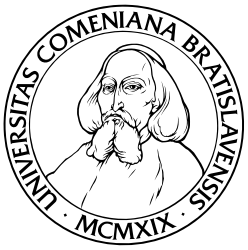
Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics
Department of Applied Informatics



Representation of object position in various frames of reference using a robotic simulator

Master Thesis

Bc. Marcel Švec 2013

Univerzita Komenského v Bratislave
Fakulta Matematiky, Fyziky a Informatiky
Katedra Aplikovanej Informatiky

Reprezentácia pozície objektu v rôznych referenčných rámcoch s využitím robotického simulátora

Diplomová práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiteľ:	doc. Ing. Igor Farkaš, PhD.

Bc. Marcel Švec

Bratislava, 2013

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Marcel Švec
Study programme:	Applied Computer Science (Single degree study, master II. deg., full time form)
Field of Study:	9.2.9. Applied Informatics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title:	Representation of object position in various frames of reference using a robotic simulator
Aim:	1. Study the cognitive neuroscience literature related to the frames of reference. 2. Make an overview of existing neural network models designed for coordinate translation between different frames of reference. 3. Implement and evaluate a neural network model capable of coordinate translation using the robotic simulator iCub.
Annotation:	Humans use several egocentric frames of reference (coordinate systems) for effective sensorimotor coordination (eye-hand). This knowledge is applied in cognitive robotics, where the goal of the design is to create inner representation of the surrounding space in simulated or physical agents.
Comment:	Requirements: good English, interest in cognitive robotics, ability to work independently, own initiative and systematic work.
Keywords:	frame of reference, artificial neural network, object coordinates, learning

Supervisor:	doc. Ing. Igor Farkaš, PhD.
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. PhDr. Ján Rybár, PhD.
Assigned:	18.10.2012
Approved:	20.11.2012

doc. RNDr. Roman Ďurikovič, PhD.
Guarantor of Study Programme

.....
Student

.....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Marcel Švec
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Reprezentácia pozície objektu v rôznych referenčných rámcoch s využitím robotického simulátora

Cieľ:
1. Naštudujte si problematiku z kognitívnej neurovedy o referenčných rámcoch.
2. Urobte stručný prehľad existujúcich modelov neurónových sietí, ktoré realizujú prepočítavanie súradníc medzi referenčnými rámcami.
3. Implementujte a otestujte model neurónovej siete, ktorá sa naučí realizovať túto funkciu, s využitím simulovaného robota iCub.

Anotácia: Človek využíva viaceré egocentrické referenčné rámce (súradnicové sústavy) pri efektívnej senzomotorickej koordinácii (oko-ruka). Tieto poznatky sú využívané v kognitívnej robotike, kde cieľom dizajnu je vytvorenie vnútorných reprezentácií okolitého priestoru u (simulovaného alebo fyzického) agenta.

Poznámka: Požiadavky: pasívna znalosť angličtiny, záujem o kognitívnu robotiku, relatívna samostatnosť, vlastná iniciatíva a priebežná práca.

Kľúčové slová: referenčný rámec, umelá neurónová sieť, súradnice objektu, učenie

Vedúci: doc. Ing. Igor Farkaš, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.

Dátum zadania: 18.10.2012

Dátum schválenia: 20.11.2012

doc. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

DECLARATION

I hereby declare that this thesis is my own work and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

.....

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, doc. Ing. Igor Farkaš, PhD., for his helpful advice, encouragement and continual support. His insightful guidance and friendly approach helped me in all stages of this project. A big thanks belongs also to my friends and family, especially to my mum for all the numerous and delicious meals she prepared while I was studying or working on this thesis.

Abstract

Almost every human interaction with the world is made by hands, while the primary source of information for the perception is the visual system. Neurons in early visual pathways and primary visual cortex encode spatial information in retinotopic frame of reference and their activities change with every eye movement. To look at a specific target or to reach for an object, the brain has to transform perceived sensory information into motor plans that cannot be purely retinotopic, but have to consider also postural signals. This process of sensorimotor transformation has been of great interest for neuroscientists in the past two decades. The concept of reference frames borrowed from physics has helped to formulate the problem as a coordinate transformation and became a main tool for studying the computational aspects of spatial processing in the human brain. Research suggests that one of the crucial and widespread parts of this processing is the phenomenon known as gain modulation. Important insight into the problem can be obtained through computational models that are mostly based on artificial neural networks and that are in the focus of this master thesis. We reviewed existing neural network models trained to perform coordinate transformations and proposed our own model that does not differ much in the network architecture, but which processes inputs obtained from the humanoid robotic simulator iCub. Trained network was able to successfully perform coordinate transformation from eye- to body-centered reference frame using the information about gaze direction with the accuracy within 2°. We proposed several visualisation techniques for analysing the hidden structures of the network and observed the effect of gain modulation and shifting receptive fields. We also formulated hypothesis about the crucial role of gain modulation in the process of spatial transformations. The main potential of our approach lies in the fact that iCub simulator reflects the real geometry of the human body and sensory system in 3D.

Keywords: frame of reference, coordinate transformation, gain field, artificial neural network, computational neuroscience, iCub simulator

Abstrakt

Najdôležitejším nástrojom ľudskej interakcie s okolím sú ruky. Na to, aby sme dokázali objekt uchopiť, potrebujeme poznať jeho polohu v priestore. Táto informácia je v našom mozgu reprezentovaná určitou populáciou neurónov, ktorá by podľa všetkého mala reprezentovať pozíciu objektu nezávisle na tom, kam sa pozeráme, alebo ako máme natočenú hlavu. Je známe, že neuróny zrakovej dráhy a zrakového centra kódujú priestorovú informáciu retinotopicky. To znamená, že objekt ktorý dopadne na sietnicu vľavo, je touto populáciou reprezentovaný ako nachádzajúci sa vľavo. Pri každom pohybe oka sa preto táto reprezentácia zmení a vzniká otázka, akým spôsobom nastáva transformácia tejto prvotnej informácie do iných reprezentácií, ktoré sú vhodné napríklad na uchopenie objektu. V oblasti výpočtovej neurovedy sa ako vhodný nástroj na analýzu priestorových transformácií v prostredí neurónových sietí ukázal koncept referenčného rámca. Výskumy z posledných dvoch desaťročí odhalili ako jeden zo základných princípov tejto transformácie efekt známy pod názvom zisková modulácia alebo modulácia zisku. V práci sa najskôr venujeme teoretickým základom priestorových transformácií v neurónových sieťach a uvádzame prehľad výpočtových modelov navrhnutých na tento účel. Potom prezentujeme náš vlastný model neurónovej siete, ktorá sa od pôvodných modelov líši tým, že spracováva vstupy generované robotickým simulátorom iCub. Natrénovaná sieť bola schopná transformovať retinotopické súradnice do súradníc tela použitím informácie o natočení očí. Presnosť siete bola 2°. Analýza výpočtových vlastností sieti založená na vizualizácií parametrov skrytých neurónov odhalila efekt ziskovej modulácie a posúvajúcich sa receptívnych polí. V práci vyslovujeme hypotézu o kľúčovej úlohe ziskovej modulácie v transformácií priestorových súradníc. Hlavnú výhodu použitia robotického simulátora vidíme v tom, že odráža skutočné geometrické vlastnosti ľudského tela.

Kľúčové slová: referenčný rámec, transformácia súradníc, gain field (ziskové polia), úmelá neurónová sieť, výpočtová neuroveda, iCub simulátor

Foreword

The simplest questions are the hardest to answer. We live in a world where almost every our action requires precise hand movements. We thus simply ask: how do we make it that we reach for the desired object? Because this question is too simple for being answered, we made the question harder hoping that we might find the answer more easily: what are the computations that the brain has to carry out in order to transform the sensory and postural signals into the spatial representation that can be further used for the reach? Well, the answer is still hard to find, but we already have some clues. Our thesis is about one of them.

Contents

List of Figures

List of Symbols and Abbreviations

ANN	Artificial neural network
BP	Back-propagation
CS	Coordinate system
FANN	Fast Artificial Neural Network Library
FF-ANN	Feed-Forward Artificial Neural Network
GF	Gain Field
HLU	Hidden layer unit
LIP	Lateral intraparietal area (brain area)
MSE	Mean Squared Error
ODE	Open Dynamic Engine
OpenGL	Open Graphics Library
PPC	Posterior parietal cortex (brain area)
PRR	Parietal reach region (brain area)
RBF	Radial basis functions; Radial basis function network
RF	Receptive field
RNN	Recurrent neural network
ROC	Rank-order coding algorithm
RPC	Remote procedure call
SOM	Self-organizing map
V1	Primary visual cortex (brain area)
VIP	Ventral intraparietal area (brain area)
VOR	Vestibulo-ocular reflex, indicates opposite eye-head movements
YARP	Yet Another Robot Platform

1 Introduction

Human brain is a highly complex system able to perform a huge number of non-trivial computations. The exemplar case of such computation is the transformation of the object position in the retina into the spatial representation that permits to reach for the given object. In this computation, the brain has to use also the information about the eye and head position in order to be accurate. In our thesis we simply ask: what are the computational principles underlying these transformations in the environment of neural networks?

After the short introduction into the field of computational neuroscience we focus on the concept of reference frames. We explain the connections between reference frames and multimodal integrations, specifically in the form of sensorimotor transformations. We introduce the phenomenon known as gain modulation and discuss its crucial role in coordinate transformations. Later we describe two main types of network architectures used for multimodal integrations and explain past and recent works in this research area. The next chapter is dedicated to the descriptions of main methods that we used in our experiment, namely to the robotic simulator iCub and the basic theory of artificial neural networks. At the end we explain our experiment that consisted of training artificial neural network to perform coordinate transformations from eye- to body-centered reference frame using information about the eyes position. We then discuss the results by the means of several visualisation techniques used for examining the computational principles in the network.

1.1 Computational neuroscience

Neuroscience, the scientific field that studies the nervous system, aims to explain many interesting, but highly complex questions about various aspects of human perception, mind and behaviour. Naturally, the first research was oriented at the biology of the nervous system, its basic role in the human or animal body, overall structure, physiological properties and principles of underlying chemical processes. Such understanding is essential for the treatment of the nervous system diseases. At a different level, many other questions arose about how neural circuits can give rise to cognitive functions, how mental abilities develop, in what form is the memory encoded or what is the source of emotions. This way the neuroscience became an interdisciplinary field with many overlapping branches from neurophysiology, neuroanatomy and molecular neuroscience, ranging to be-

havioural, developmental, cognitive and computational neuroscience.

Our thesis is settled in the context of computational neuroscience, which itself is an interdisciplinary field that (roughly speaking) combines knowledge from cognitive and computer sciences. Computational neuroscience investigates the functions and mechanisms behind cognitive abilities such as processing sensory signals or generating motor commands by constructing computational models that are able to perform the given task and that are usually inspired by the essential features of the biological system. Created models are used to test hypotheses that arose from physiological experiments, or on the other hand, to formulate new hypotheses that can be verified later. The majority of models is based on the theory of neural computing and artificial neural networks (section 3.2).

2 Frames of Reference

To determine the object position in space we always relate to some point at known location. In real life we usually speak about the object position relative to our body or with respect to some other object we see or know - we use egocentric or allocentric frame of reference. In mathematics, we represent and transform positions in coordinate systems (CS) describing the position by a set of numbers (coordinates). There are many coordinate systems that differ in the interpretation of coordinates, for example in Cartesian coordinate system each coordinate specifies the distance from the origin to the point along one axis; in spherical CS one coordinate represents the distance from the origin and other two represent angles from fixed orthogonal axes. In the contrast, the notion of *frame of reference* or *reference frame* symbolises the relation between the observed object and the point from which we observe. For instance, we use egocentric frame of reference when we say "the ball is next to *me*" and allocentric frame when we say "the ball is next to *you*". In general, reference frame may be anchored to practically anything, to our head, hand, or any other object. Within each frame of reference we can define a coordinate system. However, this strict distinction is not always necessary and both terms, frame of reference and coordinate system, are sometimes used interchangeably.

Neuroscientists naturally adopted the concept of reference frames to better understand how space is represented in our brains. It is known that single neurons do not represent positions of objects directly, but the spatial information is distributed over the populations of many neurons in specific brain areas. Cells in the visual system respond to the stimuli located only in particular location called the *response field* of the neuron, or the *receptive field* (RF) in cases where the response is considered to be strictly sensory. Sometimes receptive field also refers to the kind of pattern that causes neuron to response. Brain measurements have shown that primary visual cortex (V1) contains retinotopic neurons, meaning that the receptive fields of adjacent neurons represent points nearby in visual space. This kind of organisation is called *topographic*, but it is not inevitable for neurons to form a map of the space in order to encode information in given frame of reference (Batista, 2002). Exploring the reference frame in which a neuron encodes spatial information is often based on the observation of changes in neuron's response while moving only one part of the body at a time. When the response changes along with the movement, it is interpreted as encoding the space in the reference frame anchored to that body part. For example, the activity of a neuron

encoding the location in eye-centered frame of reference (also called retinocentric) should remain constant as long as the object's image falls on the same locus on the retina, irrespective of the eye or head positions (Soechting and Flanders, 1992).

Encoded spatial information is later used by other parts of the brain to generate and guide the behaviour, for instance to catch a ball or focus on a specific object. The encoding strategies are thus closely tied to the way how the information is processed further. Our focus is on the computational aspect of this processing, that is, what are the computational principles behind transforming spatial information between various representations and how can we model these transformations using artificial neural networks.

2.1 Sensorimotor transformations

The fundamental way how we interact with the world is based on perceiving our surrounding by sensory system (mostly by vision) and manipulating with objects by hands. When we look at the object, the light that falls on our retina induces neural signal that is carried through the thalamus to the primary visual cortex. Neurons in this area and connected pathways use retinal reference frame, which means that our primary information about the object position is specified by its position at retina. Despite the fact that visual information changes with every eye movement, we still perceive the world as stable. This implies that our brain has to take into account also *posture signals* of eye and head position in order to create a stable model of our environment and the objects within. The natural questions arise about how are positions of these objects represented by populations of neurons and whether there are any areas in the brain that encode the space in specific reference frames, for example head or body-centered. The head-centered representation could be theoretically formed by combining the information about eye position and retinal location of a visual stimulus and the body-centered representation could be achieved by combining head, eye and retinal position information (Andersen et al., 1993). Research on this topic that has been continuing for past 25 years suggests that these reference frames does not always exist in an explicit form, but rather as some intermediate representations of space that are further processed for specific purposes, for instance to generate reaching commands. The process of converting sensory stimuli into the motor plans is called *sensorimotor transformation*. It can be seen as a specific case of *multimodal integrations*, which deal with how information from different modalities (e.g. sensory and postural

signal) are processed and used together in the human brain. Sensorimotor transformations are often formalised in terms of spatial transformations from eye or head-centered into shoulder-centered coordinates. The reaching targets may be as well determined by audition or proprioception (the information about some body part, e.g. position of the hand as set of joint angles). It is known that auditory signals are encoded in head-centered coordinates and information about limbs should be encoded relative to the body part. However, there is a recent evidence that sensorimotor system encodes also auditory and proprioceptive targets in gaze-centered coordinates, even though these senses are not fixed to the eye (Blohm et al., 2008).

2.1.1 Eye–hand transformations

Eye-hand coordinate transformation is a nonlinear operation that transforms visual targets into motor plans for reaching. For this operation being accurate, it has to take into account the rotational and translational aspects of body geometry (Fig. 1A). From the computational perspective it is being complicated by the fact that the centers of rotation of the eye, head and shoulder do not align and shift relative to each other with each head rotation. Also because rotation is a non-commutative operation, the transformation has to take place in a stepwise manner.

The eye-hand transformations were closely studied by Blohm and Crawford (2007). They have developed a mathematical model of the 3D transformation for reaching that accounts for body geometry and transforms the retinal desired movement vector (the difference between eye and target retinal location) into a shoulder-centered motor plan using extraretinal signals of eye and head orientation together with the knowledge of eye-head-shoulder linkage geometry. Schema of their model is shown in Figure 1B. To validate the model, the authors realised behavioural reaching experiment, in which human subjects were asked to reach out in complete darkness to a remembered target position while fixating a small light-emitting diode with different head postures. (Unfortunately, the full details of the model and the experiment are beyond the scope of this thesis, but see references for more information.) The analysis of the initial movement direction and experimental data supported the existence of a feed-forward visuomotor transformation for reaching that is geometrically correct. Blohm et al. (2009) later used this model as a teacher to train feed-forward network to perform the 3D transformation for reach. We describe their experiment in section 2.3.2.

