Adaptive Behaviour, doi: 10.1177/1059712315607363.

A Neural Model of Binocular Saccade Planning and Vergence Control

Wasif Muhammad and Michael W. Spratling
King's College London, Department of Informatics, London. UK.

Abstract

The human visual system uses saccadic and vergence eye movements to foveate visual targets. To mimic this aspect of the biological visual system the PC/BC-DIM neural network is used as an omni-directional basis function network for learning and performing sensory-sensory and sensory-motor transformations without using any hardcoded geometric information. A hierarchical PC/BC-DIM network is used to learn a head-centred representation of visual targets by dividing the whole problem into independent subtasks. The learnt head-centred representation is then used to generate saccade and vergence motor commands. The performance of the proposed system is tested using the iCub humanoid robot simulator.

Keywords: basis function networks; sensory-sensory transformations; sensory-motor control; saccades; eye movements; neural networks; function approximation; iCub; vergence

1 Introduction

Sensory-sensory and sensory-motor transformations are fundamental to many cognitive and behavioural abilities in both animals and robots. For example, sensory information about an object's location might be encoded in retinotopic coordinates (for vision), head-centred coordinates (for audition), and body-centred coordinates (for touch). Hence, knowing that the object that is seen is the same as the object that is heard or touched relies on being able to find correspondences between different sensory coordinate systems. Similarly, moving the eyes to look at the hand requires the transformation of hand position information encoded in terms of arm joint angles into the corresponding eye position signals, while controlling the hand to reach for a location that has been identified visually requires a coordinate system transformation in the reverse direction.

Traditionally, sensory-sensory and sensory-motor transformations in robotics have been achieved using hard-coded, kinematic, models. An alternative approach is to perform such transformations using neural networks. This approach might be used to learn the transformation when insufficient information is available to derive the kinematic equations (Hoffmann et al., 2010). It might also be preferred in order to more closely imitate biological mechanisms of sensory-motor control. Basis function networks are a popular neural network architecture for performing sensory-sensory and sensory-motor coordination in robots (Kim et al., 2005; Marjanović et al., 1996; Meng and Lee, 2007, 2008; Molina-Vilaplana et al., 2004; Sun and Scassellati, 2005; Weber et al., 2007; Zhang et al., 2005) and as models of brain function (Chinellato et al., 2011; De Meyer and Spratling, 2013; Deneve et al., 1999, 2001; Deneve and Pouget, 2003; Pouget et al., 2002; Pouget and Sejnowski, 1994, 1997; Pouget and Snyder, 2000; Salinas and Abbott, 1995; Salinas and Sejnowski, 2001; Spratling, 2009; van Rossum and Renart, 2004). Basis function networks can approximate any linear or nonlinear mapping (Broomhead and Lowe, 1988; Park and Sandberg, 1991; Schilling et al., 2001), but for simplicity, a very simple linear example is shown in Fig. 1. The basis function approach splits the problem into two sub-problems: a layer of basis function nodes, with nonlinear activation functions, encode possible combinations of sensory input signals, and a linear readout of the responses of these basis functions is used to produce the output.

However, current implementations of basis function networks suffer from the following issues.

- Mapping is uni-directional. For example, the network shown in Fig. 1 can infer c given a and b, but it can not infer a given b and c. Deneve et al. (2001) have proposed a basis function neural network with attractor dynamics that overcomes this limitation, but which still suffers from the problems described below.

- Scales poorly with problem size. The required number of basis function neurons increases exponentially with the number of input variables (Deneve and Pouget, 2003; Pouget and Sejnowski, 1997). For example, in Fig. 1 both input variables range over five possible values, and the number of basis functions used to represent all possible combinations of these values is 5^2. If the output was a function of three input variables (each ranging over five values) then the number of basis function neurons required would increase to 5^3. More generally, if there are m input variables, and n is a measure of the precision with which each variable is to be represented (or the range of values that is to be represented with a fixed precision), then the number of basis functions required to represent the mapping is proportional to n^m. Given that in many tasks (*e.g.*,

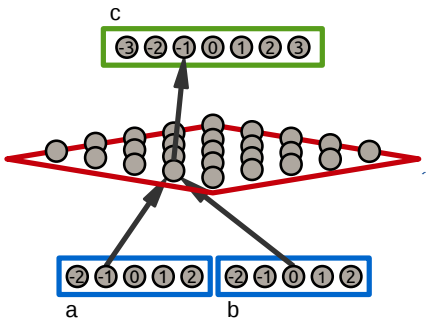


Figure 1: A simple basis function network for performing a mapping between two input variables (a and b) and an output variable (c). In this simple example, the mapping is linear, such that $c = a+b$. The values of the input variables are encoded by the activation patterns in two populations of neurons, and the value of the output is encoded by the firing of a third population of neurons. By using populating coding the inputs and outputs can represent any continuous value, rather than just discrete values as suggested by the figure. While representations of the current variable values are encoded by neural activity, the mapping between values is encoded in the connections (the "weights") between the neurons in the network. The mapping from the inputs to the output is mediated by a hidden layer of neurons, the basis function population. Each basis function node has weights that allow it to represent a possible combination of input values. Each neuron in the output layer has non-zero weights to those basis function neurons that represent the same output value (*e.g.*, an output neuron representing the value $c = -1$, would receive input from the basis function representing the combination of inputs $a = -1$ and $b = 0$ and would also receive input from the basis function representing the inputs $a = -2$ and $b = 1$, *etc.*). This network could be interpreted as performing a sensory-sensory transformation that maps retinocentric coordinates to head-centred coordinates for a very simple system with a one-dimensional retina in which eye position generates a horizontal shift of the retina along its axis. In this case, the current value of a represents the position of an object on the retina, the value of b represents eye position, and the value of c represents the corresponding head-centred position of the object.

the visual control of an arm) the correct mapping may depend on many variables (*e.g.*, the retinal location of the target, the orientation of the eyes in their sockets, the posture of the head, the posture of the upper torso) m is large and using a single basis function network is not a tractable solution for most real-world applications. To resolve this issue it is possible to decompose a transformation into multiple steps, and implement each step using a separate basis function network (Pouget et al., 2002)[a]. However, this has rarely been successful in practice (although exceptions include Chinellato et al., 2011; Meng and Lee, 2008).

- Represents only one stimulus. Standard basis function networks require that all inputs correspond to a single sensory stimulus (Pouget et al., 2002), and hence, are unable to calculate a mapping if multiple objects are present simultaneously.

In this article an alternative neural network model is proposed for performing sensory-sensory and sensory-motor transformations which overcomes all these issues. To demonstrate this method we apply it to the control of eye movements in the iCub Humanoid Robot Simulator (Metta et al., 2008; Tikhanoff et al., 2008). Specifically, we show that the new method can be used to learn a hierarchy of basis function-like networks for transforming retinotopic sensory information into a head-centred representation of visual space. We further show that this head-centred representation can be used to control movements of both eyes in order to generate saccadic and vergence movements[b].

[a]There is both psychophysical and neurophysiological evidence to suggest that a similar strategy is employed to perform sensory-sensory and sensory-motor transformations in the brain, leading to the existence of multiple coordinate systems along the dorsal pathway of the cortical visual system (Battaglia-Mayer et al., 2003; Blangero, 2008; Marzocchi et al., 2008; McGuire and Sabes, 2009; Pertzov et al., 2011): neural representations are arranged in a retinotopic map in primary visual cortex (V1) and middle temporal area (MT; Hartmann et al., 2011), while regions of the parietal cortex contain representations of space in head-centred (Andersen et al., 1985; Duhamel et al., 1997), body-centred (Brotchie et al., 1995), object-centred (Chafee et al., 2007) and world-centred (Snyder et al., 1998) coordinates.

[b]Saccades are rapid movements of both eyes in the same direction that are used to bring salient visual information onto the most sensitive part of the retina called the fovea. Vergence moves the eyes in opposite directions in order to bring visual targets at different depths onto the fovea of both eyes.

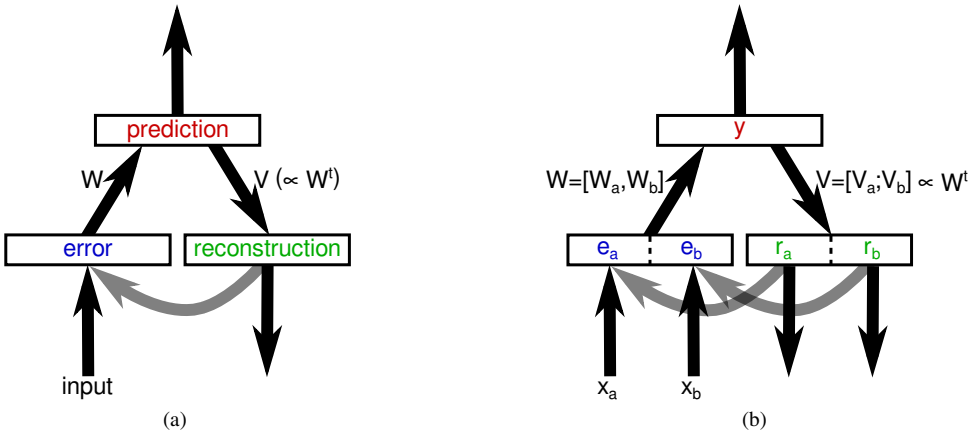


Figure 2: (a) A single processing stage in the PC/BC-DIM neural network architecture. Rectangles represent populations of neurons and arrows represent connections between those populations. The population of prediction neurons constitute a model of the input environment. Individual neurons represent distinct causes that can underlie the input (*i.e.*, latent variables). The belief that each cause explains the current input is encoded in the activation level, $\mathbf{y}$, and is used to reconstruct the expected input given the predicted causes. This reconstruction, $\mathbf{r}$, is calculated using a linear generative model (see equation 1). Each column of the feedback weight matrix $\mathbf{V}$ represents an "elementary component", "basis vector", or "dictionary element", and the reconstruction is thus a linear combination of those components. Each element of the reconstruction is compared to the corresponding element of the actual input, $\mathbf{x}$, in order to calculate the residual error, $\mathbf{e}$, between the predicted input and the actual input (see equation 2). The errors are subsequently used to update the predictions (via the feedforward weights $\mathbf{W}$, see equation 3) in order to make them better able to account for the input, and hence, to reduce the error at subsequent iterations. The responses of the neurons in all three populations are updated iteratively to recursively calculate the values of $\mathbf{y}$, $\mathbf{r}$, and $\mathbf{e}$. The weights $\mathbf{V}$ are the transpose of the weights $\mathbf{W}$, but are normalised to so that the maximum value of each column is unity. The activations of the prediction neurons or the reconstruction neurons may be used as inputs to other PC/BC-DIM processing stages. The inputs to this processing stage may come from the prediction neurons of this or another processing stage, or the reconstruction neurons of another processing stage, or may be external, sensory-driven, signals. The inputs can also be a combination of any of the above. (b) When inputs come from multiple sources, it is convenient to consider the population of error neurons to be partitioned into sub-populations which receive these separate sources of input. As there is a one-to-one correspondence between error neurons and reconstruction neurons, this means that the reconstruction neuron population can be partitioned similarly.

