann, M., Marques, H., Arieta, A., Sumioka, H., Lungarella, M., Pfeifer, R.: Body schema in robotics: A review. IEEE Transactions on Autonomous Mental Development 2(4), 304–324 (2010)
7. Knill, D.C., Pouget, A.: The bayesian brain: the role of uncertainty in neural coding and computation. Trends in Neurosciences 27(12), 712–719 (2004)
8. Pouget, A., Sejnowski, T.J.: Spatial transformations in parietal cortex using basis functions. Journal of Cognitive Neuroscience 9(2), 222–237 (1997)
9. Salinas, E., Abbott, L.F.: Coordinate transformations in the visual system: How to generate gain fields and what to compute with them. Progress Brain Research 130, 175–190 (2001)
10. Shadmehr, R., Krakauer, J.W.: A computational neuroanatomy for motor control. Experimental Brain Research 185, 359–381 (2008)
11. Xing, J., Andersen, R.A.: Models of the posterior parietal cortex which perform multimodal integration and represent space in several coordinate frames. Journal of Cognitive Neuroscience 12(4), 601–614 (2000)