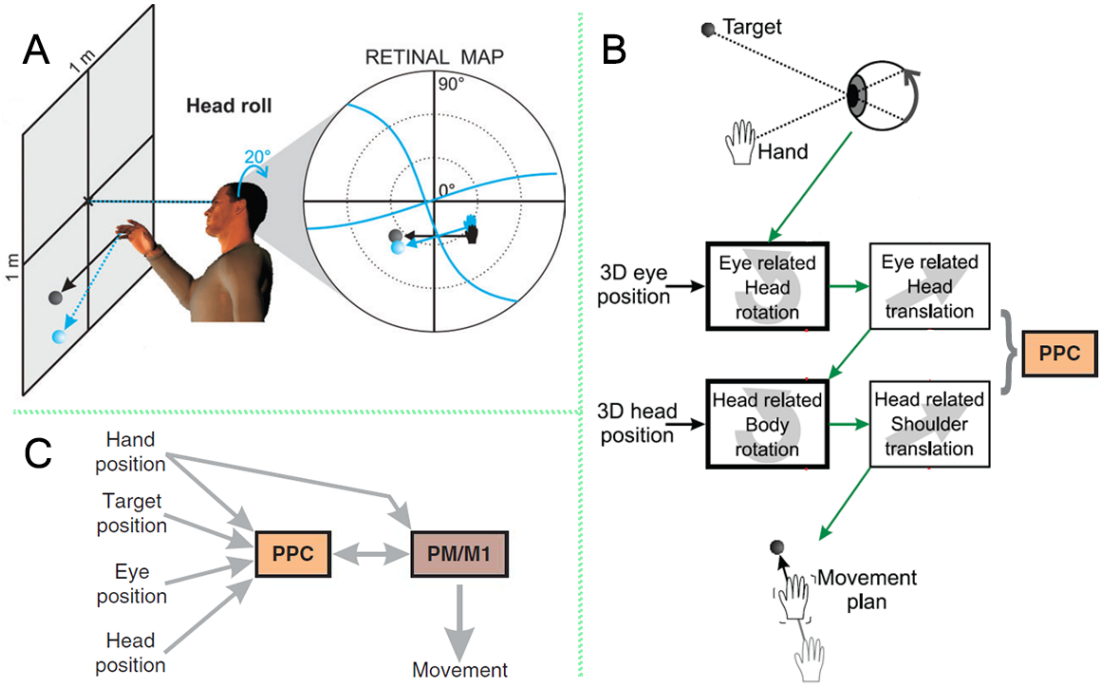


Fig. 1: A) Nonlinearity of the 3D reference frame transformation. The retinal map (right panel) shows the projection of the hand and target (black dot, left panel) as well as the screen horizontal and vertical lines. The blue arrows show the movement vector that would have been generated from the retinal projection of the real hand and target position if the head positions had been ignored. (Blohm et al., 2009)
B) Schema of the 3D visuomotor transformation. The retinal image of the target and hand is rotated and translated using the extraretinal signals of eye and head orientations and internal model of eye-head-shoulder linkage geometry. (Blohm and Crawford, 2007)
C) Assumed model of the visuomotor transformation. Visual (target and hand position), nonvisual (proprioceptive hand position), and extraretinal (eye and head position) information is combined in the posterior parietal cortex (PPC) and also in the premotor and motor cortices (PM/M1) to produce an accurate movement. For comparison with other model, see Figure 22. (Blohm et al., 2008)

2.1.2 First neuroscientific findings

The first influential model of spatial transformations was introduced by Zipser and Andersen (1988). They found out that neurons in area 7a of posterior parietal cortex (PPC) of monkeys combine retinal location of visual stimulus with gaze direction to encode spatial information. The role of PPC as a sensorimotor interface for visually guided eye and arm movements has been also supported by later findings (Buneo and Andersen (2006), Khan et al. (2012)). Cells in PPC appear to nonlinearly combine information from different modalities. Their sensitivity is modulated by one modality (e.g. gaze direction) without changing their selectivity to the other modality (visual stimuli). This phenomenon is called *gain*

modulation and the changes in sensitivity are termed *gain fields* (more in 2.2). The subsequent studies of gain modulation have revealed that it is an extremely widespread mechanism that appears to be a fundamental computational principle behind coordinate transformations (Salinas and Sejnowski (2001), Salinas and Abbott (2001), Chang et al. (2009), Blohm et al. (2009)).

We have already mentioned that the analysis of neuron's receptive fields may be used to indicate the corresponding reference frame. For instance, when neuron's activity differs for different gaze directions and is invariant to head position, we have a reason to believe that neuron represents information in head-centered frame of reference. Surprisingly, several studies of ventral intraparietal area (VIP) have found cells whose receptive fields were partially moving with the eyes (Figure 2). Therefore these neurons were somewhere between eye and head-centered frames. The amounts of partial shifts varied for every unit. Partially shifting cells were also reported for auditory targets in LIP and in the superior colliculus (Deneve et al. (2001), Duhamel et al. (1997)).

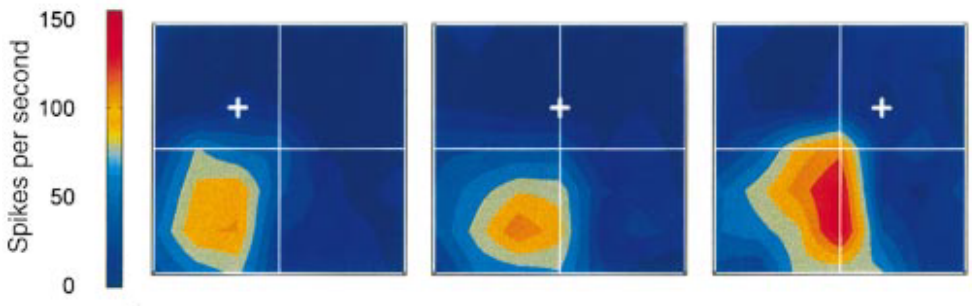


Fig. 2: Partially shifting receptive field. Each image depicts the response of a VIP cell to a visual stimuli for different gaze directions (left, center, right) indicated by white cross. The frequency of discharge is expressed by the colour intensity. The cell's receptive field is moving along the eye position, but only partially (60% of total gaze shift). The activity of cell is higher when gazing to the right, meaning this cell is also gain modulated by eye position. Adapted from Duhamel et al. (1997), Deneve et al. (2001)

Gain fields and partially shifting receptive fields were also found in the hidden layers of artificial neural networks trained to perform transformations between different frames of reference. Examination of the properties of these computational models may give us a better idea of what to look for when exploring the human brain. In the next sections of this chapter we explain the mechanism of gain modulation in more detail and discuss several network architectures that compute spatial transformations.

2.2 Gain fields

The first evidence of modulation visually evoked responses by eye position comes from Andersen and Mountcastle (1983). They tested the neurons within the visual area 7a and the lateral intraparietal area (LIP) and discovered that the neuron's response appeared to be multiplied by gaze angle. The receptive field of these neurons did not change shape nor location, but was only scaled by some gain factor, hence the term *gain field* (GF) (Blohm and Crawford, 2009). This *non-linear* effect is illustrated in Figure 3.

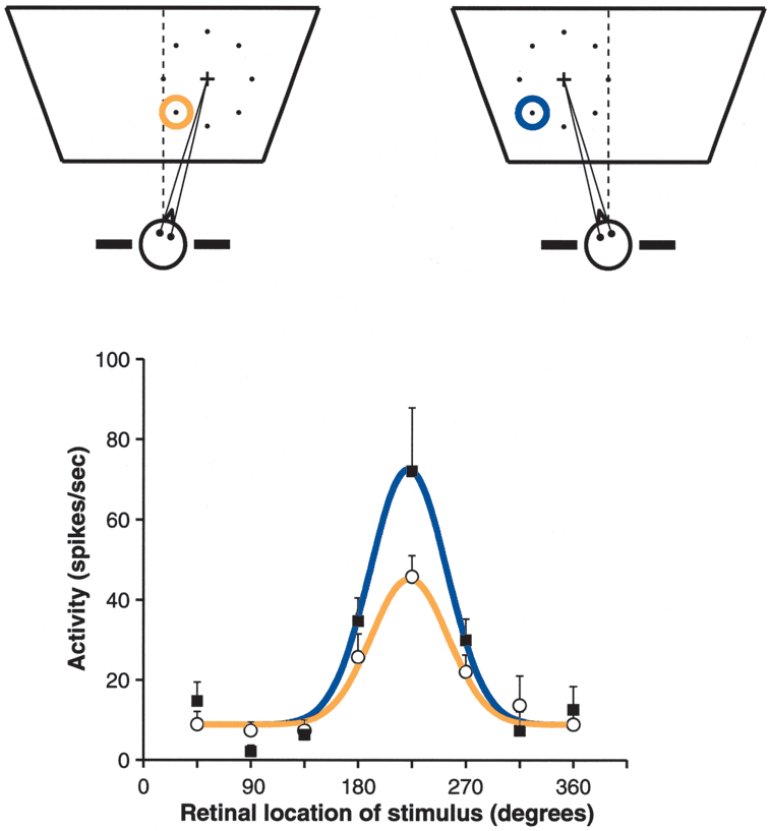


Fig. 3: Neuron's visual responses gain-modulated by gaze angle: Upper diagrams indicate two different eye positions, turned to the right and to the left. The cross corresponds to the fixation point; the 8 dots indicate positions of presented visual stimulus (the rightmost one is at 0 degrees, the topmost at 90 degrees, and so forth). The coloured circles show the position of receptive field of recorded neuron. The bottom graph plots Gaussian fits to the neural responses under two conditions indicated by corresponding colours. The response function changes its amplitude (gain), but not its preferred location or shape. Adopted from Salinas and Sejnowski (2001)

.

Zipser and Andersen (1988) realized that gain fields may play a significant role in the process of visual-motor transformation and trained an artificial neural network to compute the body-centered position of target from eye-centered visual

stimuli and gaze direction. Their network spontaneously developed visual receptive fields gain modulated by eye position similar to what had been observed in PPC. We cover their experiment in more detail in section 2.3.

The subsequent studies have discovered gain fields in many other cortical and subcortical structures, including V1, V3A, the dorsal premotor cortex, parieto-occipital area or V6A, superior colliculus, and lateral geniculate nucleus. They have been postulated for head position in LIP, attention in V4, viewing distance in V4, and eye and head velocity in the dorsal medial superior temporal area. A topographic arrangement of gain fields has been suggested in 7a and the dorsal parietal area (cited from Chang et al. (2009)).

The types of signals that could produce gain fields include gaze direction, head position, eye vergence, target distance, chromatic contrast or attention, all together leading to the suggestion that gain modulation is a general mechanism for multimodal integrations that underlie many important cognitive functions like sensorimotor transformation, object recognition, motion processing or focusing attention (Salinas and Thier (2000), Salinas and Abbott (1997)).

The essential feature of gain fields is nonlinearity. The biophysical basis that allows neurons to combine information from two sources such that their output is close to the product of two functions is still somehow unclear. According to the theoretical studies, one possibility that can give rise to nearly multiplicative gain fields are strong recurrent connections (Salinas and Abbott (1996), Zhang and Abbott (2000)). Situation with recurrently connected neurons is also more realistic in the human brain and can be highly efficient at eliminating neuronal noise. (For recurrent models, see section 2.4.)

2.2.1 Computing with gain fields

The idea of gain fields as a general computational tool has been supported by translating gain fields into mathematical terms by Salinas and Sejnowski (2001). We will expand the example illustrated in Fig. 3. Let x_{target} be the retinal location of the stimulus and $f(x_{target} - a)$ a simple response function of the neuron that has the peak of receptive field located in a. Let x_{gaze} represent the gaze angle. According to the experiments, the response amplitude r can be described though the product:

$$r = f(x_{target} - a)g(x_{gaze})$$

where $g(x_{gaze})$ is the gain field of the neuron. This equation holds for every neuron in the population, except that every neuron has a different receptive field

(a term) and somehow different g function. According to the authors, under some mild assumptions about the functions f and g, the response R of the downstream neuron driven by gain modulated population has the following form:

$$R = F(c_1 x_{target} + c_2 x_{gaze})$$

where c_1, c_2 are constants and F is a peaked function representing the receptive field. This is a key mathematical result, because it implies that a set of downstream neurons may explicitly represent the quantity $x_{target} + x_{gaze}$, while another set may represent $x_{target} - x_{gaze}$, both driven by the same population of gain modulated neurons. Imagine the situation when are you reading a newspaper and want to reach for a mug without shifting your gaze (Fig. 4). The vector for reaching in body-centered reference frame can be computed as the difference between the position of the mug on the retina x_{target} and gaze direction x_{gaze}. Downstream neurons driven on the population of gain-modulated parietal cells can achieve exactly this result (Figure 5).

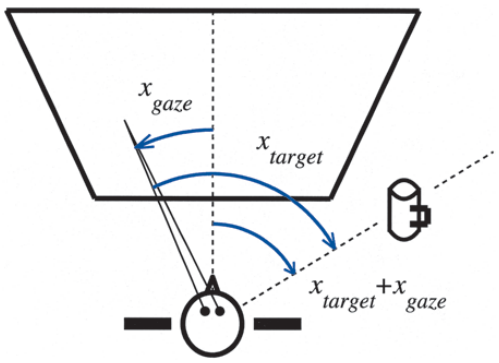


Figure 4: Coordinate transformation while reading a newspaper.
Figure illustrates the transformation from eye-centered to body-centered coordinates in reaching for a mug without shifting the gaze from the newspaper. The reaching vector $x_{target} + x_{gaze}$ does not vary with gaze. (Salinas and Sejnowski, 2001).

2.2.2 Basis functions

The point about the population of gain-modulated neurons that represent stimulus in multiple frames of reference simultaneously was in a more detail elaborated by Pouget and Sejnowski (1997), who observed that without these assumptions many psychophysical and lesion data are difficult to reconcile. They proposed that parietal neurons act as a *basis functions* from which any coordinate frame can by read according to the ongoing task. This was contrary to the very first idea that transformation of object positions is decomposed into the series of intermediate reference frames. However, as we will explain later, the concept of basis function suffers from the course of dimensionality, so nowadays it is believed that the brain compromises between basis function neurons and explicit

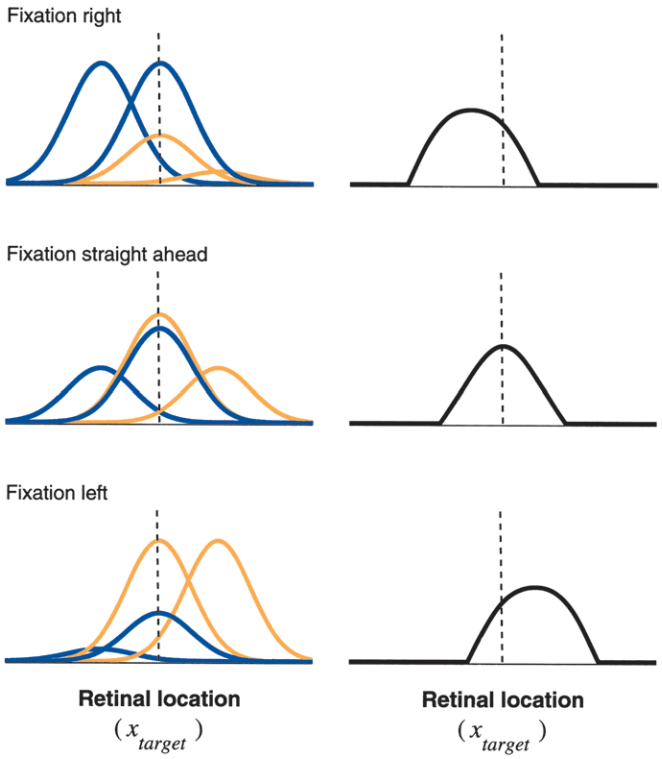


Figure 5: Idealised gain-modulated population leading to coordinate transformation. *Left column:* responses of 4 idealised gain-modulated neurons on fixed visual stimulus. The gain fields of blue ones increases when gazing to the right; orange when gazing to the left. *Right column:* response of the downstream neuron computed as weighted sum of modulated neurons shifts as gaze changes, because it is a function $x_{target} + x_{gaze}$

(Salinas and Sejnowski, 2001)

representations in various frames of references (Blohm et al., 2008).

In the theory of function approximation, any non-linear function can be approximated by linear combination of sines and cosines weighted by numbers called Fourier coefficients. Besides sines and cosines, there are many other basis functions, for matching physiological data are especially promising sigmoids and gaussians, which are a subset of a larger family known as radial basis functions (RBF). Because motor commands are non-linear in nature, they might be generated by a linear combination of basis functions of sensory and postural inputs. Written formally:

$$J = \sum_{i=1}^{N} w_i B_i(V, P)$$

where J is a motor plan, V, P are sensory and postural signals, $B_{1..N}$ are basis functions and $w_{1..N}$ are weights specific to the motor plan being computed.