2 Methods

2.1 The PC/BC-DIM Algorithm

All experiments reported here were performed using the PC/BC-DIM algorithm. PC/BC-DIM is a version of Predictive Coding (PC; Rao and Ballard, 1999) reformulated to make it compatible with Biased Competition (BC) theories of cortical function (Spratling, 2008a,b) and that is implemented using Divisive Input Modulation (DIM; Spratling et al., 2009) as the method for updating error and prediction neuron activations. DIM calculates reconstruction errors using division, which is in contrast to other implementations of PC that calculate reconstruction errors using subtraction (Huang and Rao, 2011). PC/BC-DIM is a hierarchical neural network. Each level, or processing stage, in the hierarchy is implemented using the neural circuitry illustrated in Fig. 2a. A single PC/BC-DIM processing stage thus consists of three separate neural populations. The behaviour of the neurons in these three populations is determined by the following equations:

$$\mathbf{r} = \mathbf{V}\mathbf{y} \tag{1}$$

$$\mathbf{e} = \mathbf{x} \oslash (\epsilon_2 + \mathbf{r}) \tag{2}$$

$$\mathbf{y} \leftarrow (\epsilon_1 + \mathbf{y}) \otimes \mathbf{W}\mathbf{e} \tag{3}$$

Where $\mathbf{x}$ is a (m by 1) vector of input activations, $\mathbf{e}$ is a (m by 1) vector of error neuron activations; $\mathbf{r}$ is a (m by 1) vector of reconstruction neuron activations; $\mathbf{y}$ is a (n by 1) vector of prediction neuron activations; $\mathbf{W}$ is

a (n by m) matrix of feedforward synaptic weight values; $\mathbf{V}$ is a (m by n) matrix of feedback synaptic weight values; ϵ_1 and ϵ_2 are parameters; and $\oslash$ and $\otimes$ indicate element-wise division and multiplication respectively. For all the experiments described in this paper ϵ_1 and ϵ_2 were both given the value 1×10^{-9}. Parameter ϵ_1 prevents prediction neurons becoming permanently non-responsive. It also sets each prediction neuron's baseline activity rate and controls the rate at which its activity increases when an input stimulus is presented within its receptive field (RF). Parameter ϵ_2 prevents division-by-zero errors and determines the minimum strength that an input is required to have in order to effect prediction neuron response. As in all previous work with PC/BC-DIM, these parameters have been given small values compared to typical values of $\mathbf{y}$ and $\mathbf{x}$, and hence, have negligible effects on the steady-state activity of the network. The matrix $\mathbf{V}$ is equal to the transpose of the $\mathbf{W}$, but each column is normalised to have a maximum value of one. Hence, the feedforward and feedback weights are simply rescaled versions of each other. Given that the $\mathbf{V}$ weights are fixed to the $\mathbf{W}$ weights there is only one set of free parameters, $\mathbf{W}$, and references to the "synaptic weights" refer to the elements of $\mathbf{W}$. Here, as in previous work with PC/BC-DIM only non-negative weights, inputs, and activations are used. Initially the values of $\mathbf{y}$ are all set to zero, although random initialisation of the prediction node activations can also be used with little influence on the results. Equations 1, 2 and 3 are then iteratively updated with the new values of $\mathbf{y}$ calculated by equation 3 substituted into equation 1 and 3 to recursively calculate the neural activations. This iterative process was terminated after 150 iterations in all the experiments reported here.

The values of $\mathbf{y}$ represent predictions of the causes underlying the inputs to the network. The values of $\mathbf{r}$ represent the expected inputs given the predicted causes. The values of $\mathbf{e}$ represent the residual error between the reconstruction, $\mathbf{r}$, and the actual input, $\mathbf{x}$. The full range of possible causes that the network can represent are defined by the weights, $\mathbf{W}$ (and $\mathbf{V}$). Each row of $\mathbf{W}$ (which correspond to the weights targeting an individual prediction neuron) can be thought of as a "basis vector" or "elementary component" or "preferred stimulus", and $\mathbf{W}$ as a whole can be thought of as a "dictionary" or "codebook" of possible representations, or a model of the external environment (Spratling, 2012, 2014). The activation dynamics described above result in the PC/BC--DIM algorithm selecting a (typically sparse) subset of active prediction neurons whose RFs (which correspond to basis functions) best explain the underlying causes of the sensory input. The strength of activation reflects the strength with which each basis function is required to be present in order to accurately reconstruct the input. This strength of response also reflects the probability with which that basis function (the preferred stimulus of the active prediction neuron) is believed to be present, taking into account the evidence provided by the input signal and the full range of alternative explanations encoded in the RFs of the whole population of prediction neurons.

When inputs come from multiple sources it is convenient to consider the vector of input signals, $\mathbf{x}$, the vector of error neuron activations, $\mathbf{e}$, and the vector of reconstruction neuron responses, $\mathbf{r}$, to be partitioned into multiple parts corresponding to these separate sources of input (see Fig. 2b; Spratling, sub). Each partition of the input will correspond to certain columns of $\mathbf{W}$ (and rows of $\mathbf{V}$). While it is conceptually convenient to think about separate partitions of the inputs, neural populations and synaptic weights, it does not in any way alter the mathematics of the model. In equations 1, 2 and 3, $\mathbf{x}$ is a concatenation of all partitions of the input, $\mathbf{e}$ and $\mathbf{r}$ represent the activations of all the error and reconstruction neurons; and $\mathbf{W}$ and $\mathbf{V}$ represent the synaptic weight values for all partitions.

2.2 Performing Transformations with a PC/BC-DIM Network

As described above, the prediction neurons in a PC/BC-DIM network behave like basis function neurons. Figure 3 illustrates how this can be exploited to perform a simple mapping from two input variables to an output variable, analogous to the task performed by the basis function network shown in Fig. 1. If a sub-set of the prediction neurons represent combinations of inputs that correspond to the same value of the output, then it is necessary to "pool" the responses from this sub-set of prediction neurons to produce this output whenever one of these combinations is presented to the inputs. Figure 3 shows two ways in which this can be implemented. The first method (Fig. 3a) involves using a separate population of pooling neurons that are activated by the responses of the prediction neurons. This method has been used in previous work (Spratling, 2014) and is directly equivalent to a standard basis function network. The second method (Fig. 3b) involves defining additional neurons within the reconstruction neuron population that perform the same role as the pooling neurons in the first method (Spratling, sub). In this article the second method will be used, as it has the following advantages.

- It is slightly simpler to implement, as it is not necessary to introduce a new population of neurons governed by new equations.
- Mapping is omni-directional. For example, the network shown in Fig. 3b can infer c given a and b (as illustrated in Fig. 4a), and also infer b given a and c (as illustrated in Fig. 4b). This is exploited in the eye control task considered in this article in order to perform sensory-sensory mappings to determine the

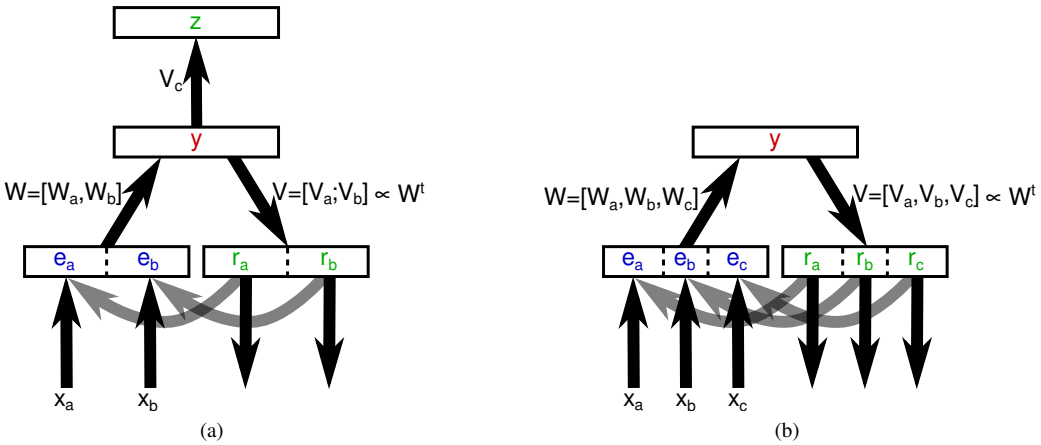


Figure 3: Methods of using PC/BC-DIM as a basis function network. The architectures shown here are analogous to that shown in Fig. 1, for the simple task of mapping from two input variables (a and b) to an output variable (c). (a) The prediction neurons have RFs in the two input spaces (defined by the weights $\mathbf{W}_a$ and $\mathbf{W}_b$) that make them selective to specific combinations of input stimuli. A population of pooling neurons receives input, via weights $\mathbf{V}_c$, from the prediction neurons in order to generate the output. The responses of the pooling neurons, $\mathbf{z}$, are calculated as a linear weighted sum of their input, *i.e.*, $\mathbf{z} = \mathbf{V}_c\mathbf{y}$. (b) The PC/BC-DIM network receives an additional source of input. Dealing with this extra partition of the input requires the definition of additional columns of feedforward synaptic weights, $\mathbf{W}$, and additional rows of the feedback weights, $\mathbf{V}$. If the additional feedback weights, $\mathbf{V}_c$, are identical to the pooling weights used in the architecture shown in (a), then (given equation 1), the responses of the third partition of the reconstruction neurons, $\mathbf{r}_c$, will be identical to the responses of the pooling neurons in (a), *i.e.*, $\mathbf{r}_c = \mathbf{V}_c\mathbf{y}$. If the feedforward weights associated with the third partition, $\mathbf{W}_c$, are rescaled versions of the corresponding additional feedback weights, $\mathbf{V}_c$, then the network can perform mappings not only from a and b to c, but also from a and c to b, and from b and c to a (see Fig. 4).

location of a visual target, and to perform sensory-motor mappings to plan eye movements that will foveate the target (see section 2.3).

- It can be easily extended into a hierarchical architecture that allows mappings to be decomposed into multiple steps, avoiding tractability issues. For example, consider using a basis function network (like that shown in Fig 3b) to map between three variables. If each variable is to be represented to a precision of n, then the number of prediction neurons required to represent the mapping is proportional to n^3. To use a single stage PC/BC-DIM network (like that illustrated in Fig. 5a) to map between four variables would require of the order of n^4 prediction neurons. However, the same task of mapping between four variables can be performed using a hierarchical network, like that illustrated in Fig. 5b. This requires of the order of $2n^3$ basis functions. These theoretical expectations are consistent with practical experience. Specifically, the PC/BC-DIM network that produced the results illustrated in Fig. 4 for mapping between three variables used 361 prediction neurons. A single stage PC/BC-DIM network for mapping between four variables with a similar level of precision needed approximately 2200 neurons. While the results shown in Fig. 6 for mapping between four variables used a hierarchical network containing 494 prediction neurons in total[c]. Hence, by using a hierarchical PC/BC-DIM network to decompose a mapping into multiple steps it is possible to have network size increase linearly (rather than exponentially) with the number of variables. This is important for the eye control application explored in this article, in which there are seven variables. As described in section 2.3, rather than using one PC/BC-DIM network with the order of n^7 prediction neurons, we decompose the problem into three stages using a total number of prediction neurons proportional to $2n^4 + n^3$.

  For any network mapping between a fixed number of variables, the number of prediction neurons will increase as n increases. Increasing the value of n increases the accuracy with which variable values can be represented, and hence, the accuracy with which mappings between those variables can be performed. What value of n is required to perform a particular mapping with sufficient accuracy will depend on the task. For

[c]To perform simulations with a hierarchical model equations 1, 2 and 3 are evaluated for each processing stage in turn (starting from the lowest stage in the hierarchy), and this process is repeated to iteratively calculate the changing neural activations in each processing stage at each time-step.

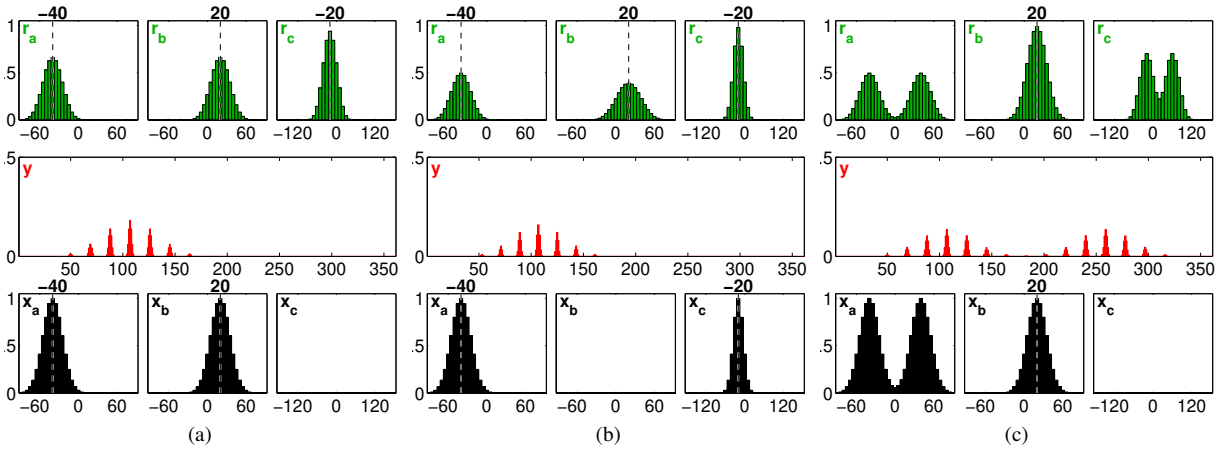


Figure 4: Mapping between three variables. A PC/BC-DIM network with the architecture shown in Fig. 3b is used, where the three partitions of the input are used to represent three different variables. If these variables are denoted as a, b, and c, then the network has been wired-up to calculate $c = a + b$. In each sub-figure the lower histograms show the inputs, the middle histograms show the prediction neuron activations, and the upper histograms show the reconstruction neuron responses. The x-axis of each histogram is labelled with the variable value, except for the histogram representing the prediction neuron responses which is labelled by neuron number. The y-axes of each histogram are in arbitrary units representing firing rate. The values of a, b, and c are represented by a population codes (using Gaussian encoding) so that the encoded value corresponds to the mean of the histogram. This encoded value is indicated by the number above the histogram. Note that c has a wider range of possible values than a and b, and hence, the x-axes of the histograms representing c have a different scale than those representing a and b. (a) When the two inputs representing a and b are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of c (as well as outputs representing the given values of a and b). (b) When the two inputs representing a and c are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of b (as well as outputs representing the given values of a and c). (c) As (a) but with two values of a represented by a bi-modal input to the first partition. The network correctly calculates two values for c represented by the peaks of the bi-modal distribution produced by the reconstruction neurons in the last partition.

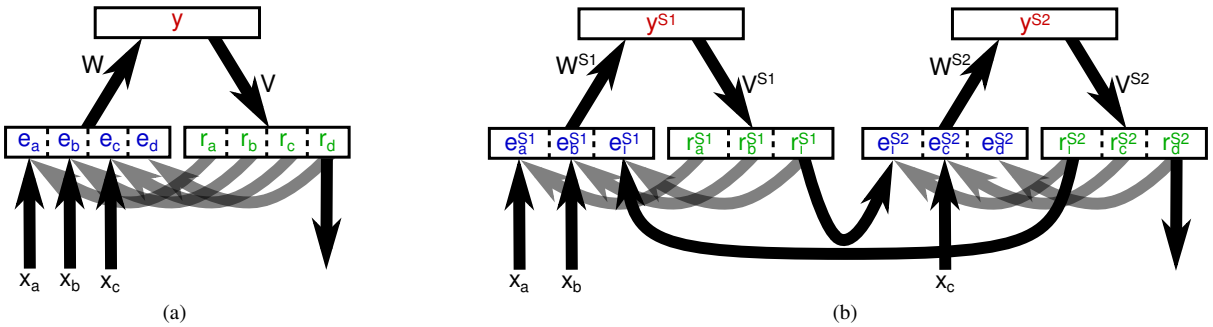


Figure 5: PC/BC-DIM neural network architectures for mapping between four variables. (a) A single-stage network to calculate d given a, b, and c. While it is possible to provide inputs to any of the four partitions, and read outputs from any of the four partitions of the reconstruction neurons, the particular combination of inputs and outputs needed to estimate d given a, b, and c is shown. (b) A hierarchical architecture, consisting of two interconnected PC/BC-DIM networks, for calculating the same function. The first network calculates an intermediate result $(a+b)$ in the third partition of it reconstruction neurons. This intermediate result provides an input to the second PC/BC-DIM network. The second network's reconstruction of this intermediate representation is fed-back as input to the first PC/BC-DIM network.

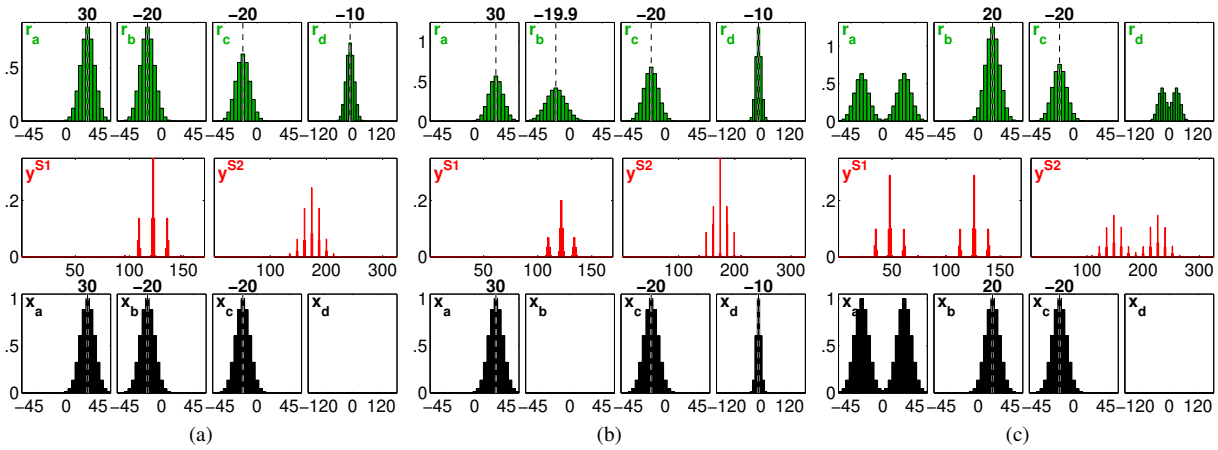


Figure 6: Mapping between four variables using the two-stage (hierarchical) PC/BC-DIM network illustrated in Fig. 5b. The PC/BC-DIM network has been wired-up to approximate the function $d = a + b + c$. The format of each diagram is otherwise the same as that used in, and explained in the caption of, Fig. 4. (a) When the three inputs representing a, b, and c are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of d (as well as outputs representing the given values of a, b, and c). (b) When the three inputs representing a, c and d are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that estimates the correct value of b (as well as outputs representing the given values of a, c and d). (c) As (a) but with two values of a represented by a bi-modal input to the first partition. The network correctly calculates two values for d represented by the peaks of the bi-modal distribution produced by the reconstruction neurons in the last partition.

the eye control application considered here the effective value of n is controlled by the training procedure (see section 2.5) and the effects on saccade accuracy are explored empirically in section 3.