Pouget and Sejnowski (1997) proposed that the responses of parietal neurons behave like the basis functions of the input signals. This approach has several advantages from computational perspective. First, once basis functions have been computed, the amount of additional computations to obtain motor plan is greatly reduced since it requires only linear projection. Second, the same basis functions can be used to compute many motor plans. Third, forming the basis functions can be accomplished in an unsupervised manner because the choice of basis functions

is independent of output functions being computed. Therefore, the learning of motor plans can be decomposed into two independent stages. The first one, learning basis functions, can be done using variations of the Hebb rule and the second one, learning motor commands, can by done using delta rule (Pouget and Snyder, 2000).

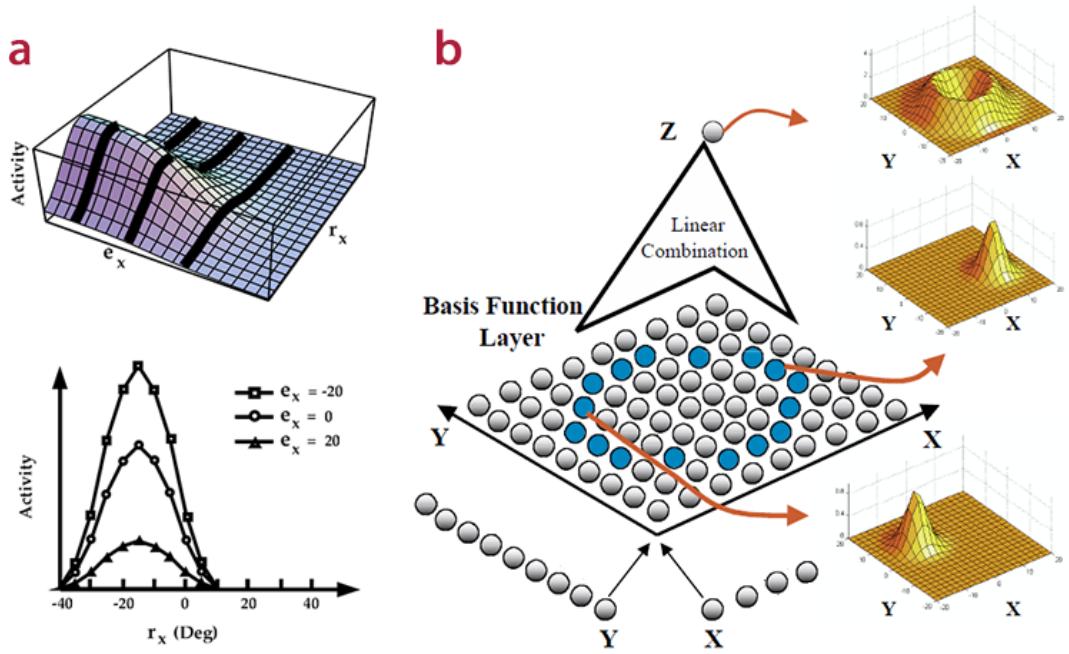


Fig. 6: Basis functions.
a) Activity of a single neuron obtained by multiplying a gaussian retinal location r_x with a sigmoid of eye position e_x (top). Three thick lines representing three different gaze angles corresponds to the visual receptive fields gain-modulated by eye-position (bottom). (Pouget et al., 1999)
b) A neural network implementation of non-linear function. Basis function layer uses gaussians of inputs X,Y to create intermediate representations between input and output layer. For every combination of selectivity for input units there is a corresponding neuron in basis function layer. Two demonstrative response functions are indicated on the right. The output unit Z computes linear combination of basis function units, where the weights from blue units were set to one and all other weights were zeros. (Pouget and Snyder, 2000)

There are two main requirements for basis function hypothesis. First, that basis functions combine their inputs nonlinearly, and second, that there must be units with all possible combinations of selectivity for given inputs. Neurons with these properties were actually found in the parietal lobe, suggesting that basis function representations may be widely used. Thanks to the first requirement, each basis function unit naturally accounts for gain fields (Figure 6a). The basis function layer with all combinations of selectivity for two inputs is illustrated in

Figure 6b. This figure also shows how a non-linear function of two inputs (X,Y) can be computed through the intermediate representation using gaussians as basis functions. The same architecture can be used for coordinate transformations. Imagine that input X encodes the target position in eye-centered reference frame and input Y the eye position in head-centered frame. Every cell in the basis-function layer is sensitive to the specific eye and target position, meaning that it has only one peak located at the position determined by this configuration. The high unit activity clearly indicates specific target position in head-centered reference frame. The same target position may result from different configurations, therefore the output unit sensitive for the desired target position needs to consider all corresponding units from the basis-layer. This example is more formally explained in section 2.4.2.

The unresolved issue about basis-function representation is that the number of neurons required increases exponentially with the number of signals being integrated. Hence a basis function map using 10 neurons per signal and integrating 12 signals would require 10^{12} neurons, more than total number of neurons available in the cortex. One solution might be to use two modules of basis functions connected in hierarchical fashion. (Pouget and Snyder, 2000)

The additional evidence in favour of basis function comes from the study of *hemineglect*, a syndrome caused by unilateral lesions of the parietal lobe. Patients suffering from hemineglect experience difficulty processing or reaching to stimuli located in the hemispace contralateral to their lesson. For instance, patients may fail to eat food located on the left side of their plates or shave the left side of their face. Neglect patient typically fail at the line cancellation task presented in Figure 7. In the context of reference frames, one may ask to what reference frames does 'left' relate, because it can be defined with respect to the eyes, head or body. We may try to determine the neglect reference frame by assessing the neglect in a variety of body postures and stimulus locations, for example we may ask the patient to turn his head left and eyes to the right while the stimulus lies in front of him. Similar studies revealed that neglect actually affects multiple frames of reference. This is exactly what happens when we introduce lesion into the basis-layer in the network model illustrated in Figure 6b. Additionally, it was demonstrated that the model based on this architecture can capture another two essential aspects of hemineglect: it can reproduce the pattern of line crossing of parietal patients in line cancellation and accounts for relative and object-centered neglect (Pouget et al., 1999).

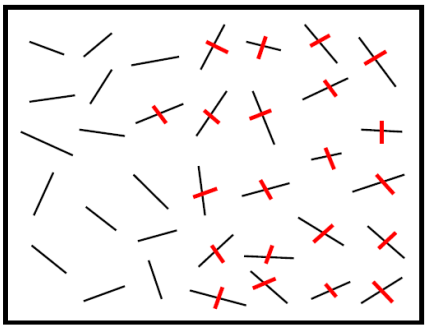


Figure 7: A typical result for a line cancellation test in a left-neglect patient. The patient was asked to cross out all line segments on a page, but he failed to cancel the lines on the left side. (Pouget et al., 1999)

2.2.3 Compound gain fields

In the example of reaching for a mug while reading the newspaper (Figure 4) we operated with the simplifying assumption that the initial hand position is at the centre of body-centered coordinate system. In reality, however, it is necessary to also consider hand position relative to the body. This situation was questioned by Chang et al. (2009) and led to the interesting findings that neurons in parietal reach region (PRR) have eye and hand gain fields that are similar in magnitude but opposite in direction. Their conclusion was supported both by measurement of PRR area and the computational model built on a simple three-layer feed-forward neural network. The negative correlation between the hand and eye gain fields strongly supports the functional role of gain modulation in the computation of the reach plan. It also suggests a new view on generating movement vector in hand-centered reference frame. Instead of sequential transformations from eye through body to hand-centered frames of reference, it seems that PRR neurons use only one step, implicitly performing all needed comparisons between the coordinate systems (Figure 8). The authors named this gain fields as *compound* eye-hand distance gain field, because their effects are indistinguishable from a single gain field for the distance between the gaze location and the initial hand position.

2.3 Feed-forward models

This section is dedicated to the models based on feed-forward artificial neural networks (FF-ANN) designed to study computational aspects of sensorimotor transformations. One of many reasons why these models are being built and studied is that they are useful tool for exploring the basis of real brain (dys)functions. In this sense, the breakthrough for a computational neuroscience came with the work of Zipser and Andersen (1988) as they showed that measured neurophysiological properties of real neurons may underlie nontrivial computations. We will

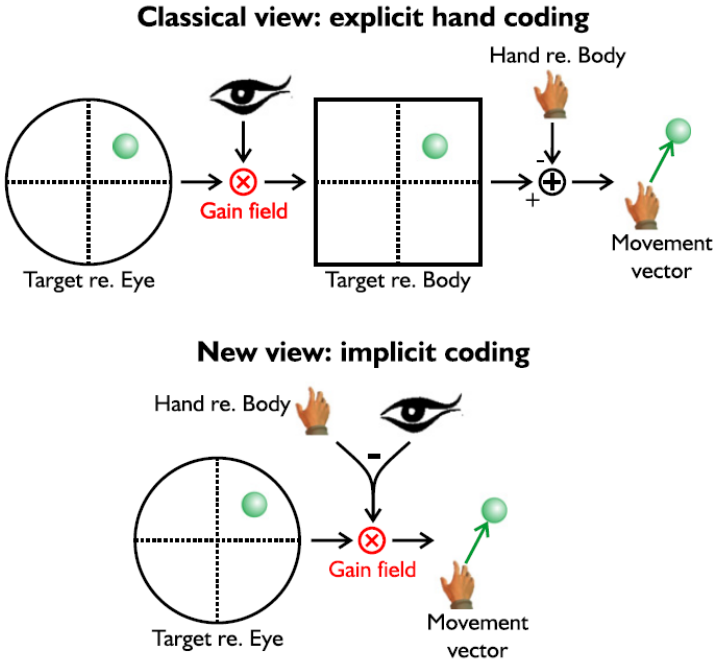


Figure 8: New view of movement planning: In classical view, the transformation from eye-centered target position to hand-centered movement vector is mediated by body-centered target encoding. In the new perspective, the movement vector is generated directly by compound gain modulation of eye and hand, where these gain fields have the same strength but opposite direction (Blohm and Crawford (2009), Chang et al. (2009))

shortly introduce their experiment.

The physiological data for their experiment comes from the measurements of neurons in area 7a in macaque monkeys. The neural activities were measured presenting the same visual stimulus while fixating the gaze at one of nine predefined positions. Figure 9-1 shows the experimental protocol and the observed effect of gain modulation. The authors used back-propagation algorithm (3.2.1) to train three layer ANN for computing coordinate transformation from eye-centered to the head-centered reference frame using the information about the head-centered eye position (Figure 9-2). The hidden layer of the network exhibited gain-modulated receptive fields very similar to the ones found in parietal neurons. This obviously does not mean that we can conclude that back-propagation algorithm is used in the brain, but interestingly enough, it can lead to the same (or strikingly similar) computational principle. Therefore, we can assume that we have a suitable tool for further studies.

2.3.1 Multimodal integration in 2D

From the point of view of input-output mappings, the multimodal integration in PPC may take many forms and perform various types of transformations. For instance, the desired transformation may be from purely sensory inputs to the head-centered representation, or there might be multiple desired output representations at once, computed from both sensory and postural inputs. The questions

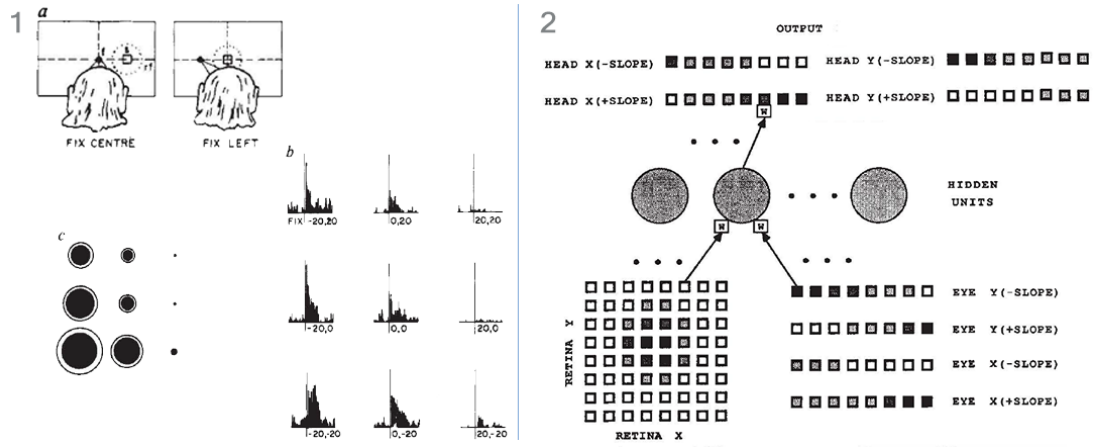


Fig. 9: 1) Experimental protocol for determining gain fields in macaque monkeys.
a) Monkey fixates at one of 9 predefined positions and sees a visual stimulus present always at the same retinal location. In figure only two fixations are shown.
b) Peri-stimulus histograms located in the same relative positions as the fixations that produce them.
c) Graphical illustration of data 1b, the outer circle diameter corresponds to the overall activity and the diameter of inner darkened circle illustrates the gain field.
2) The network architecture for coordinate transformation. The input layer consist of 64 visual units (bottom left) and four sets of 8 units for eye-position that is encoded as positive and negative horizontal and vertical slope. The output layer has the same structure for encoding head-centered position. The hidden units receive connections from all input units and project to every output unit. The hidden and output units have sigmoidal activation functions and the receptive fields of input units are gaussians. (Zipser and Andersen, 1988)

we could ask is whether there are some intrinsic differences between these various transformations, or whether there are some common principles involved. To answer these questions, Xing and Andersen (2000) constructed several sets of ANN models and performed reference frame analysis based on the shifts of receptive fields. In this section we will discuss the details of their work.

Models description

Each model was a standard three-layer feed-forward ANN (see 3.2). There were four types of possible input units:

V - visual map: 8×8 array of units for visual input. Receptive fields of units were gaussians with $1/e$ width of $15°$ and spacing $10°$ over the visual input $80° \times 80°$

A - auditory map: similar to V, except that the input was encoded in head-centered coordinates

E - eye position: four sets of 8 units; two sets encoded positive and negative slopes of the horizontal component, the other two sets encoded the vertical component in the same manner. The head-centered positions were within each set encoded linearly.

H - head position: similar to E, except that the input was encoded in body-centered coordinates

The hidden layer consisted of 20 hidden neurons. Activation functions for the units in the hidden and output layer were standard sigmoids. The output layer contained several representations that were modelled by 8×8 array of units as in the input layer:

ME - motor error in eye-centered coordinates (eye-movement)
ME = V or alternatively when V is not present, ME = A - E

HE - head-position error in head-centered coordinates (head-movement)
HE = V + E; HE = A

BE - body-position error in body-centered coordinates (body-movement)
BE = V + E + H; BE = A + H

Models were divided into four sets according to the level of multimodal integration, see Fig. 10 for the description. All models were trained by back-propagation algorithm (3.2.1). The input was picked at random and consisted of eye position and either a visual or an auditory target. In model 4, the head position was added in the same manner as eye position. Training stopped when the mean squared error could not be further decreased and the accuracy of network was less than 4°. (The performance error of 4° was typically used for training monkeys to make saccades.)

Measurements and results

After the training, all networks were capable of computing given mapping. Gain fields were observed in all models that actually performed some coordinate transformation (all except 1-1 and 1-2). Common way how to visualise GF is to plot the unit's response to a target presented in the receptive field against the different eye positions (Fig. 11A). Receptive field of the hidden unit was defined as the input area that evokes a response greater than 50% of unit's maximal response (Fig. 11B). The relationship between unit's gain and receptive field was examined by comparing their directions. The GF direction is the direction to the best-tuned unit relative to the central eye position (Fig. 11A) and RF direction

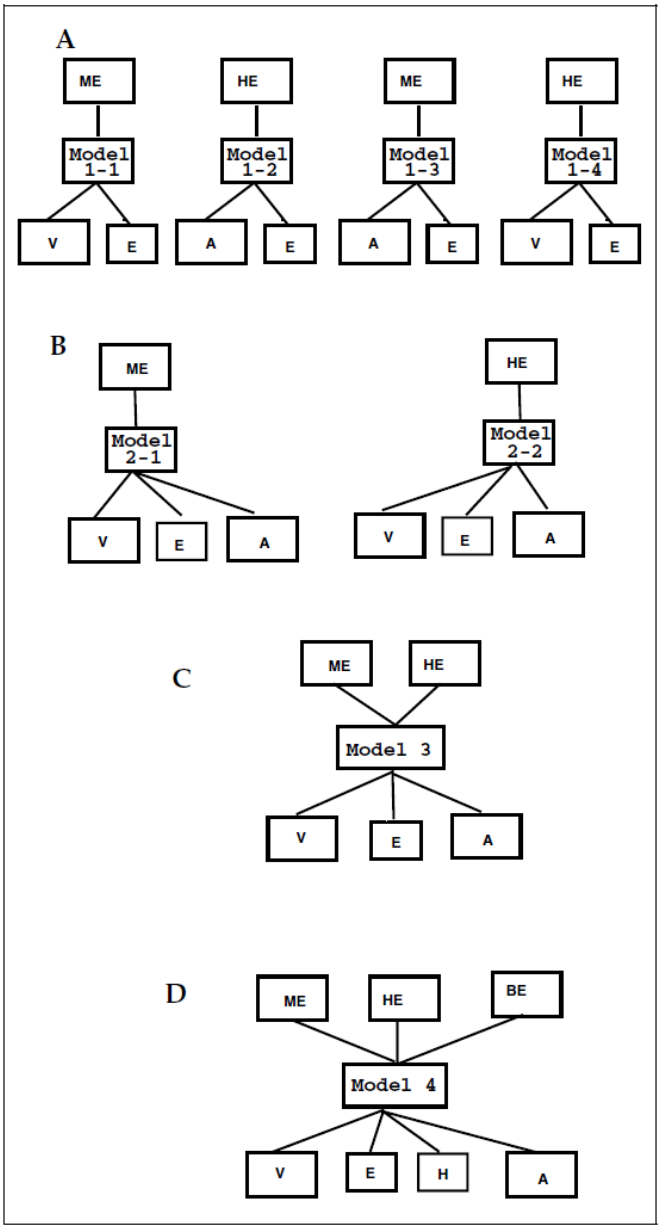


Figure 10: Models of four different levels of integration in PPC.