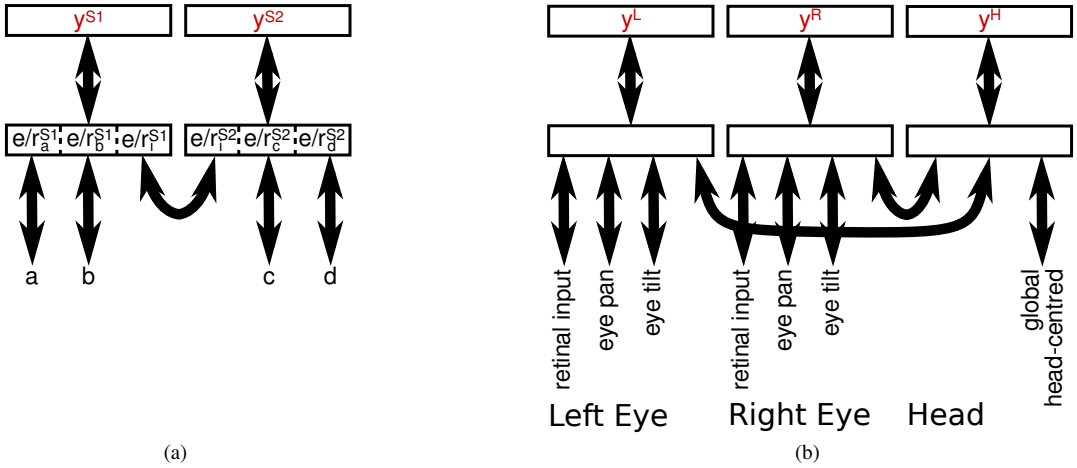
Another advantage of the PC/BC-DIM method over traditional basis function networks is that a mapping can be performed even when the inputs come from multiple sources. This is illustrated for the simple three variable case in Fig. 4c, and for the four variable case, implemented using a hierarchical network, in Fig. 6c. This is exploited for eye movement control to allow the execution of a double-step saccade (see section 3.3).

2.3 The Proposed Eye Control PC/BC-DIM Network

The proposed method for saccade and vergence control in robotics relies on a hierarchy of three PC/BC-DIM processing stages (Fig. 7b). It is unwieldy to draw large PC/BC-DIM networks in the format used previously (*i.e.*,. like that used in Figs. 2, 3, and 5). The proposed network is therefore shown in a simplified format in which the error and reconstruction neuron populations are shown as a single population and the inputs and outputs to these populations are also combined together (see Fig. 7a). The mathematical model remains unchanged, it is just the way of illustrating this model that has been simplified.

The proposed eye control network contains a PC/BC-DIM processing stage (shown on the left of Fig. 7b) that performs mappings between the position of a visual target on the left retina, the position of the left eye in the skull (the left eye pan and tilt), and the head-centred bearing of the left-eye visual target. An identical PC/BC--DIM processing stage, shown in the middle of Fig. 7b, performs the same transformations for the right eye. A third PC/BC-DIM processing stage, shown on the right of Fig. 7b, translates between the individual head-centred representations centred on the left and right eyes, and a global head-centred representation of visual space, that can be driven by targets viewed by either or both eyes.

The proposed model requires access to information about the current eye position (the pan and tilt values). This is consistent with the biological visual system in which eye position signals are known to be used in eye movement control (Donaldson, 2000), and proprioceptive information about eye position is known to be represented in the cortex (Prevosto et al., 2009; Wang et al., 2007). Furthermore, with the proposed model retinal and oculomotor signals are integrated separately for each eye before being combined into a binocular representation, which is consistent with the organisation of the human visual system (Erkelens, 2000). In addition, the model independently computes the movement of each eye which is also consistent with data from the human visual system (Enright,



(a) (b)

Figure 7: (a) The hierarchical PC/BC-DIM network shown in Fig 5b drawn using a simplified format. Here, the error neuron and reconstruction neuron populations are shown superimposed and double-headed arrows are used to show inputs and outputs to and from both these populations. (b) The hierarchical PC/BC-DIM network for eye control drawn using the same simplified format.

1984; Kenyon et al., 1980; Ono et al., 1978).

The proposed eye control network can be used to perform sensory-sensory transformations in which the inputs are visual and proprioceptive and the output is a head-centred representation. For example, a retino-centric representation of a visual target on the retina of the left eye, coupled with inputs representing the current position of the left eye can be used to generate a representation of the head-centred bearing of that visual target in the fourth partition of the first processing stage shown in Fig. 7b. This sensory-sensory transformation is analogous to that shown shown in Fig. 4a for a simpler linear system. Furthermore, if the retinal input contains multiple targets, then the resulting head-centred representation will also represent the bearing of each of those targets, analogous to the situation shown in Fig. 4c. A similar sensory-sensory mapping can be performed by the second processing stage for the right eye. The head-centred representation encoded by the fourth partition of the first and second processing stages (*i.e.*, for the left and right eyes) essentially encodes the radial direction, or bearing, of the target(s) relative to the centre of each eye. The third processing stage, shown on the right of Fig. 7b, acts as a basis function network representing all possible combinations of left and right eye target bearings. The global-head centred representation in the third partition of this PC/BC-DIM processing stage thus represents the 3-dimensional position of a target, as different neurons will respond for different radial positions of the target relative to the head, and also different neurons will respond to targets at the same radial positions but at different depths. This form of spatial representation is described as "headcentric disparity" by Erkelens and van Ee (1998).

The eye control network can also be used to perform sensory-motor transformations. For example, if the first processing stage receives an input to its fourth partition encoding the head-centred bearing of a target and another input encodes the desired retinal location of this target, then the output will be the eye position required to bring the target to this position on the retina. This is analogous to the situation shown in Fig. 4b. Particularly, if the retinal input is a Gaussian population code centred at the fovea, then the network will calculate how to move the eyes to bring the target onto the fovea. The head-centred bearing of the target could be calculated via a preceding sensory-sensory transformation performed by the first processing stage. By using this local head-centred bearing, the first processing stage could be used to plan the movement required for the left-eye to look in the radial direction of the target. A similar process could be performed, independently, using the second processing stage to control the movement of the right eye. This form of sensory-motor mapping for an individual eye was used during the training of the first two processing stages (as described in section 2.5). However, for coordinated movements of both eyes (so that they both foveate the same target), the global head-centred representation encoded by the third processing stage was used. Hence, to foveate a visual target, the following steps were performed. Firstly, for the sensory-sensory transformation step, the retinal and proprioceptive inputs (for both eyes) were transformed into a global head-centred representation of the target by executing the PC/BC-DIM algorithm. Secondly, for the sensory-motor transformation step, this global head-centred representation was provided as input to the network along with two artificial inputs to the retinal partitions of the left and right eye networks (these artificial inputs were 2-dimensional Gaussian population codes centred at the foveal of each eye). The proprioceptive inputs were suppressed so that there were no inputs encoding pan and tilt values. The PC/BC-DIM algorithm was executed

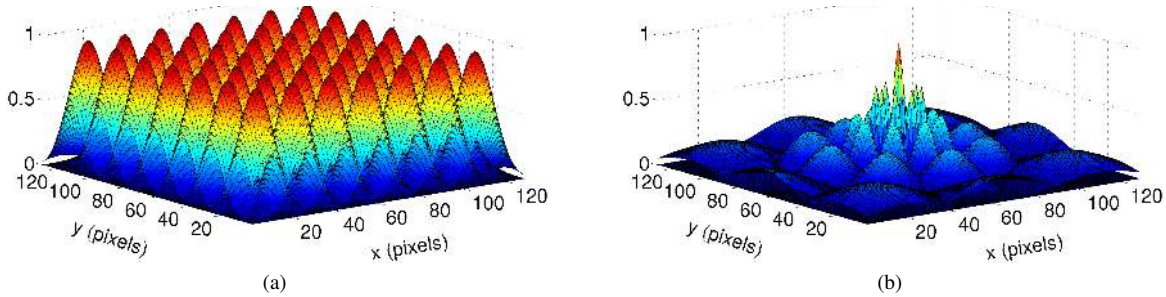


Figure 8: The distribution of Gaussian RFs on the retina. (a) a uniform distribution. (b) a log-polar distribution.

again and the required motor commands were read out from the reconstruction neurons encoding the pan and tilt signals for both eyes.

The same procedure as that described in the preceding paragraph is still used when the target is only visible to a single eye. In this case the response of the global head-centred representation will be more distributed as there is uncertainty about the depth of the target. However, using this more distributed head-centred representation to perform the sensory-motor transformation step will result in the eyes moving so that they are looking in roughly the correct head-centred direction, but are inaccurately verged. This will also result in the target being visible to both eyes, and a second eye movement can be planned (by performing the procedure described in the preceding paragraph again) to correct the position of both eyes.

2.4 Encoding/Decoding the Inputs/Outputs of the Eye Control PC/BC-DIM Network

The retinal input (*i.e.*, $\mathbf{x}_a$) to both the first and second processing stages was encoded using a 2-dimensional array of neurons with Gaussian RFs. For a given visual target, the responses of each retinal neuron was proportional to the overlap of the visual target with its receptive field. These responses were concatenated into a vector to provide the input to the PC/BC-DIM network. The retinal neurons were either arranged in a uniform grid, as illustrated in Fig. 8a, or in a log-polar distribution, as illustrated in Fig. 8b. In the latter case, the spacing between RFs and the variance of the RFs increased with distance from the centre of the retina. In either case one neuron represented the centre of the retina, the foveal location. A log-polar distribution of RFs is consistent with the organisation of the retina in primates (Schwartz, 1977), and has been in robotics on many previous occasions (Javier Traver and Bernardino, 2010).

For the purposes of the simulations reported in section 3 the retinotopic input to the model, the input encoded by the retinal neurons described above, are images captured from the iCub cameras. However, the environment in which the iCub is placed is very impoverished consisting of one or two highly salient objects in front of a blank background. In more realistic environments, it would be necessary to process the raw images to derive a retinotopically organised representations to act as the visual inputs to the model. Each retinotopic input would encode the locations of visual targets for possible saccades. It is assumed that this could be achieved by processing the images to produce some form of saliency map (Niebur, 2007). However, it would be critical that the same salient targets were identified in both the left and right images. The lack of an implemented image pre-processing stage to allow application to realistic environments is a limitation of the current model.