A) Level-1 models for unimodal neurons. Model 1-1 and 1-2 simply remap input signal onto the output layer, models 1-3 and 1-4 perform coordinate transformations ME = A - E and HE = V + E.

B) Level-2 models use bimodal inputs. 2-1: ME = V and M = A - E. 2-2: HE = V + E and HE = A.

C) Level-3 model has bimodal inputs and multiple output representations. ME = V, M = A - E and HE = V + E, HE = A.

D) Level-4 model has additional access to head position and its output layer contains body-centered map. BE = V + E + H, BE = A + H.

(Xing and Andersen, 2000)

was computed as the center of mass of the unit's response across the input map (Fig. 11B). The angle between these two directions served for testing whether GF and RF are aligned in the same or opposite way. The reference frame analysis of the hidden units was based on the idea, that if units encode in eye-centered frame, then the RF should shift along with the eye-movement and stay unchanged in case when the head-centered frame is used. To this purpose, *RF shift ratio* was computed as the distance between the centers of mass of RF profiles measured at different eye position (Fig. 11D).

Receptive fields in the first two models, 1-1 and 1-2, were clearly anchored to the eye and head respectively (the RF shift ratio was close to one). This result

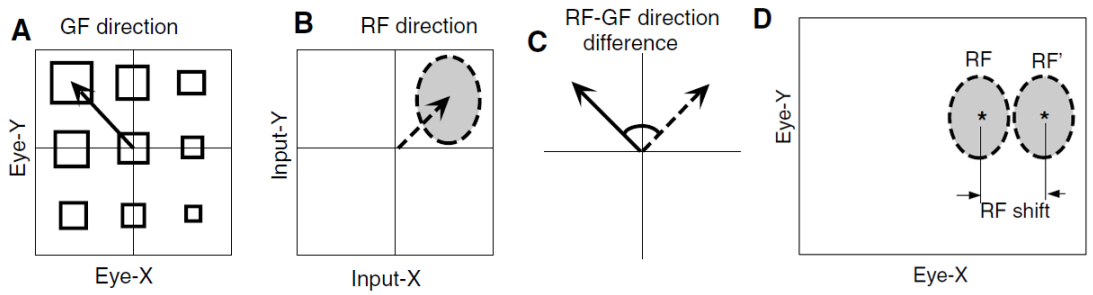


Fig. 11: A) Gain field: the size of squares corresponds with the amplitude of unit response to a visual stimuli and different eye position. The GF direction (arrow) points to the unit with maximal response. B) Receptive field: dashed ellipse represents the input area that causes neuron to response. The RF direction is indicated by the arrow. C) The difference between RF and GF directions. D) The change of eye-position makes receptive field to shift. The RF shift ratio was computed as the distance of the RF centers divided by the change in eye position. (Xing and Andersen, 2000)

was expected as these two models do not perform any multimodal integration. Networks 1-3 and 1-4 (Zipser–Andersen model) developed localized gain fields and receptive fields that shifted horizontally and vertically with eye position. Fig. 12A-C shows typical RF and GF of one hidden unit, we can see that they have opposite directions. Horizontal and vertical RF shift ratios were calculated for all hidden units and plotted in the same histogram (Fig. 12D). The mean shift ratio at 0.5 indicates that the hidden units do not encode coordinates in any exclusive frame of reference, but rather in some intermediate representation. The authors also noticed the contrast between GF-RF relationships: while in model 1-3 were GF approximately opposite to the direction of RF (Fig.12), model 1-4 had GF tuned for the same direction as RF. We assume that this is just a consequence of the simplified model, since it does not account for the fact that retinal image is flipped.

The second level networks integrated auditory and visual inputs and therefore developed localized auditory and visual receptive fields (ARF, VRF). The response properties of the hidden units in these models were similar to those in models 1-3 and 1-4, because these two pairs of models performed the same transformations (from head to eye-centered and vice versa). Both ARF and VRF have been found to partially shift with eye-position. Note that model 2-1 did not need to perform any coordinate transformation for visual inputs, since the output was in eye-centered frame. The VRF were slightly partially shifting even in this case.

Model 3 mapped visual and auditory inputs into both eye-centered and head-centered frames. The RFs and GFs were similar to those in previous models and hidden units encoded both visual and auditory targets in intermediate reference

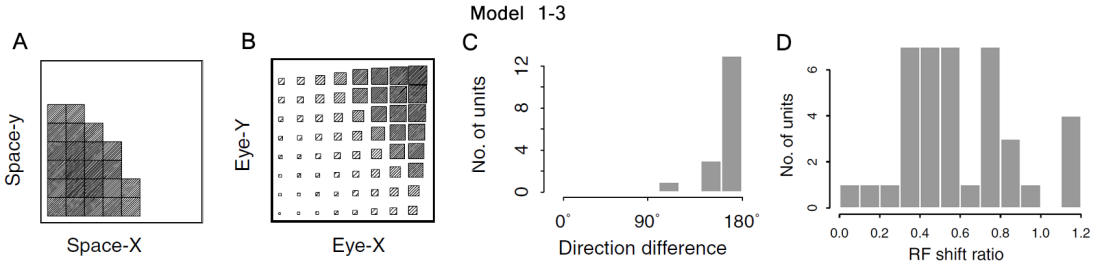


Fig. 12: A) Typical receptive field of one hidden unit in model 1-3. The neuron is active when the target is present in the area illustrated by gray squares.
B) The gain field of the same unit as in A. The gray level and the size of the squares represent the activation of the unit at different eye positions. The target was always present at the same location in the RF.
C) The histogram of direction differences of the gain field and the receptive field of all hidden units in the model 1-3. Most units have direction differences close to 180°, i.e. they are on opposite sides like in A and B.
D) The histogram of vertical and horizontal RF shift ratios of all hidden units. Ratio close to zero indicates the head-centered reference frame, ratio close to one the eye-centered frame. Because the mean shift ratio was 0.5, most hidden units encode location in intermediate coordinate frames.
(Xing and Andersen, 2000)

frames. The RF shifts were similar to those in second level models. Therefore, the measurement of RF shifts cannot differentiate neurons that provide a single-output representation from neurons that provide multiple-output representations. In addition to model 3, the output layer of model 4 contained body-centered output map. As a result, hidden units developed also head position GF from which the majority was in the same direction as eye GF.

To sum up, gain modulation was present in all models that performed two dimensional coordinate transformation; the hidden units encoded targets in intermediate reference frame and a single feed-forward network is capable of computing target representation in several frames of reference simultaneously.

2.3.2 Visually guided reaching in 3D

Up to now, we discussed models for sensorimotor transformations that operated with several simplifying assumptions. Above all, they performed only 2D coordinate transformations and the actual motor plan was omitted. As we explained in section 2.1.1, eye-hand transformations in 3D space are complex and non-linear operations. Therefore we cannot assume that the properties arisen from 2D simulations will hold up also in 3D and our understanding of electrophysiological experiments remains limited. To investigate more realistic scenario of how neu-

ral networks compute motor plan for 3D reach, Blohm et al. (2009) constructed a four-layer feed-forward neural network and trained it to perform visuomotor transformation from gaze-centered inputs to a shoulder-centered output. Their work deserves our best attention as we haven't found any other similar experiment. However, because the experiment was very comprehensive, we will focus only on the most important details and results. At first we briefly describe the model architecture and the training procedure; later we introduce three techniques used for comparing input output properties of individual units; and at the end we conclude that hidden layers (and even individual units) show different reference frames when tested using different methodology.

Model architecture and training

The network architecture is illustrated in Fig. 13. The input layer consisted of seven distinct inputs that fully describe the body geometry. The inputs were: retinal target position, retinal target disparity, retinal hand position, retinal hand disparity, eye position, head position and vergence. Retinal positions were represented by topographical maps with uniformly distributed gaussian receptive fields. Retinal disparities were also represented by topographical maps, but the tuning curves were given profiles similar to those found in monkey neurons (similar to the product of two gaussians). Eye-in-head and head-on-body positions were characterized by 3D angle vector representations, where every component was encoded by two inputs for positive and negative rotations. Both these positions were therefore coded in 6 inputs in a linear manner. The vergence angle was coded similarly in one input.

The activation functions for the units in hidden layers were sigmoids. Several numbers of units in the second layer were examined, ranging from 9 to 100. Presented results come from the network with 36 hidden layer units (HLU). The third network layer (population output) consisted of 125 cosine-tuned units with preferred directions randomly, uniformly distributed on a unit sphere. This layer was assumed to code movement direction in extrinsic (shoulder-centered) coordinates as have been observed in monkeys. The output (read-out) layer consisted of 3 units that coded movement in space (horizontal, vertical and posterior-anterior direction). Note that the weights between population output and read-out layer were not adapted by training process, but calculated in a very specific manner that reflected the implicit assumption of cosine-tuned units in the third layer.

The training set consisted of 500 000 patterns randomly generated by the model for 3D eye-hand transformations described in section 2.1.1. As a training

algorithm was chosen resilient back-propagation (3.2.4) and the training stopped when the gradient of root mean squared error became $< 10^{-6}$.

The trained network was capable of performing visuomotor transformation with accuracy similar to human subjects. The majority of absolute reach errors for the network with 36 HLU were smaller than 10 cm and the mean was 6.4 cm.

Neural network analysis

As well as in previous 2D experiments, the goal of analysis was to identify the mechanisms of reference frames transformation and investigate input output properties of individual units. The analysis of input properties was based on examining RF shifts and the output properties were investigated by two neurophysiological techniques: *motor fields* and *microstimulation*. We will explain them in turn.

A general observation about all hidden units was that their responses were largely gain-modulated by eye, head and hand positions. The analysis of units input properties applied the same rule as we have seen before: if the visual RF encodes targets in shoulder-centered coordinates, then the center of mass should shift in the direction opposite to the eye orientation. In the first hidden layer (the second layer of the network), the centers of mass of RF did not shift. This observation was interpreted as a gaze-centered encoding scheme (represented by an eye icon in Fig. 13C). On the contrary, the third layer also contained units with shifting receptive fields. Figure 14A shows receptive field of one such unit across various horizontal and vertical eye positions. The center of mass is depicted as magenta square with black border. To obtain the entire representation of these shifts, the eye position was changed in a systematic fashion (5° horizontally and vertically) and all centers of mass were plotted in one diagram (B). Further quantifications of RF shifts for all hidden units were made by regression analysis that provided horizontal and vertical gains of the centre of mass. These *shift gains* can be used to indicate the reference frame in the very similar manner as the *RF shift ratio* in 2D experiments. Horizontal and vertical shift gains are plotted in panels C–D and their combination in panel E. The same measurement was realised on the networks with various numbers of hidden units and plotted in one diagram (F). We can see a broad distribution of gain values in all networks, resulting in a conclusion that neurons in the third layer use an intermediate frame of reference between eye and shoulder coordinates.

Output properties were firstly analysed by examination of unit's *motor fields*. Motor fields provide information about the unit's contribution to the motor output, that is, how the unit's activity changes as a function of the movement pro-

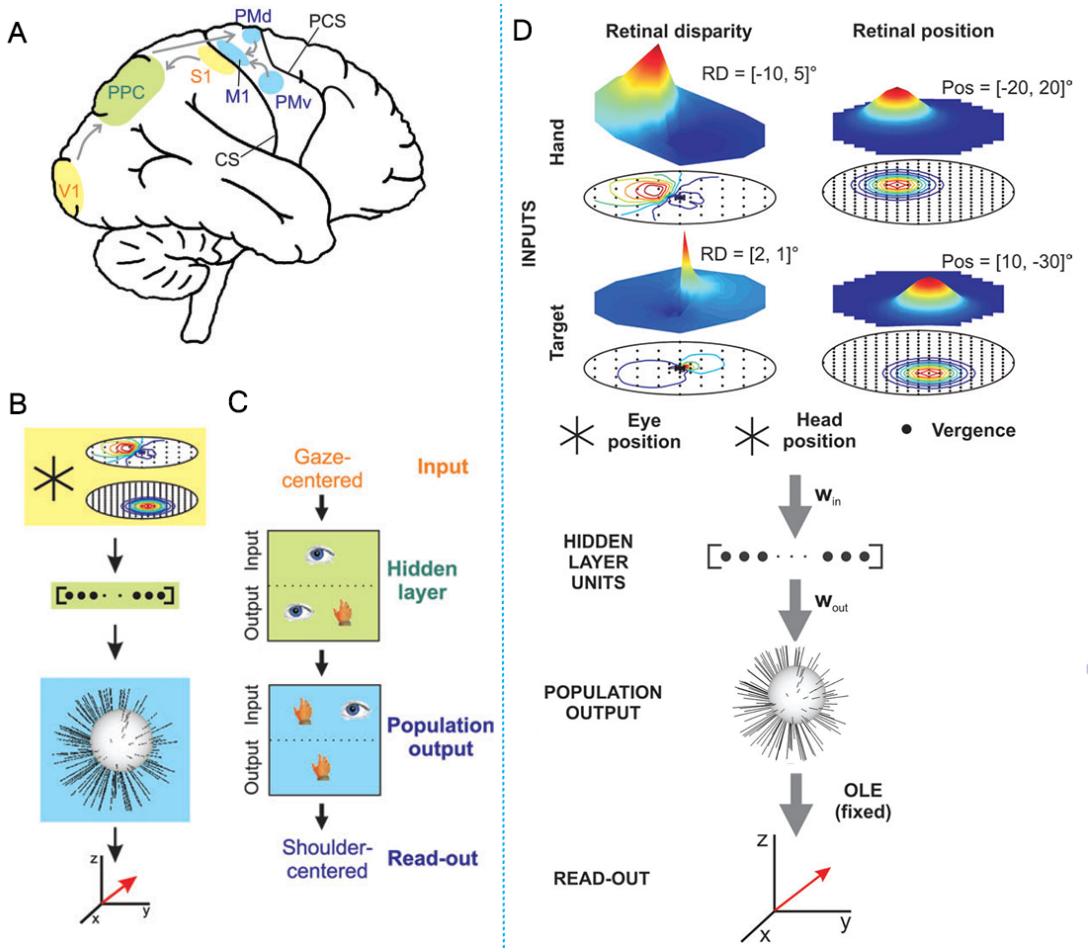


Fig. 13:
A) Brain structures known to be part of the visuomotor transformation pathway in the brain. V1 - visual cortex (gaze-centered hand and target positions); PPC - presumably the hidden layer of neural network model; S1 - somatosentory cortex (a potential source of the extraretinal eye and head position signals); PMd/v, dorsal/ventral PM cortex (the hypothetical population output); M1, primary motor cortex;
B) Neural network implementation of the different brain structures.
C) Interpretation of how reference frame transformations might be performed in distributed computing. The eye icon stands for gaze-centered coordinates and the hand icon represents shoulder-centered coordinates. The presence of both icons depicts a spread of reference frames between and beyond gaze- and shoulder-centered coordinates. Same colors in panels (A-C) refer to corresponding levels of processing.
D) Neural network model with four layers. Both hand and target positions in the input layer are represented by 2 two-dimensional maps: cyclopean retinal position and retinal disparity. The hidden layer consisted of 9-100 neurons. Population output layer had 125 units with random preferred movement directions in shoulder-centered space. Read-out layer coded three components of the shoulder-centered movement in 3D space. See text for more information.
(Blohm et al., 2009)