The eye position signals, the eye pan (*i.e.*, $\mathbf{x}_b$) and the eye tilt (*i.e.*, $\mathbf{x}_c$) for both eyes, were each encoded using a 1-dimensional array of neurons with Gaussian RFs that were uniformly distributed between the maximum and minimum values. Decoding these values was performed using standard population vector decoding (Georgopoulos et al., 1986) to find the mean of the distribution of responses.

2.5 Training the Eye Control PC/BC-DIM Network

The networks used above to illustrate how PC/BC-DIM can perform simple linear mappings (*i.e.*, the networks used to produce Figs. 4 and 6) were hard-wired to perform these tasks. Producing networks to perform complex or unknown mappings requires some method of learning the appropriate connectivity. Previous work has shown that this can be achieved using unsupervised learning (De Meyer and Spratling, 2011; Spratling, 2009). However, this learning procedure is slow and rather impractical. Here, we describe a faster, but biologically implausible, procedure for training the weights.

To train the first processing stage (for the left eye) a single, stationary, visual target was presented to the robot.

This target was of a suitable size and distance from the robot so as to produce an image comparable in size to the foveal RF. The left eye was moved systematically while the robot's head and body was kept stationary. As the eye moved distinct combinations of eye pan/tilt and retinal input were generated. These combinations of inputs were represented by different prediction neurons. Each of these prediction neurons was also connected to a single reconstruction neuron in the fourth partition which represented that head-centred bearing. Having trained the network to represent one head-centred direction, the visual target was moved to another location and this training procedure was repeated. Repeating this process systematically for a range of different target positions enabled the first PC/BC-DIM processing stage to learn a head-centred representation of visual space centred on the left eye, *i.e.*, a local representation of the left-eye's world that is invariant to eye position.

One issue with the above method is to decide how many positions to place visual target during training. Clearly the target needs to appear over the full range of positions that the robot needs to learn. However, how finely does this grid of possible locations need to be sampled? Too fine a sampling will lead to a network with an excess of prediction neurons and fourth partition reconstruction neurons. A second issue is to decide how many eye movements the robot needs to make to learn about one head-centred direction. Again, it is clearly necessary for the eye movements to cover the full range of possible eye positions, but how finely does this range need to be sampled? Too fine a sampling will lead to a network with an excess of prediction neurons. To address these issues the following procedure was used. For any given visual target bearing, eye pan, and eye tilt the inputs $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{x}_c$ were saved in memory, but, initially no learning was performed. Instead the PC/BC-DIM network in its current state was used to perform a sensory-sensory mapping in order to estimate of the head-centred bearing of the visual target (as described in section 2.3). The PC/BC-DIM network was then used to perform a sensory-motor mapping in order to calculate the eye motor commands required to bring the visual target into the centre of the retina (as described in section 2.3). These movements were performed. If successful, the target would now be at the fovea, and no learning was performed. If unsuccessful and the target was not in the centre of the retina, then the network was trained so that it would be able to perform these sensory-sensory and sensory-motor transformations in the future. To measure the success of the robot in foveating the target, the responses of all the retinal neurons was normalised by dividing by the maximum response, and then the activation of the neuron in the centre of the retina was measured (see section 2.4). If the response of this neuron was at least 0.8, then the saccade was considered successful.

If the response of the neuron representing the centre of the retina was less than the 0.8 threshold, then the network was updated as follows. If the visual target was at a new head-centred bearing, then a new reconstruction neuron was added to the fourth partition, otherwise the head-centred bearing was already associated with a fourth partition reconstruction neuron. The vector providing input to the fourth partition (*i.e.*, $\mathbf{x}_d$) was set to all zeros, except for the single element corresponding the fourth partition reconstruction neuron representing the current head-centred bearing, which was given a value of one. A new prediction neuron was added to the network. This prediction neuron was given weights corresponding to the inputs received by the first three partitions prior to the movement and the newly calculated input to the fourth partition. Specifically, a new row of $\mathbf{W}$ was created and set equal to $[\tilde{\mathbf{x}}_a; \tilde{\mathbf{x}}_b; \tilde{\mathbf{x}}_c; \tilde{\mathbf{x}}_d]^T$ and a new column of $\mathbf{V}$ was created and set equal to $[\hat{\mathbf{x}}_a; \hat{\mathbf{x}}_b; \hat{\mathbf{x}}_c; \hat{\mathbf{x}}_d]$ (where $\tilde{\mathbf{x}}$ is equal to $\mathbf{x}$ after it has been normalised to sum to one; and $\hat{\mathbf{x}}$ is equal to $\mathbf{x}$ after it has been normalised to have a maximum value of one).

It should be noted that, given the above criteria for adding neurons to the network, the smaller the size of the retinal RF representing the fovea the larger the network will become. Similarly, the smaller the size of the target object used during training the larger the network will become. Furthermore, a larger network would result from using a higher threshold value to decide if a saccade was successful. A smaller foveal RF (or smaller object or a larger threshold) will therefore lead to higher computational cost associated with simulating a network containing more neurons, but will also result in more accurate eye movements. This relationship between computational cost and accuracy will be different when the retinal input is encoded using a uniform grid of equal sized RFs (Fig. 8a), and when it is encoded using RFs arranged in a log-polar distribution (Fig. 8b). The relationship between foveal RF size, computational cost and accuracy are explored in the results section. It is expected that a similar relationship would be found between computational cost and accuracy and training target size or threshold value, but this has not been investigated.

The second processing stage (for the right eye) can be trained in the same way as the first processing stage (for the left eye). However, the results would be identical. Hence, to reduce training time, only the first processing stage was trained, as described above, and the weights were copied over to the second processing stage.

To train the third processing stage, visual targets were presented at all head-centred bearings and at all depths corresponding to vergence angles between 0° to 20°. For each target location the position of both left and right eyes were systematically changed. When the visual targets came into the view of both eyes the local head-centred representations (*i.e.*, y^L and y^R) were produced by the first and second processing stages. To determine if the third processing stage needed to learn this correspondence between the local head-centred representations

a similar criteria to that described previously for learning in the first processing stage was used. Specifically, the PC/BC-DIM network in its current state was used to calculate the global head-centred representation of the visual target. This global head-centred representation was then used to perform a sensory-motor mapping to foveate the visual target with both eyes (as described in section 2.3). The binocular saccade was considered successful if the normalised response of the neurons representing the centre of the both retinas was at least 0.8. In this case, no learning was performed. Otherwise the saccade was considered unsuccessful and a new prediction neuron was added to the third processing stage to associate y^L and y^R with a new global head-centred representation of the visual target. In either case the visual target was moved to the next location to be learnt. For those visual targets at the edge of the visual field which could only be seen by one eye, the local head-centred representation of the viewing eye was used as the local representation of non-viewing eye during the learning procedure described above. This enables the system to control the movements of both eyes, even when the target is beyond the field of view of one eye, although the movement of the non-viewing eye will be inaccurate.

3 Results

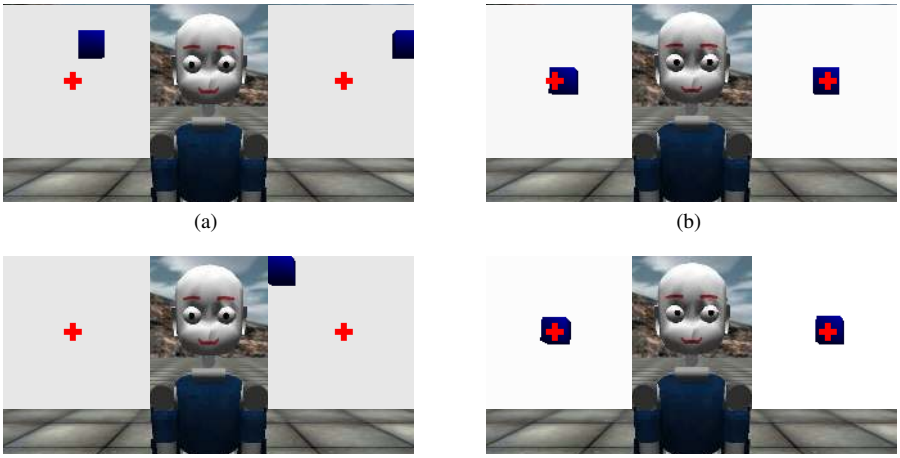
A simulated iCub humanoid robot (Metta et al., 2008; Tikhanoff et al., 2008) with stationary head and body was trained using a visual target which was a box with no gravity and a width, height and length of 0.038 for uniform and 0.01 for log-polar RFs distributions. Each eye of the iCub had a retinal image size of 128x128 pixels, which corresponds to 25.6x26.4 degrees of visual angle. In all experiments, except where explicitly specified otherwise, when using uniformly distributed Gaussian RFs (Fig. 8a) each RF has a size of $\sigma = 7$ pixels, the peak spacing between RF centres was 14 pixels, and 81 RFs were used to uniformly tile the input image. When using a log-polar distribution (Fig. 8b) the retinal plane was populated with 33 RFs, a foveal RF of size $\sigma = 2$ pixels, and 32 further RFs arranged in four concentric circles around the fovea, with the RFs equally spaced around each circle. For all RFs outside the fovea the size (*i.e.*, σ) increased with distance from the fovea, and the amplitude of the RF was reduced proportionally. Eye pan had a range of -20° to +20° and tilt ranged from -12° to +12° and were varied in steps of 1° during learning. The eye position signals were encoded with 1-dimensional Gaussian RFs evenly spaced every 4° and with $\sigma = 2°$.

3.1 Saccade accuracy

To assess performance of the trained PC/BC-DIM network with the iCub simulator, the robot's eyes were given a random pose, and then a visual target was generated at a bearing and depth chosen at random but so that it was visible to at least one eye. The visual input corresponding to the target, together with the proprioceptive information about eye pan/tilt, was used to determine the global head-centred position of the target (see section 2.3). This head-centred target position was subsequently used to determine the eye pan and tilt values (for each eye) required to bring the target to the fovea (see section 2.3). Fig. 9 shows two example simulations of the iCub performing such saccades when the PC/BC-DIM network was trained using a uniform distribution of RFs in each retina.

When using a log-polar distribution of retinal RFs, the initial saccade to peripheral visual targets was inaccurate. This is due to the large size of the peripheral RFs which can not accurately localize the target in the visual periphery. However, the initial saccade does bring the target closer to the fovea where the resolution of the retinal RFs is greater. Hence, it is beneficial to perform a subsequent, "corrective", saccade (similar corrective saccades are seen in human infants and adults; Salapatek et al., 1980). This corrective saccade was performed using a procedure identical to that used for the initial saccade (as described in the preceding paragraph). Fig. 10 shows an example simulation of the iCub performing such saccades.