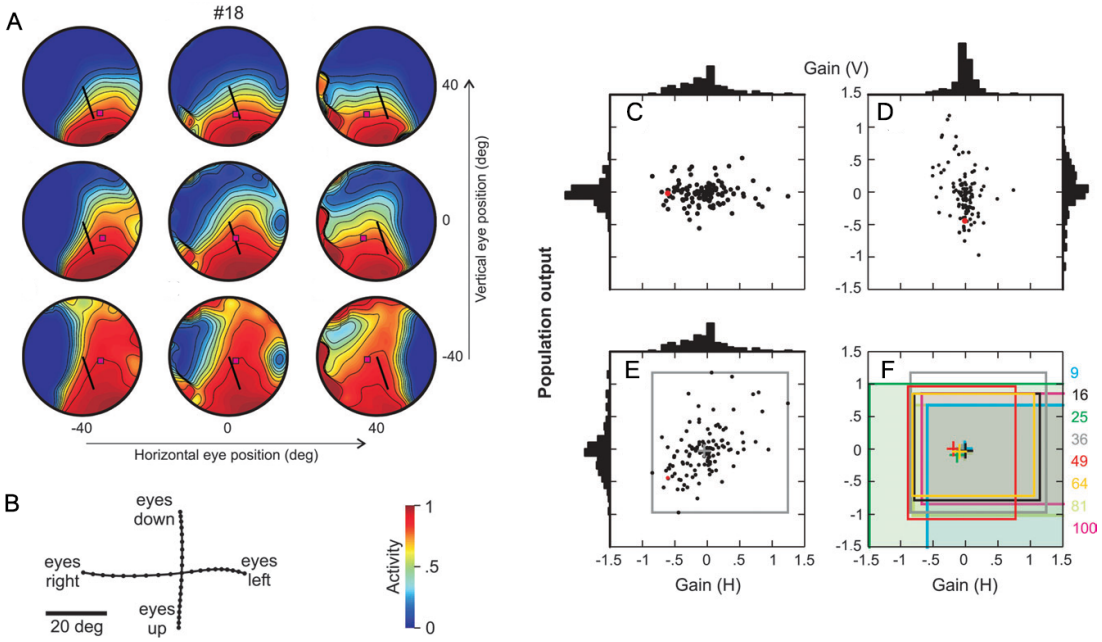


Fig. 14: A) Visual receptive field of hidden unit #18 is modulated as a function of eye positions. Magenta square denotes center of mass that clearly shifts.
B) Representations of the vertical and horizontal shift of the center of mass as a function of eye positions in 5° steps.
C, D) Horizontal and vertical shift gains provided by regression analysis for all units. Each dot represents one hidden unit in third layer.
E) The combination of horizontal and vertical gains. The gray square indicates the range of obtained gain values for the hidden network with 36 HLU.
F) Summary of gain values through the population codes of all networks. The coloured numbers shown to the right of the graph indicate the network sizes.
(Blohm et al., 2009)

duced. To compute motor fields it is necessary to produce movements in all 3 dimensions and measure a unit's activity related to those specific movements. We say that unit has a preferred direction if it preferentially participates in generating movements to the specific location in space. The changes of motor field amplitude and preferred direction can be analysed in the same manner as visual receptive fields. Such analysis resulted into the observation, that it was not possible to identify any reasonable reference frame in any of the hidden layers. This was surprising particularly for the second hidden layer, because the weights to the read-out layer were computed prior to the training, so the network was indirectly designed to encode movement vector in shoulder-centered coordinates.

Another method to assess the output properties of individual units was to simulate *microstimulation* in the network. Microstimulation consisted in setting the specified unit's activity artificially to the value = 2 and observing the effect of eye position on the generated movement vector. To determine only the effect

of eye position, other inputs were chosen such that the network would naturally not produce any movement. The analysis produced four typical results that are shown in Fig. 15. Black lines represent movement vectors obtained for different eye position ranging horizontally from $-45°$ to $45°$ in $5°$ steps. The first typical result was a fixed vector (A). Since the generated movement did not depend on eye position, the fixed vector indicates shoulder-centered coordinates. The second typical behaviour is shown in panel B. Here, the unit shows gaze-centered coordinates because the movement vector followed the eye position. Other units showed intermediate behaviour (C) between A and B. Finally, there were units for which the evoked movement vector converged at a particular location (D). According to the results from subsequent regression analysis, the first hidden layer uses a mixture of different reference frames intermediate between the gaze and shoulder-centered coordinates and the population output layer uses only shoulder-centered coordinates.

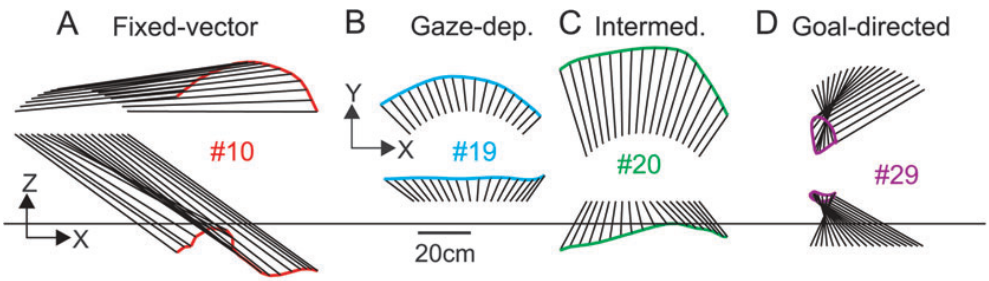


Fig. 15: Reference frame analysis through microstimulation. Each line represents one movement vector generated under different horizontal eye positions. The end points of movements are connected by coloured line. The top part of each panel represents a view from above, the lower part view from behind.
A) Fixed vector unit: movement vector does not change (interpreted as shoulder-centered)
B) Gaze dependent unit: movement vector follows eye position
C) Intermediate unit has behaviour between A and B.
D) Goal directed unit: movement vector converged at particular location.
(Blohm et al., 2009)

So far we analysed unit's reference frames focused only on gaze versus shoulder-centered coordinates. Including the head movements can possibly reveal units coding in head-centered reference frame. To be able to discriminate between three possible encodings (eye, head, shoulder-centered), it was necessary to perform analysis under three different conditions, that is, eye-only movements, head-only movement and opposite eye-head movements. (Opposite eye-head movements are characteristic for vestibulo-ocular reflex (VOR)). This reflex stabilises images on the retina during the head movement by producing eye movement in a different

location.) The authors used the same analysis techniques as described above and evaluated unit gains for changes in eye, head and VOR movements. These three values were later used to determine a point in space where each gain corresponds to one axis. In this space, we can make some predictions about the reference frames, for instance, in an analysis based on RF shifts we would expect that if unit uses head-centered encoding scheme, then its gain for head-movement will be zero, for eye-movement -1 and for VOR movement equal to 1. This prediction (0,-1,1) also determines a point in space. Units with gains close to this prediction are assumed to use head-centered coordinates. Given three predictions for each reference frame, we may define a plane and make orthogonal projections of units gains onto this plane. Results of this analysis for all three physiological techniques are visualised in Figure 16. We see that the results copy the previous findings, but in addition some of the units show behaviour close to head-centered when probed using motor fields and microstimulation. This was remarkable because there never was any explicit head-centered encoding in the network's input or output. (Note that analysis considered only horizontal movements, but qualitatively the same results were observed for vertical movements.)

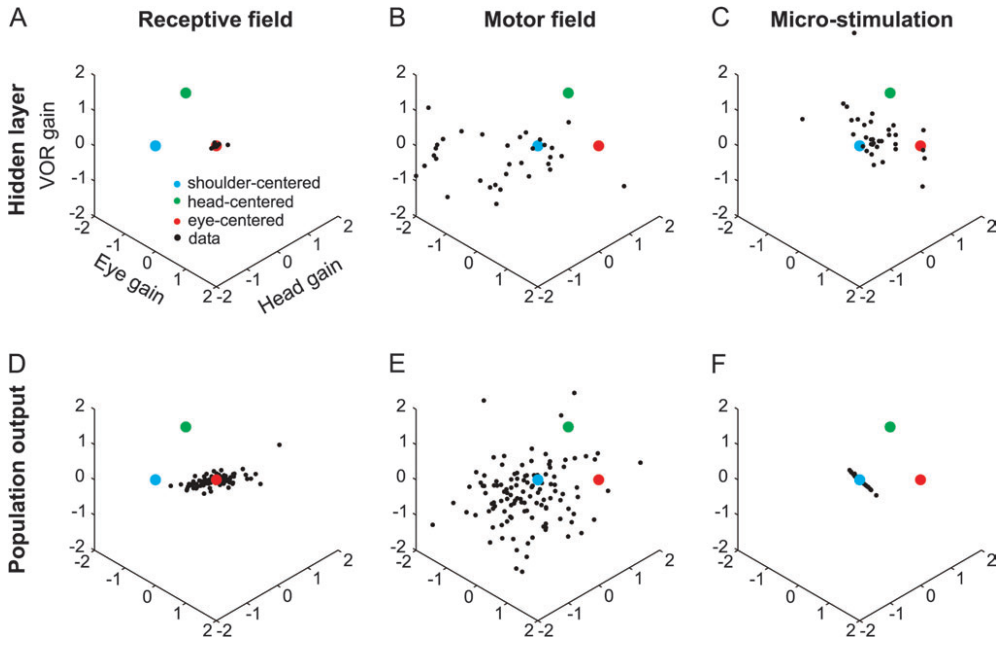


Fig. 16: Complete reference frame analysis across three electrophysiological techniques. Black data points represent individual units in the first hidden layer (top) and population output layer (bottom). Coloured data points represent predictions for eye-centered (red), head-centered (green), and shoulder centered (blue) coordinates. The view of the 3D plot was chosen to be ortoghonal to the plane of the 3 predictions. (Blohm et al., 2009)

Conclusions

The four-layer feed-forward artificial neural network was able to perform the 3D visuomotor transformation from gaze-centered inputs to a shoulder-centered output. The network with 36 hidden-layer units achieved accuracy very similar to the accuracy of human reaching and therefore was chosen for presenting the results. The activity of many units in hidden layers was largely gain-modulated by eye, head and hand positions. Analysis by different techniques revealed that hidden layers and even individual units have different input-output coding properties. This is schematically depicted in Fig. 13C. For example, units in the first hidden layer showed purely gaze-centered visual receptive fields, but their output properties displayed reference frame intermediate between eye and shoulder-centered coordinates. Due to this input-output relationship, each unit performed a fixed input-output transformation. The authors hypothesized that the contributions of these individual transformations are combined by gain modulation mechanism in a way to accurately produce the overall transformation.

In addition, the work by Blohm et al. (2009) provides several other interesting findings. The trained network was able to reproduce and explain many findings of real neurons in the frontal-parietal network, which supports the neurophysiological significance of their work. As a general methodological implication, the authors stressed the importance of multivariate analysis in performing discrimination between potential reference frames, because different experimental techniques can lead to different observations. In comparison with the network model based on basis functions (see 2.4.2), the most noticeable difference was that the RF of the first hidden layer units never shifted. However, as stated by authors, shifting receptive fields are inconsistent with PPC data.

2.4 Recurrent models

Recurrent connectivity is a well described feature of cortical circuits and has been deeply studied also in recurrent neural networks (RNN). It is a class of networks where connections between units form directed cycles that allow the network to create internal states and exhibit dynamic temporal behaviour. Their computational power is equivalent to Turing machines, however, the training is somewhat more complex and can be tricky especially for a large number of units. In the context of sensorimotor transformations, recurrent connections were firstly the subject of study because of their ability to produce multiplicative gain fields. Later they were used in a combination with basis functions to create

an interesting model that gives us better insight into the actual neural basis of multisensory spatial representations. We describe this model later in this chapter, after the short explanation of a simple recurrent network architecture that produces multiplicative gain fields.

2.4.1 Multiplicative gain fields

A biologically plausible mechanism that would allow single neurons to perform product operation of their inputs is still waiting for being fully explained. However, multiplicative responses are an emergent property of the network with recurrent synaptic connections even though single neurons are not capable of computing a product of its inputs.

The mathematical model studied by Salinas and Abbott (1996) was for convenience constrained to one dimension. Modelled parietal neurons received two kinds of inputs: external input representing retinal location of visual stimulus and gaze direction, and recurrent input from neighbouring neurons (Figure 17).

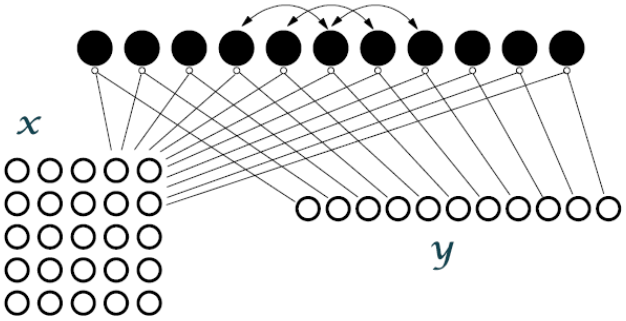


Figure 17: Simple recurrent model for multiplicative gain fields. Parietal neurons (filled circles) receive two kinds of inputs: external inputs of visual stimuli (x) and gaze direction (y), and recurrent inputs from their neighbours.

The effect of recurrent connections was excitatory for near neurons with overlapping receptive fields and inhibitory for neurons further apart with separated receptive fields. This is a common feature of recurrently connected cortical models. Receptive fields of neurons were modelled as gaussians and their preferred locations spanned the full range of possible visual stimulus locations. The sensibility for gaze direction was a linear function and was the same for every cell. We will describe the overall architecture in mathematical terms. Given the neuron i, location of the visual stimulus x and gaze direction y, the external input h_i to the neuron has the following form:

$$h_i = h_i^V(x) + h_i^G(y)$$

The visual input h_i^V is the gaussian function of the difference between the stimulus location x and neuron's preferred location x_i. The width of receptive field is σ_V.

The input representing gaze direction h_i^G is a linear function of gaze angle y with slope m_i and positive offset b_i (baseline). Positive slope m_i will intuitively correspond to the gain fields increasing to the right.

$$h_i^V(x) = \exp\left(-\frac{(x_i - x)^2}{2\sigma_V^2}\right) \qquad , \qquad h_i^G(y) = m_i y_i + b_i \tag{1}$$

Additional recurrent input to the unit is determined by synaptic connections between neurons i and j. The weight of the connection W_{ij} depends on the distance between the preferred locations x_i, x_j and is given by a difference of two gaussians:

$$W_{ij} = A_E \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_E^2}\right) - A_I \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_I^2}\right)$$

where $A_E > A_I$ and $\sigma_I > \sigma_E$. The firing rate r_i of neuron i is given by a linear activation function s that is positive only when the sum of external and recurrent input exceeds the threshold h_{th}.

$$r_i = s(h_i + \sum_j W_{ij} r_j - h_{th}) \tag{2}$$

The model was evaluated on noisy inputs where random noise was added to each cell. As a result, model parietal neurons exhibit multiplicative gain fields and the network also effectively suppressed the input noise. This property is common for recurrent architectures. Figure 18A depicts the responses of one model neuron for six different gaze directions and the same visual stimuli. The external inputs to the cell are shown below (Fig. 18B). The linear gaze direction acts simply as an additive constant to the visual signal. The resulting response curves are almost exactly scaled versions of each other so they have clearly multiplicative character.

To evaluate the effect of recurrent connections, two feed-forward models were included for comparison. The first model (18C) corresponds to turning off the recurrent connections in the original model. The neuron responses are affected by gaze direction but not in the form of a product. Because the threshold h_{th} was set to one, only the inputs above this value were effective and the responses seem to only scale the input. The second feed-forward model (18D) had sigmoidal activation functions:

$$r_i = \frac{r_{max}}{1 + \exp(c(h_{th} - h_i))} \tag{3}$$

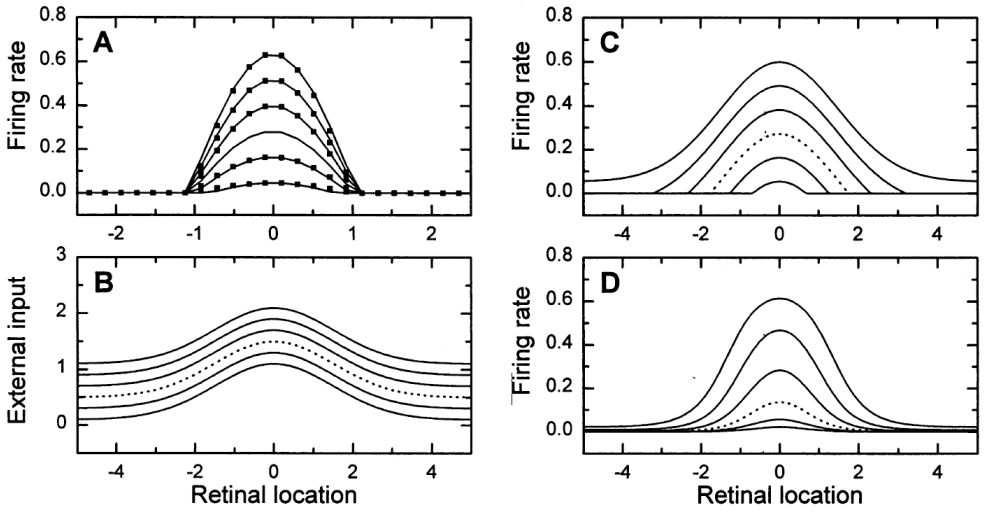


Fig. 18: Multiplicative gain fields. Adapted from Salinas and Abbott (1996).
A. Responses of gain-modulated parietal neuron in the network with recurrent connections. The horizontal axis indicates the location of visual stimulus and curves correspond to the different gaze directions. The curve without squares represents gazing straight ahead. Modulation by gaze-angle has evidently multiplicative character.
B. Plot shows the external input h_i for presented visual stimuli and different gaze angles. Note that the scale of horizontal axis is different from A.
C. Responses of a unit in simple feed-forward network equivalent to turning off the recurrent connections in the original model.
D. Responses of a unit in feed-forward network with sigmoidal activation functions (3). The response is approximately multiplicative but deviates in lateral gaze directions.

where c is a constant. In this case, the tuning curves were approximately multiplicative when gaze angle was close to zero, but exhibit deviations from a truly multiplicative response for lateral gaze directions.