To quantitatively assess the network's performance at saccade control the post-saccadic distance between the fovea and the centre of the visual target for both eyes was measured on the retinal plane. The mean and standard deviation of this distance was found for 100 trials. Experiments were performed to measure the mean post-saccadic distance for both uniform and log-polar retinal RF distributions. Furthermore, as noted in section 2.5 the size of the foveal RF is expected to affect saccade accuracy. Hence, these experiments were also repeated with different foveal RF sizes, and hence, different size PC/BC-DIM networks. The results are summarised in Fig. 11. It can be seen that saccade accuracy increases slightly as foveal RF size decreases. However, as foveal RF size decreases the size of the PC/BC-DIM network increases (Fig. 11b), which results in longer computation time (Fig. 11c). There is thus a trade-off between the saccade accuracy and computational cost. Comparing performance when using uniform and log-polar RF distributions, it can be seen that for the same foveal RF size, the network with the log-polar distribution of retinal RFs is much faster as it contains fewer neurons than the corresponding network with a uniform distribution of retinal RFs. After corrective saccades are performed by the network with the log-polar distribution of retinal RFs, the accuracy with which the visual target is foveated is only slightly worse than for the model with uniformly distributed retinal RFs. Hence, there is a better trade-off between saccade accuracy and

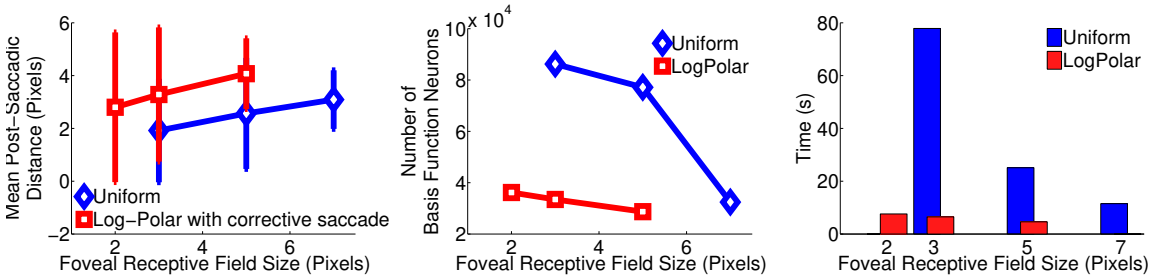


(a) (b) (c) (d)

Figure 9: Example simulations of saccadic eye control with the trained PC/BC-DIM network using the uniform retinal RF distribution. The two windows to the left and right of the iCub show the views of both eyes. The box within these windows is the visual target and the cross hairs mark the location of the fovea in middle of each retina (the cross hairs were not visible to the robot). (a) Before the saccade the visual target is visible in the periphery of both eyes. (b) After saccade execution the target is brought to the centre of both retinas. (c) Before saccade the visual target is visible in only one eye. (d) After the saccade visual target is foveated accurately by both eyes.

(a) (b) (c)

Figure 10: Example simulation of saccadic eye control with the trained PC/BC-DIM network, as in Fig. 9, but using the log-polar distribution of retinal RFs. (a) Before the saccade. (b) After the initial saccade. (c) After the corrective saccade.



(a) (b) (c)

Figure 11: Saccade control performance for the trained PC/BC-DIM network. (a) The effect of foveal RF size on saccade accuracy (measured in terms of the mean post-saccadic distance from the fovea to the centre of the visual target). Error bars show standard deviations. Results are shown for uniform and log-polar retinal RF distributions with corrective saccades. (b) The effect of foveal RF size on the size of the PC/BC-DIM network (measured in terms of the total number prediction neurons). Results are shown separately for uniform and log-polar retinal RF distributions. (c) The effect of foveal RF size on the computational cost per saccade. Error bars show standard deviations. These timings were found using a computer with a Centrino 2 CPU running at 2.4GHz and with 4GB of RAM. Results are shown for uniform and log-polar retinal RF distributions without corrective saccades.

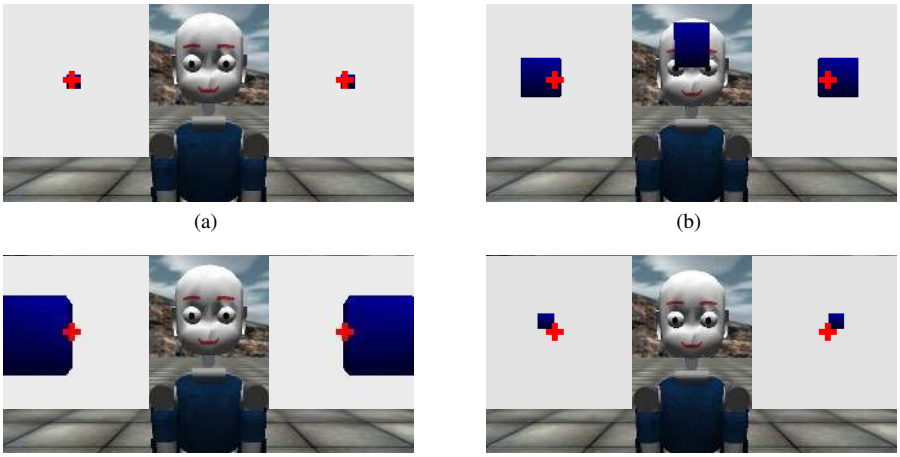


Figure 12: Example simulation of binocular vergence control using the uniform retinal RF distribution. (a) Initial configuration before a covergent movement: both eyes were foveated on a distant object. (b) Final configuration after convergent eyes movements caused by the object coming closer to the eyes. (c) Initial configuration before a divergent movement: both eyes were foveated on a near object. (d) Final configuration after divergent eyes movements caused by the object moving away from the eyes.

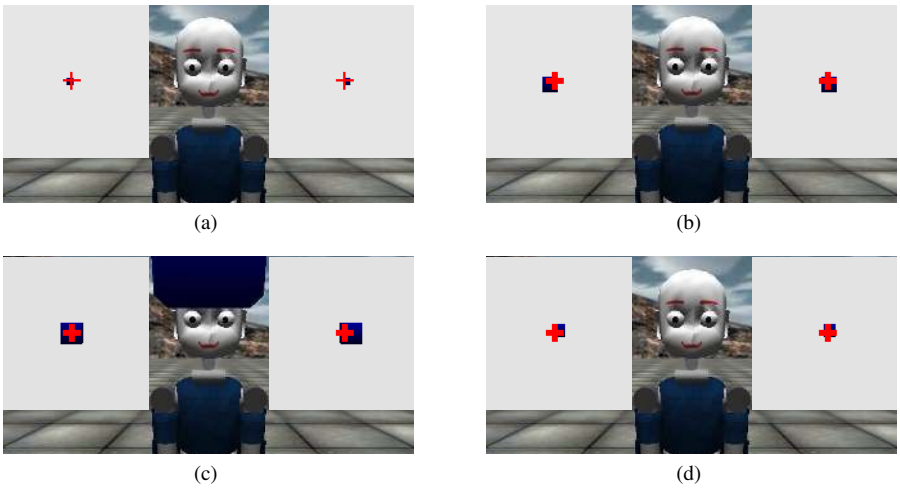
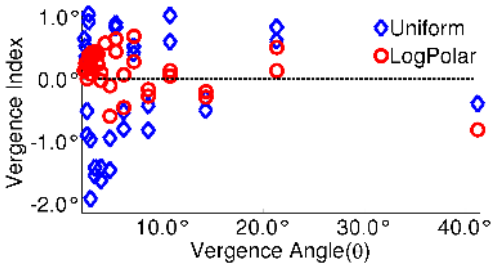


Figure 13: Example simulation of binocular vergence control, as for Fig. 12, but using the log-polar distribution of retinal RFs.

computational cost for a log-polar distribution of retinal RFs. In all experiments with different retinal encoding methods and different foveal RF sizes the mean post-saccadic error remained below five pixels. As five pixels corresponds to approximately 1 degree of visual angle, saccades were performed with an accuracy similar to that of the monkey (Albano and Wurtz, 1982).

3.2 Vergence accuracy

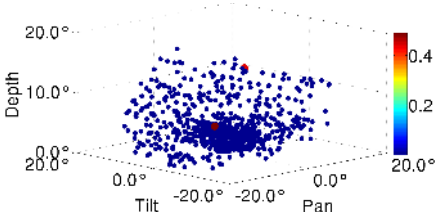
To test vergence control, the depth of the visual target relative to the iCub was varied. As the depth was reduced, the eyes converged to bring the visual target onto the fovea of both eyes. As the depth of the object was increased, the eyes diverged. Examples of the iCub performing vergence control when the PC/BC-DIM network was trained using a uniform distribution of RFs in each retina are shown in Fig. 12, and for a log-polar distribution of retinal RFs in Fig. 13. When performing these vergence movements, the eyes should move an equal amount but in opposite directions (Mays, 1984). To assess the accuracy of vergence movements the sum of the left eye and right eye motor commands were calculated, and this value is referred to as the vergence index. For perfect vergence movements the vergence index would be zero. The values of the vergence index recorded in the iCub simulations



(a)

Figure 14: Vergence control accuracy for the trained PC/BC-DIM network.

(a)



(b)

(c)

(d)

Figure 15: Example simulation of the double-step saccade task using the uniform retinal RF distribution. (a) The two visual targets before saccade execution. (b) The two peaks in the global head-centric map generated by the two targets. Each neuron in the global head-centred map represents a different location in 3-dimensional head-centred visual space. Each dot shows the location represented by a neuron (the neuron's RF centre), and the colour of the dot indicates the response of the neuron to the stimulus shown in (a). (c) After the first saccade the first target is visible near the fovea of both eyes, but the second target is no longer visible to either eye. (d) After the second saccade the second target is visible near the fovea of both eyes.

are shown in Fig. 14. It can be seen that the model (using either method of distributing the retinal RFs) produces results comparable to the $\pm 2^o$ error observed in human subjects (Cornell et al., 2003). Given that the accuracy of foveation varies with foveal RF size (as shown in Fig. 11a), the accuracy of vergence movements would also be expected to vary with foveal RF size. The results shown here were produced using a foveal RF of size of $\sigma = 7$ pixels for the uniform distribution and $\sigma = 2$ pixels for the log-polar distribution of retinal RFs.

3.3 Double-step saccades

If more than one target is presented to the visual field, the human oculomotor system can perform saccades sequentially to each location even if the second object is invisible to both eyes after the first saccade (Aslin and Shea, 1987; Heide et al., 1995; Komoda et al., 1973). When more than one visual target is presented simultaneously to the PC/BC-DIM network, it represents these visual targets by global head-centred representations. Each visual target is represented by a separate peak in the activity of the reconstruction neurons in the third partition of the third processing stage (see section 2.3). By storing in memory each of these peaks, it is possible to perform a saccade to each location in turn as illustrated in Figs. 15 and 16. Over 100 trails with randomly chosen head-centred target locations, the accuracy of both the first and second saccades was equal to that shown in Fig. 11, except in cases when the two targets were in close proximity. When two targets are separated by a distance less than the retinal RF size they produce one peak, rather than two peaks, in the global head-centred representation. In such

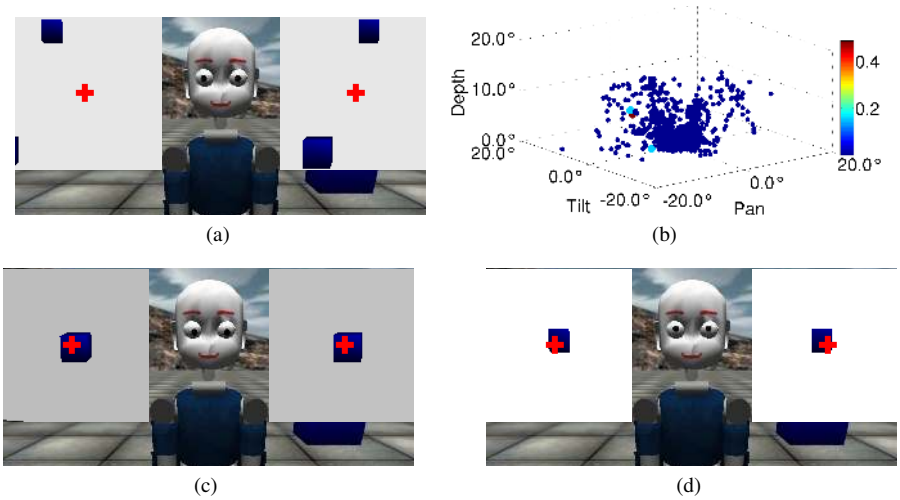


Figure 16: Example simulation of the double-step saccade task, as for Fig. 15, but using the log-polar distribution of retinal RFs.

circumstances, rather than a double-step saccade, one saccade is made to a position intermediate between the two targets. This is a particular problem when the retinal RFs are arranged in a log-polar distribution, as the distance between the retinal RFs in the periphery of the retina is large, meaning that this problem is more often encountered for a log-polar than a uniform retinal RF distribution.

4 Discussion

This article has introduced a novel basis-function type neural network that can perform omni-directional mappings between different sensory and motor representations. To demonstrate this method, it has been applied to saccade planning and vergence control in the iCub humanoid robot simulator. We have described a simple method to learn the appropriate connectivity of the network which uses eye movements to learn an internal representation of head-centred visual space (*i.e.*, one that is invariant to eye-movements). Once trained the network can take visual and eye pan/tilt signals as inputs and map these to the corresponding head-centred representation of visual space. Because the network can perform omni-directional mappings, the same network that performs this sensory-sensory mapping can be used to also perform a sensory-motor mapping. Specifically, it can take the head-centred representation of the target (calculated previously) and a desired retinal location for the target and output the eye pan and tilt values required to achieve this. Hence, if the desired retinal location is defined as the fovea, then the network can be used to generate a saccade. Because the trained network produces coordinated movements of both eyes to the same target, it is able to perform vergence eye movements in addition to saccades. Other models of vergence control rely on disparity detection (Gibaldi et al., 2013, 2015, 2009; Patel et al., 1997; Theimer and Mallot, 1994; Vikram et al., 2014; Zhao et al., 2012). The approach proposed in this article is complementary to these disparity-based methods. Specifically, the existing methods perform fine adjustments to eye position in order to cancel disparity. These small movements are performed under closed-loop control. In contrast, the proposed approach performs large-scale eye movements using open-loop control, consistent with the ballistic (open-loop) saccades performed by the biological visual system (Chao et al., 2010; Findlay and Walker, 2012). It seems likely that both these approaches will be used in animals, and that both could be combined in robotic systems to obtain the complimentary advantages of both approaches.

Some particular advantages of the proposed approach derive from it using a head-centred representation of the target location to plan eye movements. This head-centred representation is invariant to eye position, and hence, it is unaffected by eye jitter or eye fixation errors. This allows accurate eye movement control in the presence of such errors. In addition, the head-centred representation of target location enables the model to execute a saccade even when the target is visible to only one eye, or if it is no longer visible to either eye. This is consistent with evidence that humans perform double-step saccades using head-centred representations (Heide et al., 1995; Pertzov et al., 2011; Zimmermann et al., 2011). There are also several other ways in which the proposed model is consistent with biology. For example, the model and the biological system both use proprioceptive information about eye position (Donaldson, 2000) to plan ballistic eye movements (Findlay and Walker, 2012). The model and the biological system both integrate retinal and oculomotor information separately for each eye before combining this

information into a binocular representation (Erkelens, 2000). The model independently computes the movement of each eye which is also consistent with data from the human visual system (Enright, 1984; Kenyon et al., 1980; Ono et al., 1978). Finally, the model represents visual information in multiple coordinates systems (a retinotopic one and a head-centred one) which is consistent with evidence for the existence of multiple coordinate systems in cortex (Battaglia-Mayer et al., 2003; Blangero, 2008; Marzocchi et al., 2008; McGuire and Sabes, 2009; Pertzov et al., 2011).

We explored two different methods of encoding retinotopic information about target location: using a uniform distribution of retinal RFs, and using a log-polar distribution of RFs. In the latter case it was necessary to perform corrective saccades to produce accurate eye movements. For both methods the mean post-saccadic error was less than 1° which is comparable to the error in the biological ocular-motor system (Albano and Wurtz, 1982). The dissimilarity between binocular vergence motor commands was up to $\pm 2^\circ$ for the uniform distribution of retinal RFs and up to $\pm 1^\circ$ for the log-polar distribution. This is comparable to the error observed in humans which is up to $\pm 2^\circ$ under natural conditions (Cornell et al., 2003). While both methods produced accurate eye movements, the log-polar distribution had a distinct advantage in terms of computation cost, as it resulted a network containing fewer neurons. However, it also had the disadvantage that objects appearing in close proximity in the periphery could not be distinguished due to the lower acuity in the periphery of the retina in this version of the model.

The proposed network is capable of representing multiple visual targets simultaneously. This was demonstrated using a double-step saccade task. It was shown that the iCub could be controlled to perform saccades sequentially to two different targets. It was capable of doing so even when the initial saccades made the second target invisible to both eyes. The proposed method is also capable of decomposing complex tasks into multiple, more tractable, sub-tasks by using a hierarchical neural network architecture. This was demonstrated here by using a three-stage hierarchical network to perform all tasks. Specifically, separate stages were used to perform mappings between sensory inputs and a local head-centred representation for each eye, and a third processing stage was used to map between these local head-centred representations and a global one. This global head-centred representation enabled both eyes to saccade to the same visual target, even if the target was only visible to one eye. It is planned to further exploit this ability to build hierarchical networks in future work. We plan to learn transformations to and from a head-centred representation of visual space, and a body-centred representation, which can be used to develop a more comprehensive model of coordinated eye and head movement control and to plan visually guided reaching.

Acknowledgements

This work was partially funded by Higher Education Commission Pakistan under grant No. PM(HRDI-UESTPs)/-UK/HEC/2012.

References

Albano, J. and Wurtz, R. (1982). Deficits in eye position following ablation of monkey superior colliculus, pretectum, and posterior-medial thalamus. *Journal of Neurophysiology*, 48(2):318–337.

Andersen, R. A., Essick, G. K., and Siegel, R. M. (1985). Encoding of spatial location by posterior parietal neurons. *Science*, 230(4724):456–8.

Aslin, R. N. and Shea, S. L. (1987). The amplitude and angle of saccades to double-step target displacements. *Vision Research*, 27(11):1925–1942.

Battaglia-Mayer, A., Caminiti, R., Lacquaniti, F., and Zago, M. (2003). Multiple levels of representation of reaching in the parieto-frontal network. *Cerebral Cortex*, 13(10):1009–22.

Blangero, A. (2008). *The Sensorimotor Functions Of The Posterior Parietal Cortex: Evidence from patients with Optic Ataxia*. PhD thesis, L'Universite De Lyon, France.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–55.

Brotchie, P. R., Andersen, R. A., Snyder, L. H., and Goodman, S. J. (1995). Head position signals used by parietal neurons to encode locations of visual stimuli. *Nature*, 375(6528):232–5.

Chafee, M. V., Averbeck, B. B., and Crowe, D. A. (2007). Representing spatial relationships in posterior parietal cortex: Single neurons code object-referenced position. *Cerebral Cortex*, 17(12):2914–32.

Chao, F., Lee, M. H., and Lee, J. J. (2010). Adevelopmental algorithm for ocular-motor coordination. *Robotics and Autonomous Systems*, 58:239–48.

Chinellato, E., Antonelli, M., Grzyb, B., and del Pobil, A. P. (2011). Implicit sensorimotor mapping of the

peripersonal space by gazing and reaching. *IEEE Transactions on Autonomous Mental Development*, 3(1):43–52.

Cornell, E. D., Macdougall, H. G., Predebon, J., Curthoys, I. S., et al. (2003). Errors of binocular fixation are common in normal subjects during natural conditions. *Optometry and Vision Science*, 80(11):764–771.

De Meyer, K. and Spratling, M. W. (2011). Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function. *Neural Computation*, 23(6):1536–67.

De Meyer, K. and Spratling, M. W. (2013). A model of partial reference frame transforms through pooling of gain-modulated responses. *Cerebral Cortex*, 23(5):1230–9.