The comparison with feed-forward models leads to a conclusion that recurrent connections are critical for generating multiplicative gain fields. Note that, as we discuss elsewhere in this thesis, it is not really necessary for gain fields to be multiplicative as their core property is non-linearity. However, the recurrent connections are ubiquitous in human brain and the recurrent model exhibits also some other notable properties, for instance its responses are very robust to the input noise. Also when presenting two visual stimuli at the same time, the model is sensitive only to the stronger one (Figure 19). This mechanism might be possibly used for the selection of targets in visual scene, because visual areas encode information about many objects and every non-targets may need to be filtered out.

Other interesting behaviour of recurrent network appears when the baseline b (eq. 1) is greater than the input threshold h_{th} (eq. 2). In this case, the activity

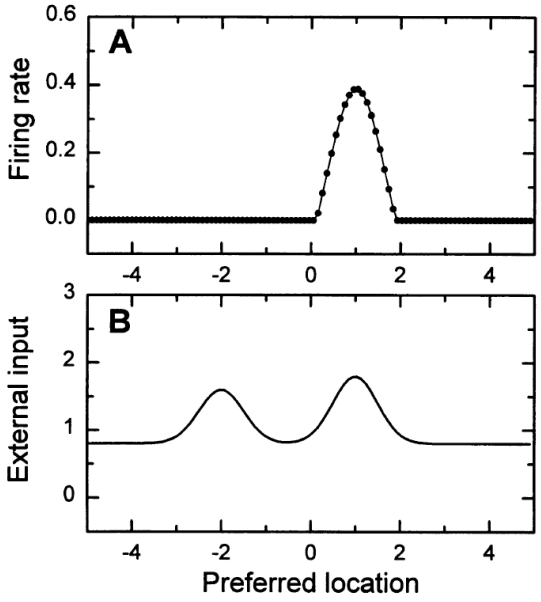


Figure 19: Two visual stimuli presented simultaneously. (Salinas and Abbott, 1996)
A. Only one peak of activity is generated by the population of network neurons when presented two visual stimuli simultaneously. Each dot corresponds to one neuron in the population, cells are arranged according to their preferred retinal location. The peak is matching the location of the strongest stimuli.
B. The input evoking the activity in A.

of network at the location of visual stimulus persists even after the visual input was removed, and also stays fixed in the presence of other inputs at different locations. The network can be reset by reducing the baseline below the input threshold. Such mechanism might act as a short-term memory buffer for target location during the reaching task. (Salinas and Abbott, 1996)

To this point we have investigated only the relation between recurrent connections and multiplicative gain fields without considering the actual output of the network. As described in section 2.2.2, the basis-function units also generate non-linear gain-fields and the network model is able to reproduce large number of effects found in neglect patients. Pouget et al. (2002) brought these two concepts together in an attempt to discover more about the neural basis of multisensory spatial representations from a computational perspective. We describe their findings in the following chapter.

2.4.2 Basis functions network with attractor dynamics

In this chapter we focus on a model created by Pouget et al. (2002). They stated two main aspect of multisensory integration and created a theory that has some interesting implications for our understanding of neural basis of sensorimotor transformations and the notion of frame of reference itself.

The first mentioned aspect is called the recoding problem and refers to the fact, that sensory modalities do not use the same representations and must be recoded into a common format before they can be combined. The recoding problem

is in the context of spatial representations reduced to a change of coordinates. The second factor is the reliability of sensory modalities because their reliability changes with the context and contains noise. Therefore, there must be a mechanism that allows preferring more reliable cues by means of statistical inference or probability approach. The presented model aims to address both aspects simultaneously. It is based on the basis-function networks, which provide plausible solution for spatial transformations, and recurrent connectivity, which brings optimal statistical properties. Combination of these two ideas leads to an architecture with an intermediate layer that contains gain-modulated neurons with partially shifting receptive fields.

Model architecture

The architecture and dynamics of the proposed model is illustrated in Figure 20. The network contains three external layers that encode eye-centered location of the object, head-centered position of the eye and head-centered position of the object. Basis function layer contains units that combine the activities of input units and intermediate the coordinate transformation. In addition to the model that we informally described in section 2.2.2 (Figure. 6b), there are also recurrent connections from the basis layer to the input layers and from the output layer to the basis function layer. As the activity can flow in any direction, coordinates can be transformed into any layer using the information from the other two layers. Therefore all layers are equal in the sense of input-output operations.

In order to compute accurate coordinate transformations, the connections in the network must meet some requirements. Consider the situation when we want to calculate the object position in head-centered frame of reference (x_a) from its eye-centered position (x_r) and eye position (x_e). This could be written in an equation as $x_a = x_r + x_e$. But because all layers encode the information in population codes, we need to apply this relation on single units. The tuning curves for individual units in all layers are bell-shaped, so the pattern of population activity will also have a bell-shaped profile. The position x will therefore be represented as the hill of activity located at x. Let us denote x_a^k as the preferred head-centered object location of unit k in the output layer. Similarly, x_r^i and x_e^j denote preferred eye-centered object location and preferred eye-position for unit ij in the basis function layer. To ensure that unit k will compute desired mapping, it simply needs to receive connections from all the basis functions units ij, such that $x_a^k = x_r^i + x_e^j$. The weights between layers and the activation functions of the basis function units were chosen so as it is guaranteed that when the network

is initialised with two hills in any pair of input layers, it eventually stabilises onto three hills that peak at locations x_a, x_r and x_e, linked through the relation $x_a = x_r + x_e$. Authors refer to this model as a *basis function network with attractor dynamics*, because in the context of dynamical systems stable network states are called attractors. Note that this architecture can be also seen as a special form of radial basis networks (RBF) (Pouget et al., 2002).

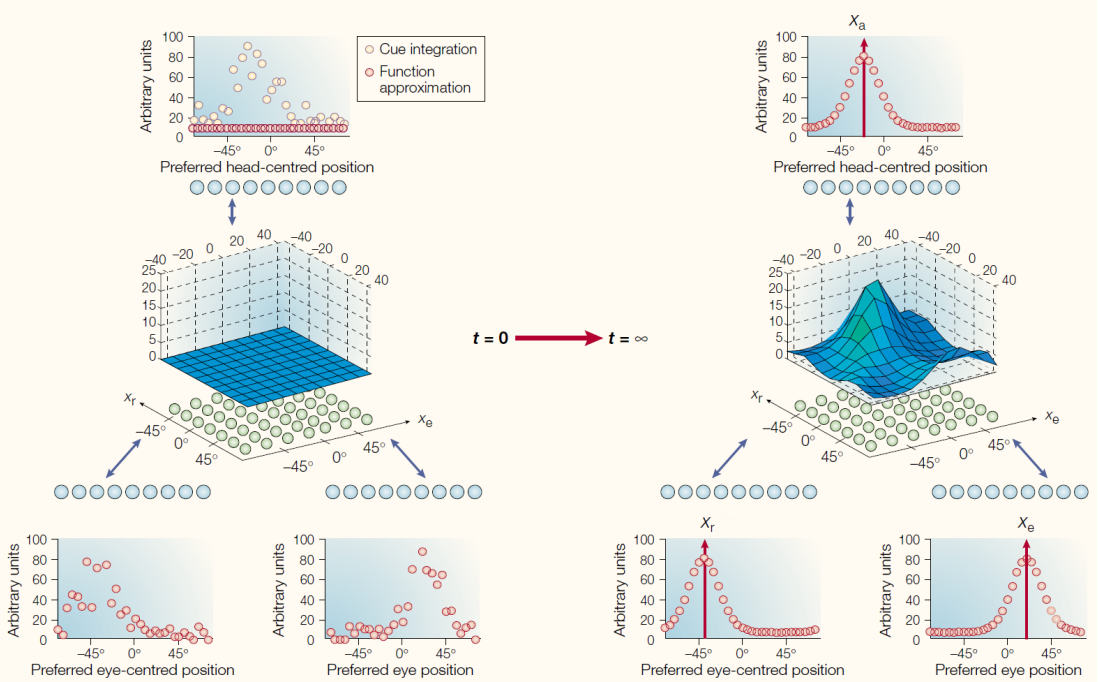


Fig. 20: A recurrent basis function network with attractor dynamics. The network is similar to the one shown in Fig. 6, but all connections are bidirectional. The network is initialised with noisy inputs (left), it settles into the stable state (right) where the positions of hills of activity are related through the function $x_a = x_r + x_e$ (see text) (Pouget et al., 2002)

.

Results and conclusions

The basis function network can perform spatial transformations from eye to head-centered frame of reference. The addition of recurrent connections enables the network to translate coordinates also in the opposite direction. Moreover, the recurrent network works as a maximum-likelihood estimator, meaning that when the network is initialised with noisy hills of activity and iterated, it stabilises to three smooth hills that represent the most likely position of the object. So the same network architecture can deal with the recoding problem (coordinate transformation) and statistical issues simultaneously. Units in basis function layer

have receptive fields that are gain modulated and partially shifting (Fig. 21). It indicates that they use both eye and head-centered frames of reference.

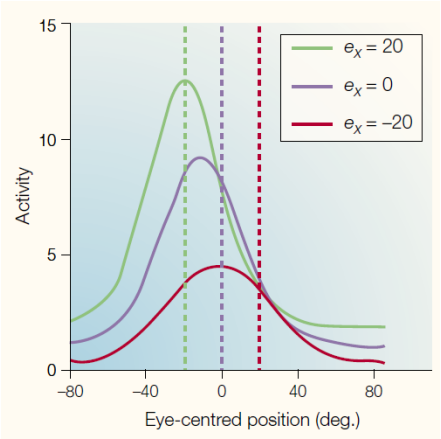


Figure 21: A partially shifting receptive field of typical basis function unit in the recurrent network. Three curves corresponds to three eye-positions (e_x). The shift of the receptive field is only half of the shift predicted for a head-centered receptive field (vertical lines), indicating that the receptive field cannot be assigned a single frame of reference (eye nor head-centered). (Pouget et al., 2002)

According to the authors (Pouget et al., 2002), basis function networks with attractor dynamics bring a new perspective on the multimodal spatial representations for reaching. Several studies have suggested that reaching motor commands are specified in eye-centered coordinates, regardless of the modalities in which the reaching target is defined. This may be just a consequence of the dominant role of vision in human behaviour, but it may also be the result of network architecture that can perform multiple tasks at once. The model of such network is shown in Figure 22. The motor plan can be computed by any combination of visual and posture modalities. As a result, the reaching command is encoded simultaneously in several frames of references, what could explain why reaching for an auditory target may be encoded in eye-centered reference frame even though such representation may seem unnecessary. Owing to the recurrent connections, the network is also able to make predictions of the sensory consequences of a motor plan. As the basis function maps integrate eye, head and body-centered coordinates, the suggested question is whether the notion of frame of reference is the best way to characterize these neural representations.

2.5 Robotic simulations

Although the use of artificial neural networks in robotics is very common, at the time there are not many works that would concern sensorimotor transformations from the perspective of our thesis and use either real or simulated robots. However, we did find some preliminary experiments in this field, so in this chapter we shortly review one of them.

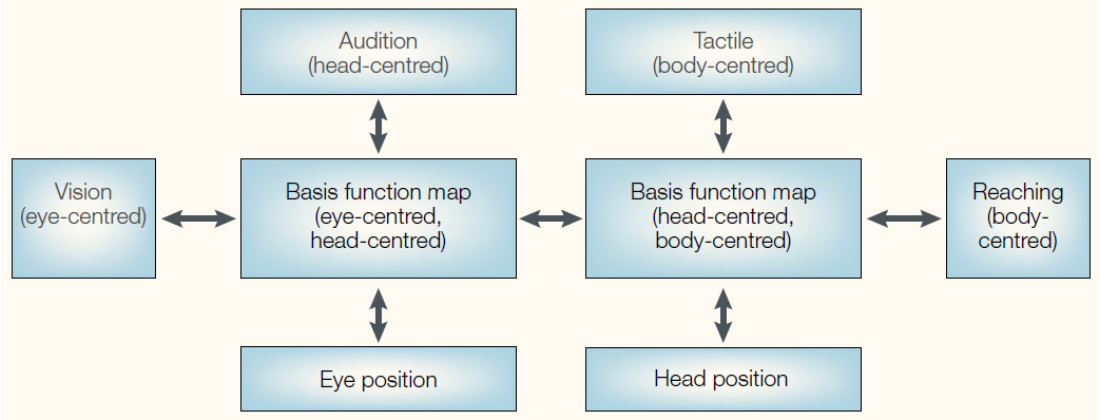


Fig. 22: A basis function network for reaching towards visual, auditory and tactile targets. The recurrent connections allows network to perform sensorimotor transformations from one sensory modality to another and predictions of the sensory consequences of motor plans. For comparison see former model in Figure 1C (Pouget et al., 2002).

The task of visuomotor arm control is in robotics often handled by self-organizing maps. Chinellato et al. (2011) designed a model based on basis functions network similar to the one we described in section 2.4.2 and implemented it in a simulated environment. These two concepts, SOM and basis functions network, were put together and explored in physical robot-head environment by Pitti and Blanchard (2012). The goal of their work was to model multimodal integration and spatial cognition in neonates. For this purpose, they constructed a robot-head that consists of a box with one camera and two bionic ears (Fig. 23A). The head and camera can rotate in horizontal direction, so the robot can provide together four kind of inputs: camera image with resolution 40×30, converted audio signal, and eye and head motor signals.

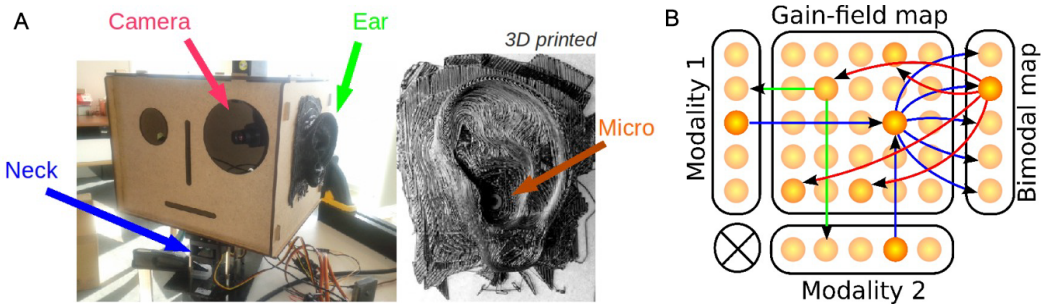


Fig. 23: A) Head-robot consisted of a box with one camera and two bionic ears. The robot has two degrees of freedom as it can rotate eye and head in horizontal direction. B) Architecture of the network model. Arrows illustrate the reentrant mechanism. The unimodal neurons fed univocal sensory signals to the gain-field neurons and to the downward neurons, and receive back the multimodal response. See text for more detail. (Pitti and Blanchard, 2012)

The network architecture was inspired by the model described in section 2.4.2 and is shown in Figure 23B. The gain-field neurons receive the activity from two neural populations by multiplying values of particular units with each other (two straight blue arrows). Then, the downward population can learn the neural activity from gain-fields neurons (blue arrows to bimodal map). Computed information is then used by reentry mechanism (red lines). In this stage, the triggered gain field neurons reinforce their links to downward neurons.