Deneve, S., Latham, P. E., and Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. *Nature Neuroscience*, 2(8):740–5.

Deneve, S., Latham, P. E., and Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, 4(8):826–31.

Deneve, S. and Pouget, A. (2003). Basis functions for object-centered representations. *Neuron*, 37:347–59.

Donaldson, I. M. (2000). The functions of the proprioceptors of the eye muscles. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 355(1404):1685–754.

Duhamel, J.-R., Bremmer, F., BenHamed, S., and Graf, W. (1997). Spatial invariance of visual receptive fields in parietal cortex neurons. *Nature*, 389.

Enright, J. (1984). Changes in vergence mediated by saccades. *The Journal of physiology*, 350(1):9–31.

Erkelens, C. J. (2000). Perceived direction during monocular viewing is based on signals of the viewing eye only. *Vision Research*, 40(18):2411–2419.

Erkelens, C. J. and van Ee, R. (1998). A computational model of depth perception based on headcentric disparity. *Vision Research*, 38(19):2999–3018.

Findlay, J. and Walker, R. (2012). Human saccadic eye movements. *Scholarpedia*, 7(7):5095. revision 122018.

Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233:1416–9.

Gibaldi, A., Canessa, A., Chessa, M., Solari, F., and Sabatini, S. P. (2013). Population coding for a reward-modulated hebbian learning of vergence control. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Gibaldi, A., Canessa, A., Solari, F., and Sabatini, S. P. (2015). Autonomous learning of disparityvergence behavior through distributed coding and population reward: Basic mechanisms and real-world conditioning on a robot stereo head. *Robotics and Autonomous Systems*, 71:23–34.

Gibaldi, A., Chessa, M., Canessa, A., Sabatini, S. P., and Solari, F. (2009). A neural model for binocular vergence control without explicit calculation of disparity. In *Proceedings of the European Symposium on Artificial Neural Networks*.

Hartmann, T. S., Bremmer, F., Albright, T. D., and Krekelberg, B. (2011). Receptive field positions in area MT during slow eye movements. *The Journal of Neuroscience*, 31(29):10437–44.

Heide, W., Blankenburg, M., Zimmermann, E., and Kömpf, D. (1995). Cortical control of double-step saccades: implications for spatial orientation. *Annals of Neurology*, 38(5):739–48.

Hoffmann, M., Marques, H. G., Arieta, A. H., Sumioka, H., Lungarella, M., and Pfeifer, R. (2010). Body schema in robotics: A review. *IEEE Transactions on Autonomous Mental Development*, 2(4):304–24.

Huang, Y. and Rao, R. P. N. (2011). Predictive coding. *WIREs Cognitive Science*, 2:580–93.

Javier Traver, V. and Bernardino, A. (2010). A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 58(4):378–398.

Kenyon, R., Ciuffreda, K., and Stark, L. (1980). Dynamic vergence eye movements in strabismus and amblyopia: symmetric vergence. *Investigative ophthalmology & visual science*, 19(1):60–74.

Kim, D., Huh, S.-H., Seo, S.-J., and Park, G.-T. (2005). Self-organizing radial basis function network modeling for robot manipulator. In Ali, M. and Esposito, F., editors, *Innovations in Applied Artificial Intelligence*, volume 3533 of *Lecture Notes in Computer Science*, pages 579–87. Springer Berlin Heidelberg.

Komoda, M. K., Festinger, L., Phillips, L. J., Duckman, R. H., and Young, R. A. (1973). Some observations concerning saccadic eye movements. *Vision Research*, 13(6):1009–20.

Marjanović, M., Scassellati, B., and Williamson, M. (1996). Self-taught visually guided pointing for a humanoid robot. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats: Proceedings of the International Conference on Simulation of Adaptive Behaviour*, pages 35–44, Cambridge, MA. MIT Press.

Marzocchi, N., Breveglieri, R., Galletti, C., and Fattori, P. (2008). Reaching activity in parietal area V6A of macaque: eye influence on arm activity or retinocentric coding of reaching movements? *European Journal of Neuroscience*, 27(3):775–89.

Mays, L. E. (1984). Neural control of vergence eye movements: convergence and divergence neurons in midbrain.

Journal of Neurophysiology, 51(5):1091–1108.

McGuire, L. M. M. and Sabes, P. N. (2009). Sensory transformations and the use of multiple reference frames for reach planning. *Nature Neuroscience*, 12(8):1056–61.

Meng, Q. and Lee, M. H. (2007). Automated cross-modal mapping in robotic eye/hand systems using plastic radial basis function networks. *Connection Science*, 19(1):25–52.

Meng, Q. and Lee, M. H. (2008). Error-driven active learning in growing radial basis function networks for early robot learning. *Neurocomputing*, 71(7–9):1449–61.

Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). The icub humanoid robot: An open platform for research in embodied cognition. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, PerMIS '08, pages 50–6, New York, NY, USA. ACM.

Molina-Vilaplana, J., Pedreño-Molina, J. L., and López-Coronado, J. (2004). Hyper RBF model for accurate reaching in redundant robotic systems. *Neurocomputing*, 61:495–501.

Niebur, E. (2007). Saliency map. *Scholarpedia*, 2(8):2675. revision 147400.

Ono, H., Nakamizo, S., and Steinbach, M. J. (1978). Nonadditivity of vergence and saccadic eye movement. *Vision Research*, 18(6):735–739.

Park, J. and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–57.

Patel, S., Öğmen, H., White, J., and Jiang, B. (1997). Neural network model of short-term horizontal disparity vergence dynamics. *Vision Research*, 37(10):1383–1399.

Pertzov, Y., Avidan, G., and Zohary, E. (2011). Multiple reference frames for saccadic planning in the human parietal cortex. *The Journal of Neuroscience*, 31(3):1059–68.

Pouget, A., Deneve, S., and Duhamel, J. R. (2002). A computational perspective on the neural basis of multisensory spatial representations. *Nature Reviews Neuroscience*, 3:741–7.

Pouget, A. and Sejnowski, T. J. (1994). A neural model of the cortical representation of egocentric distance. *Cerebral Cortex*, 4(3):314–29.

Pouget, A. and Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience*, 9(2):222–37.

Pouget, A. and Snyder, L. (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience*, 3(supplement):1192–8.

Prevosto, V., Graf, W., and Ugolini, G. (2009). Posterior parietal cortex areas MIP and LIPv receive eye position and velocity inputs via ascending preposito-thalamo-cortical pathways. *European Journal of Neuroscience*, 30:1151–61.

Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.

Salapatek, P., Aslin, R. N., Simonson, J., and Pulos, E. (1980). Infant saccadic eye movements to visible and previously visible targets. *Child Development*, pages 1090–1094.

Salinas, E. and Abbott, L. F. (1995). Transfer of coded information from sensory to motor networks. *The Journal of Neuroscience*, 15:6461–74.

Salinas, E. and Sejnowski, T. J. (2001). Gain modulation in the central nervous system: where behavior, neurophysiology and computation meet. *The Neuroscientist*, 7(5):430–40.

Schilling, R. J., Carroll, J. J., and Al-Ajlouni, A. F. (2001). Approximation of nonlinear systems with radial basis function neural networks. *IEEE Transactions on Neural Networks*, 12(1).

Schwartz, E. L. (1977). Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194.

Snyder, L. H., Grieve, K. L., Brotchie, P., and Andersen, R. A. (1998). Separate body- and world-referenced representations of visual space in parietal cortex. *Nature*, 394:887–91.

Spratling, M. W. (2008a). Predictive coding as a model of biased competition in visual selective attention. *Vision Research*, 48(12):1391–408.

Spratling, M. W. (2008b). Reconciling predictive coding and biased competition models of cortical function. *Frontiers in Computational Neuroscience*, 2(4):1–8.

Spratling, M. W. (2009). Learning posture invariant spatial representations through temporal correlations. *IEEE Transactions on Autonomous Mental Development*, 1(4):253–63.

Spratling, M. W. (2012). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1):60–103.

Spratling, M. W. (2014). Classification using sparse representations: a biologically plausible approach. *Biological Cybernetics*, 108(1):61–73.

Spratling, M. W. (sub.). Predictive coding as a model of cognition. *submitted*.

Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image compo-

nents using divisive input modulation. *Computational Intelligence and Neuroscience*, 2009(381457):1–19.

Sun, G. and Scassellati, B. (2005). A fast and efficient model for learning to reach. *Int. J. Human. Robot.*, 2(4):391–413.

Theimer, W. M. and Mallot, H. A. (1994). Phase-based binocular vergence control and depth reconstruction using active vision. *CVGIP: Image Understanding*, 60(3):343–358.

Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., and Nori, F. (2008). An open-source simulator for cognitive robotics research: The prototype of the icub humanoid robot simulator. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, PerMIS '08, pages 57–61, New York, NY, USA. ACM.

van Rossum, M. C. W. and Renart, A. (2004). Computation with populations codes in layered networks of integrate-and-fire neurons. *Neurocomputing*, 58–60:265–70.

Vikram, T. N., Teuliere, C., Zhang, C., Shi, B., and Triesch, J. (2014). Autonomous learning of smooth pursuit and vergence through active efficient coding. In *Proceedings of the IEEE International Conferences on Development and Learning and Epigenetic Robotics*, pages 448–53.

Wang, X., Zhang, M., Cohen, I. S., and Goldberg, M. E. (2007). The proprioceptive representation of eye position in monkey primary somatosensory cortex. *Nature Neuroscience*, 10:640–6.

Weber, C., Elshaw, M., Triesch, J., and Wermter, S. (2007). Neural control of actions involving different coordinate systems. In Hackel, M., editor, *Humanoid Robots: Human-like Machines*. I-Tech Education and Publishing, Vienna, Austria.

Zhang, P.-Y., L, T.-S., and Song, L.-B. (2005). RBF networks-based inverse kinematics of 6R manipulator. *The International Journal of Advanced Manufacturing Technology*, 26(1-2):144–7.

Zhao, Y., Rothkopf, C. A., Triesch, J., and Shi, B. E. (2012). A unified model of the joint development of disparity selectivity and vergence control. In *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6.

Zimmermann, E., Burr, D., and Morrone, M. C. (2011). Spatiotopic visual maps revealed by saccadic adaptation in humans. *Current Biology*, 21(16):1380–4.