Weights of connections were adjusted in unsupervised manner by Rank-order coding algorithm (ROC). Briefly, ROC neurons are sensitive to the sequential order of incoming signals; that is, its *rank code*. The ordinal rank code can be obtained by sorting the signals vector by amplitude or temporal order. The activity of a neuron is then a function of the difference between input rank code and the rank code of neuron's weights. The updating rule is similar to the winner-takes-all learning algorithm used in SOMs. Since the synaptic weights of ROC neurons follow a power-scale density distribution, the ROC neurons are similar to basis functions.

The authors performed two experiments, one for encoding retinal coordinates into a head-centered reference frame using eye motor signal; and second for mapping auditory information to head and body reference frames. In the first experiment, there were 20 neurons coding eye motor signal and 50 neurons for retinal stimuli. The gain-field map therefore consisted of $20 \times 50 = 1000$ units. The downward part had 150 units. Unsupervised learning was done online in winner-takes-all style, i.e. only the weights of most salient neurons were updated. Over the time, the neural net self-organized itself to map retina and eye motor signals. The second experiment was performed in similar manner, but with auditory inputs and head motor signal. Here, the auditory neurons self-organized into two distinct receptive fields for left and right side. (We omit specific details because they actually were not provided by the cited paper.)

Both experiments revealed the presence of gain modulation in the downstream neurons (Fig. 24). In the first case, the response of the downstream neuron to the same visual stimuli was gain-modulated by eye motor signal (A); and in the second experiment, the response to the auditory signal was modulated by head motor signal (B).

Considering the context of our thesis, we see two interesting conclusions from this work. First, it illustrates an approach to modelling multimodal integration with realistic data. The direct consequence of this approach can be seen in clearly different profiles of gain fields generated in two different experiments (Fig. 24).

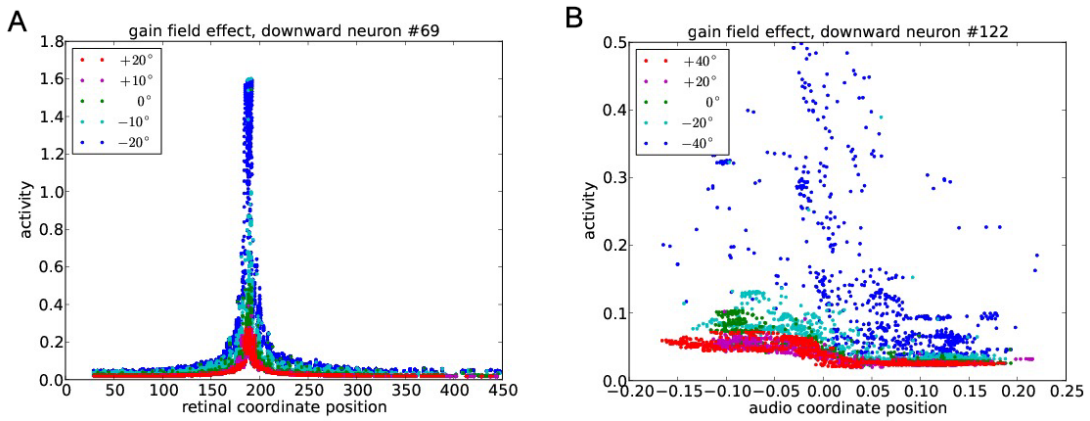


Fig. 24: Gain modulation of downstream neurons.
A) response to the same visual stimuli is modulated by eye motor signal,
B) response to the auditory signal is modulated by head motor signal
(Pitti and Blanchard, 2012)

We can compare this observation with the results from section 2.3.1, where we described simple feed-forward models trained to perform similar transformations. In both cases we observed the effect of gain modulation, but in the former experiment we did not observe any remarkable distinction between the gain field profiles.

The second conclusion is related to the training algorithm. We have already mentioned that the fact that a neural network trained by back-propagation develops gain fields does not mean that brain actually uses some variation of back-propagation. The same holds for basis-function approach. Now we have seen another modification of training algorithm that led to the population of neurons that coded spatial information in gain-modulated fashion. This suggests an idea, that gain modulation is a general mechanism used by every population of neurons that performs some multimodal integration, independent of the algorithm that was used to train the population.

3 Methods

3.1 Robotic Simulator iCub

The iCub is an open-source humanoid robot platform that was developed under collaboration of several European labs for the research in embodied cognition, cognitive development and advancing the understanding of natural and artificial cognitive systems (Metta et al., 2008). iCub was designed completely from the scratch, it has rich perceptuo-motor capabilities with 53 degrees of freedom and a cognitive capacity for learning and development. The software architecture encourages reuse and easy integration (Metta et al., 2010).

The robot is 104 cm tall and has the size of a three and half year old child. Nowadays, there are twenty physical iCubs in the world (see photo on Figure 25). The cost of iCub starts at €200,000. Therefore, many users and developers have to use the simulator instead.

The iCub simulator was designed to reproduce the physics and dynamics of the robot and its environment. Simulated robot is composed of rigid bodies connected via joint structures that correspond to the real robot design specifications, which means that simulated robot has the same height, mass and degrees of freedom as physical robot. The simulator uses ODE (Open Dynamic Engine) for simulating rigid bodies and collision detection. The Open Graphics Library (OpenGL) is used for rendering graphics. The robot may interact with objects that can be dynamically created and modified. Figure 26 demonstrates the simulator architecture (Tikhanoff et al., 2008).

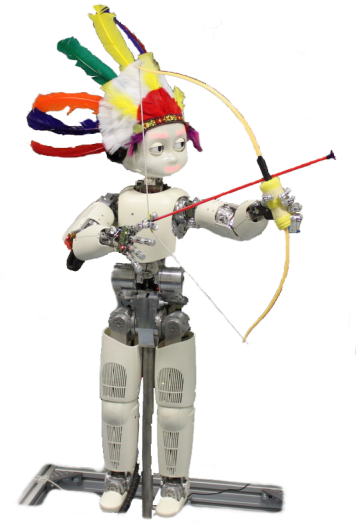


Figure 25: iCub - the humanoid robot.
The image shows experimental setup for the archery task. iCub learns to shoot arrows and hit the target center. Kormushev et al. (2010)

The iCub software infrastructure and inter-process communication is based

on top of YARP (Yet Another Robot Platform). YARP is an open source set of libraries, protocols and tools designed for dealing with common difficulties in robotics, such as interfacing with diverse and changing hardware or keeping modules and devices cleanly decoupled. It is OS neutral and written almost entirely in C++ (Fitzpatrick et al., 2013). The iCub simulator has the same interface as actual robot, so they can be used interchangeable from a user perspective.

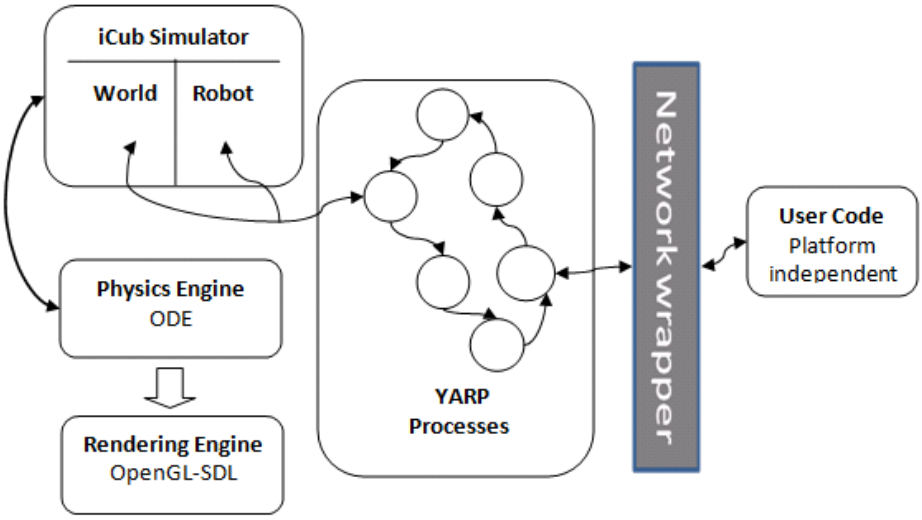


Fig. 26: iCub simulator architecture. User sends commands to the robot (controlling motors, sensors, cameras) and the world (managing objects and properties). Network wrapper exports YARP interface and allows device remotization (Tikhanoff et al., 2008)

.

3.2 Artificial neural networks

The idea behind artificial neural network comes from the mechanisms found in our brain. It is well known that the human brain is composed of $\sim 10^{11}$ highly interconnected cells called neurons, whose principal function is the collection, processing and dissemination of electrical signals (Haykin, 1998). The communication between neurons over synapses forms the basis of all brain functions (Society for Neuroscience, 2012).

The artificial neural network is a computational model that reflects these neuroscientific findings. Networks are composed of units connected by directed links of some weights, that determine the strength of the connections. The activation of each unit then depends on the weighted sum of signals from incoming connections (net) and is calculated by some activation function f, which is usually the

sigmoid $f(net) = 1/(1 + e^{-net})$ or a similar differentiable function. The so called feed-forward networks, units are arranged into layers and the activations spread from the first (input) layer through all neurons in the inner (hidden) layers to the final (output) layer. The schema of a simple 3-layer network is shown on Figure 27.

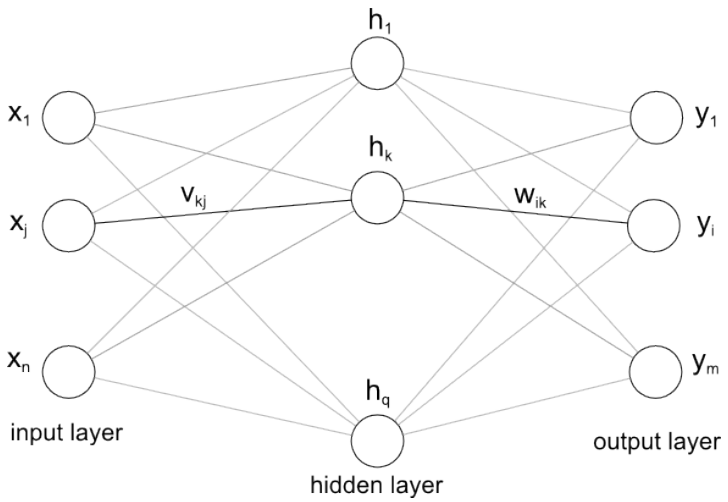


Fig. 27: Schema of a simple feed-forward neural network with three layers. v_{kj} is the weight of the connection from j-th neuron in the input layer to the k-th neuron in the hidden layer. w_{ik} expresses the weight from the hidden neuron h_k to the output neuron y_i

Mathematically, we can express the activation propagation as follows:

- The activations of neurons in the input layer are given
- Activations of hidden neurons: $h_k = f(\sum_{j=1}^{n+1} v_{kj} x_j)$
- Activations of output neurons: $y_i = f(\sum_{k=1}^{q+1} w_{ik} h_k)$
- Bias input: $x_{n+1} = h_{q+1} = -1$. The bias input sets the threshold for the unit, in the sense that the unit is activated only when the weighted sum of real inputs exceeds the bias value.

Feed-forward neural networks without hidden layers are called perceptrons or single layer networks. Considering the single output unit in perceptron, we can see that its activation function has the form $f(\mathbf{Wx})$, where $\mathbf{W}$ is the weight matrix and $\mathbf{x}$ is the input vector. The equation $\mathbf{W} \cdot \mathbf{x} = 0$ defines a hyperplane in the input space. It means that single layer networks can represent only linearly separable functions. The addition of hidden layers enlarges the space of hypotheses that the network can represent. The network with sufficiently large hidden layer is able to represent any continuous function and with two hidden layers, even discontinuous functions can be represented (Haykin, 1998).

Neural networks are commonly used for classification or regression, but they have a wide variety of other applications as well (e.g. image compression, stock market prediction or medicine applications). They have many interesting and powerful properties, from which the most significant is their ability to learn. Technically, learning of ANN is the process of changing the network parameters so as the network gives desired outputs for specific inputs.

3.2.1 Back-propagation learning algorithm

Back-propagation (BP) is the dominating training algorithm for feed-forward neural networks. The network is trained on a prepared dataset that should be big enough to sufficiently represent the problem domain. It is a supervised learning method, which means that the network is given a feedback about its performance. The feedback is usually the value of the error that the network produces.

Given a specific input, the perceptron output is a function of the weights of connections. The error of i-th output unit is the difference between the real (y_i) and desired value (d_i) and the network error is usually measured as the sum of squared unit errors. The size of this error depends on the number of output neurons, so many times it is more useful to use the mean value of squared unit errors (MSE). The goal of learning is to minimize the network error, what can be formulated as the optimization problem in weight space and the gradient descent method can be used. The learning rule for continuous single layer perceptron is as follows: $w_{ij} \leftarrow w_{ij} + \alpha(d_i - y_i)f_i'x_j$, where index j denotes the index of input neuron, i index of output neuron and $0 < \alpha < 1$ determines the speed of learning that is described later.

Intuitively, the rule increases the weight of connection when the error $(d_i - y_i)$ is positive and decreases when the error is negative. In multilayer perceptrons, it was not clear how to compute the error on the hidden layers. It turned out that the error from the output layer can be back-propagated to the hidden layers. Every hidden unit contributes to the error of output units according to the strength of connections. Therefore, the error on the hidden unit is calculated as weighted sum of errors from the output or following hidden layer. Equations for adjusting the weights are:

- for hidden-output weight: $w_{ik} \leftarrow w_{ik} + \alpha\delta_i h_k$, where $\delta_i = (d_i - y_i)f_i'$
- for input-hidden weight: $v_{kj} \leftarrow v_{kj} + \alpha\delta_k x_j$, where $\delta_k = (\sum_i w_{ik}\delta_i)f_k'$

- α is the learning rate that influences the speed of convergence. Setting the right learning rate could be tricky, small value may require too many steps to reach the solution, a large value may lead to oscillations.

On-line and batch back-propagation

The training using BP can be done in incremental (on-line) or batch manner. Incremental training updates the weights after each training pattern has been presented to the ANN. The following outline summarizes the steps of incremental learning:

- choose input (pattern) from the training set and compute the output
- evaluate the error $Err = \frac{1}{2}\sum_i (d_i - y_i)^2$
- back-ward pass - calculate errors on hidden neurons
- update weights according to the equations for hidden and output layers
- repeat until some stopping criterion is met, for example $Err < threshold$

In batch training, the weights are adjusted after each epoch, which means after the entire training set has been presented to the network. The error used for weight modification is computed as the average of errors on every pattern.

When considering the basic back-propagation algorithm, the on-line training learns faster than batch training because it performs more steps per epoch and does not get stuck in a local optimum so easily. Also as the size of the training set gets larger, batch training must use a smaller learning rate in order for its learning to remain stable (Wilson and Martinez, 2003). However, the benefit of batch learning is a better global view of the training process, what can in the combination with adaptive parameter modification (like size of the weight change or learning rate) lead to more advanced and effective algorithms, from one of the most noticeable is RPROP (3.2.4).

Local minima, step-size and the moving target problem

Numerous modifications and enhancements of the original algorithm were proposed in order to eliminate some limitations such as convergence to a false minima or a slow learning progress. The gradient descent methods naturally steer towards local minima, so the challenge for the algorithm is to step over and reach a global minimum. The major problems that contribute to the slowness of BP are *step-size problem* and the *moving target problem* (Fahlman and Lebiere, 1990).

The *step-size* problem is the problem of determining the largest step that can be taken in order to make the fastest descent to the minima. If the step-size is too large, the ANN may skip over optimal solution. If the step-size is too small, the network will always reach a local minima, but in a very long time. To target this problem, number of schemes were proposed such as adding *momentum* (3.2.2) or *quickprop* algorithm (3.2.3).

The *moving target problem* is related to the fact, that each time a weight is altered during the learning, the output of ANN is also altered and so are the gradients for all of the other connections. This problem has two sides. One side is that all weight changes except the first one are made on the basis of gradients that have changed since they were calculated. The other part of the problem is the inability to cooperate between the weight updates, what leads to a situation where all weights try to solve the same problem, even though an optimal solution may require that some units would focus on different problems. Instead of a situation in which each unit moves quickly and directly to assume some useful role, we see a complex dance among all the units that takes a long time to settle down (Fahlman and Lebiere, 1990). In large networks with many weights, the combination of all the independent weight updates can cause the final output of the ANN to move in an undesired direction (Nissen, 2007). Because BP algorithms update weights independently, they all suffer from the moving target problem. The Cascade-Correlation algorithm therefore addresses this problem by allowing only some of the weights to change at any given time. The topology of the network is not fixed, but evolves during the training and creates multilayer structure. New units that are added to the network play the role of permanent feature detectors and their weights are frozen once added (Fahlman and Lebiere, 1990).

3.2.2 Momentum

The *momentum* term (Rumelhart et al., 1988) has been found to dramatically increase the rate of convergence. It uses the old weight change as a parameter for the computation of the new weight change and thus helps to avoid oscillations problems when the error surface has a very narrow minimum area. The equation for the change of weight with momentum is:

$$\Delta w_{ik}(t) = \alpha \ \delta_i(t) \ h_k(t) + \mu \ \Delta w_{ik}(t-1)$$

where μ is the momentum parameter. It was demonstrated that the momentum parameter in gradient descent algorithm is equivalent to the mass of Newtonian particles that move through a viscous medium under a conservative force field. In the continuous case, the system is guaranteed to converge and the momentum parameter can improve the speed of convergence for most eigen components in the system. For the discrete time case, the momentum term provides the additional benefit of nearly doubling the parameter range over which the system converges. When the momentum parameter μ is close to one, the learning rate α can be nearly doubled (Qian, 1999).

3.2.3 Quickprop algorithm

The Quickprop algorithm (Fahlman, 1988) computes the gradient just as in standard BP, but it uses second-order method, based on Newton's method, to estimate the size of step and jump directly into the minimum of parabola. Quickprop relies on two assumptions: first, that the error function with respect to each weight is locally quadratic; second, that the small changes in one weight have relatively little effect on the error gradient observed at other weights.

To determine the parabola, the algorithm keeps for each weight current and previous slope and last weight change. The rule for computation of the weight change has form:

$$\Delta w(t) = \frac{S(t)}{S(t-1) - S(t)} \, \Delta w(t-1)$$

where $S(t)$ and $S(t-1)$ are the current and previous slopes ($= \partial Err/\partial w$). This value is only an approximation to the optimum value for the weight, but when applied iteratively it was reported to be extremely effective. Considering the current and previous slope, 3 cases can happen:

- $S(t) < S(t-1)$ and both slopes have the same direction, the weight is changed in the same direction towards the minima
- if the slopes are in the opposite direction, the optimum was skipped and the new value will be between previous and current position
- both slopes are in the same direction, but the current slope is the same size or larger in magnitude - this case can lead to taking the infinite step or moving backwards. This situation can be solved by parameter that constrains the maximum size of the step (maximum growth factor).

In many applications quickprop was reported to perform significantly faster than simple on-line BP, however, in our experiment it was the slowest from tested algorithms.

3.2.4 RPROP algorithm

RPROP is a batch learning algorithm for feed-forward networks that was proposed by Riedmiller and Braun (1993). The name stands for resilient backpropagation. The key concept is to perform a local adaptation of the weight update according to the behaviour of the sequence of signs of partial derivatives in each dimension of the weight space. In comparison with other gradient descent techniques the learning takes considerably less steps (Riedmiller and Braun, 1993).

Algorithm 3.1 RPROP — Riedmiller (1994)

$\forall i,j : \Delta_{ij}(t) = \Delta_0$
$\forall i,j : \frac{\partial E}{\partial w_{ij}}(t-1) = 0$
repeat
  Compute gradient $\partial_w E(t)$
  for $\forall$ weights and biases **do**
    if $(\partial_{w_{ij}} E(t-1) * \partial_{w_{ij}} E(t) > 0)$ **then**
      $\Delta_{ij}(t) = \mathbf{minimum}(\Delta_{ij}(t-1) * \eta^+, \Delta_{max})$
      $\Delta w_{ij}(t) = -\mathbf{sign}(\partial_{w_{ij}} E(t)) * \Delta_{ij}(t)$
      $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
      $\partial_{w_{ij}} E(t-1) = \partial_{w_{ij}} E(t)$
    else if $(\partial_{w_{ij}} E(t-1) * \partial_{w_{ij}} E(t) < 0)$ **then**
      $\Delta_{ij}(t) = \mathbf{maximum}(\Delta_{ij}(t-1) * \eta^-, \Delta_{min})$
      $\partial_{w_{ij}} E(t-1) = 0$
    else if $(\partial_{w_{ij}} E(t-1) * \partial_{w_{ij}} E(t) = 0)$ **then**
      $\Delta w_{ij}(t) = -\mathbf{sign}(\partial_{w_{ij}} E(t)) * \Delta_{ij}(t)$
      $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
      $\partial_{w_{ij}} E(t-1) = \partial_{w_{ij}} E(t)$
    end if
  end for
until converged

The principle behind RPROP is to consider only the sign of partial derivative and use it to determine the direction of weight update. For each weight w_{ij} there

is an adaptive 'update value' Δ_{ij} that determines the size of weight change:

$$\Delta w_{ij}^{(t)} = \begin{cases} & \text{if} \\ -\Delta_{ij}^{(t)} & \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & \text{else} \end{cases} \tag{4}$$

The $\frac{\partial E}{\partial w_{ij}}^{(t)}$ denotes the summed gradient information over all patterns of the training set. The value of Δ_{ij} evolves during the learning process according to the following learning rule:

$$\Delta_{ij}^{(t)} = \begin{cases} & \text{if} \\ \eta^+ * \Delta_{ij}^{(t-1)} & \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)} & \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)} & \text{else} \\ & \text{where } 0 < \eta^- < 1 < \eta^+ \end{cases} \tag{5}$$

The adaptation rule (5) is split on the change of sign of corresponding partial derivative. Change of sign indicates that the last update was too big and the algorithm has jumped over a local minimum. Therefore the update-value Δ_{ij} is decreased by factor η^-. When the sign remains unchanged, the update-value is slightly increased by the factor η^+ to accelerate convergence in shallow regions. The good values for η^- and η^+ are 0.5 and 1.2 (Riedmiller, 1994).

The weight update rule (4) follows the simple logic: if the error is increasing (positive derivative), the weight is decreased, if the derivative is negative, the update-value is added. The update-values and the weights are changed after the epoch. Note that in implementations there are also upper and bottom limits for the weights so as they don't grow too big or small. The pseudocode of RPROP is shown in Algorithm 3.1 (Riedmiller, 1994).

3.3 Self-organizing maps

Self-organizing map (SOM) is a type of artificial neural network that can perform projections from high dimensional data into low-dimensional representations while preserving the most important topological and metric relationships of original data. The model is biologically inspired and was introduced by Kohonen

(1982). SOMs are often used for visualisation and clustering of high dimensional data.

The network architecture consists of two layers with full connectivity. In addition, cells in the output layer have lateral connections with synaptic weights that decrease as a function (h) of the distance between the units. The distance function has usually shape of mexican hat, but in simulations also many simplified versions are used.

Learning is done in unsupervised manner based on the rule known as *winner-takes-all*. The algorithm can be written in following steps (Kvasnička et al., 1997, chap. 7)

1. randomly choose an input $\mathbf{x}$
2. find winner i^* for $\mathbf{x}$: $i^* = \text{argmin}_i \parallel \mathbf{x} - \mathbf{w}_i \parallel$
3. update weights: $\mathbf{w}_i(t+1) = \mathbf{w}_i + \alpha(t) \cdot h(i^*, i) \cdot [\mathbf{x}(t) - \mathbf{w}_i(t)]$
4. update SOM parameters (neighborhood h, learning rate α)
5. repeat until stop criterion is met

The training process for 2D is illustrated in Figure 28.

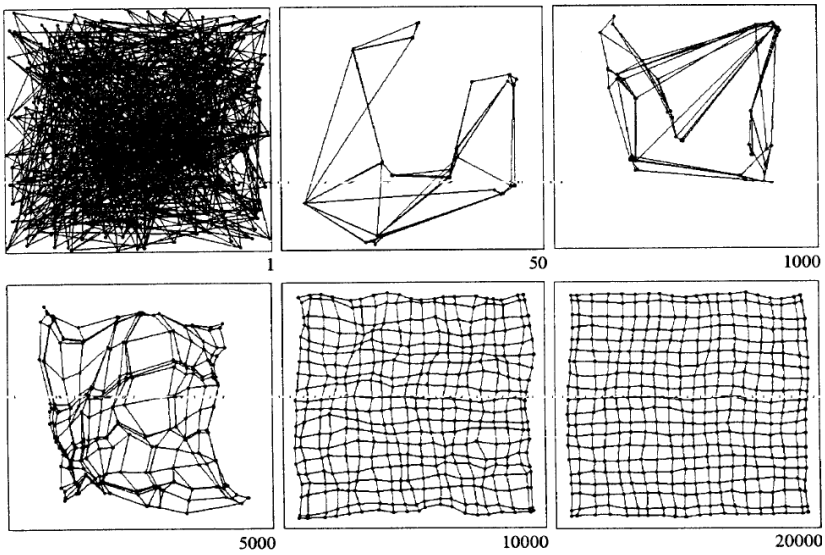


Figure 28: The process of training SOM. Network gradually learns the distribution of 2D data. Each image depicts the weight vectors at different iteration of the training process. (Kvasnička et al., 1997, chap. 7)

3.4 Population coding

Population coding is a strategy for encoding information by populations of cells, rather than by single neuron. It is a natural form of representing information in human brain and it is robust to noise and neuronal mortality. We cannot understand how our brain works without understanding coding mechanisms, because all other processes depend on them. Population codes are therefore very often

used also in computational models, either in purely mathematical or based on artificial neural networks.

One of the most common population codes is illustrated in Figure 29. The angular value is encoded into the population of four neurons with preferred directions uniformly distributed over the interval. The preferred direction is the location of neuron maximal activity – the peak of its tuning curve. The activity of i-th neuron is given by gaussian function of the difference between the preferred direction x_i and coded value x:

$$y_i(x) = \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right)$$

where σ is the width of the tuning curve. As we can see, the information can be decoded from the activities of first two neurons. The redundancy is always present and may lead to noisy representations, but also improves robustness. In a neural network, the output tuning curves should be wider than input tuning curves, because the information in the output layer cannot be greater than in the input layer and the wide tuning contains more information.

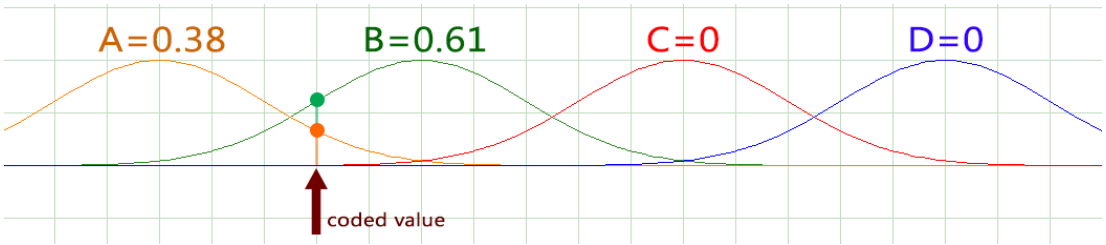


Fig. 29: Example of population coding. The value is encoded into the population of four neurons with gaussian tuning functions and uniformly distributed preferred directions. The activities of four neurons are: $\{0.38, 0.61, 0.00, 0.00\}$

4 Experiments

The goal of our experiment was to create a framework for studying spatial representations and transformations performed by neural networks. We found inspiration in the very first influential model proposed by Zipser and Andersen (1988). Their three-layer feed-forward network transformed retinal target into head-centered reference frame using information about eye-position (section 2.3). We decided to reproduce their experiment with more realistic representations of input stimuli and possibly including another modalities like head movements or eye vergence. According to the Dayan and Abbott (2005), it is a frequent mistake to assume that a more detailed model is necessarily superior. However, as we pointed out in section 2.5, using realistic data may lead to noticeable differences in observations. Also the work of Blohm et al. (2009), that we described in section 2.3.2, stressed the importance of accounting for full body geometry in order to sufficiently understand processes underlying eye-hand transformations. Therefore we chose as an experimental environment robotic simulator iCub. This approach has two main advantages. First, it implicitly accounts for full body geometry of 3.5 year old child and so can to some extent replace extensive mathematical model for 3D reach developed by Blohm and Crawford (2007). Secondly, the level of complexity seems to be perfectly appropriate for this kind of experiments.

Our experiment consisted of three steps. We first used iCub simulator to generate the training and testing data. Then we trained three-layer feed-forward artificial neural network to perform transformation from eye-centered to body-centered frame of reference using information about gaze direction. After the network was able to accurately perform this spatial transformation, we applied several visualisation methods to understand internal structures of the network. We will now describe these steps in more detail.

Note that we did not consider the model based on basis function network because of the curse of dimensionality. We would need more than $7 \cdot 10^5$ hidden units in order to perform the same transformation.

4.1 Generating the dataset in iCub simulator

Our artificial neural network was supposed to perform transformation from the eye-centered (retinal) target position to the body-centered position using the information about eye positions. To generate this kind of data in iCub simulator, we randomly moved iCub's eyes and put a random object in the space where

iCub can see it. Then we gathered three pieces of information: retinal images of the target (bitmaps from cameras), gaze direction, and the global target position. For illustration see Fig. 30.

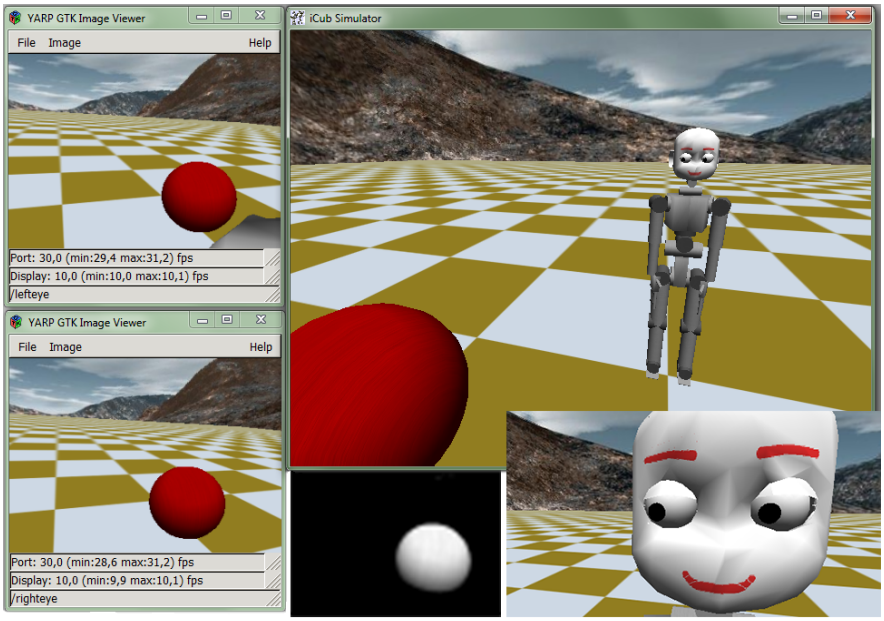


Fig. 30: Generating dataset in iCub simulator involves three steps: 1) randomly moving the eyes. 2) putting a random object in the scene. 3) capturing retinal images (left), gaze direction and global target position. Images from camera were downscaled and the background was removed (bottom, white ball on black background).

Controlling the iCub simulator

As mentioned in Methods (3.1), iCub simulator is almost entirely written in C++, so we also chose this programming language. Our programming environment was Microsoft Visual Studio 2010. Communication with iCub is based on top of YARP and consists of making remote procedure calls (RPC) to iCub simulator ports. These ports belong either to the iCub environment or to iCub itself. Therefore, putting an object in the scene is done through RPC call to the *world* port and controlling iCub by call to some of the iCub's ports. To get a better idea, we provide two small examples. Listing 1 shows how we can delete all objects in the simulator scene by making RPC call to the world port.

Listing 1: Deleting all objects in simulator scene

```
#include <yarp/os/Network.h>
#include <yarp/os/RpcClient.h>
using namespace yarp::os;
```

```
// create RPC client and connect it to the world port
RpcClient rpcWorld;
rpcWorld.open("/a/world");
Network::connect("/a/world", "/icubSim/world");

// prepare command for RPC
Bottle worldDelAll;
worldDelAll.addString("world");
worldDelAll.addString("del");
worldDelAll.addString("all");

//make RPC call
Bottle response;
rpcWorld.write(worldDelAll, response);
cout << "Deleting all objects: " << response.toString() << endl;
```

In Listing 2 we make iCub randomly turn its eyes. Every movement is in iCub specified by the position of some joint. Eyes movement, that is, setting horizontal and vertical angle of gaze direction, is implemented through two joints that belong to the iCub's head. One joint controls vertical direction (tilt) and one horizontal direction (version). Both joints have limits, for tilt the limits are $-35°$ (down), $15°$ (up) and for version from $-50°$ (left) to $50°$ (right).

Listing 2: Moving eyes to random position

```
//create driver for the head
Property options;
options.put("device", "remote_controlboard");
options.put("local",  "/a/head");
options.put("remote", "/icubSim/head");
PolyDriver robotHead(options);
if (!robotHead.isValid()) {
    printf("Cannot connect to robot head\n");
    return 1;
}

// get number of joints
IPositionControl *pos;
robotHead.view(pos);
int jnts = 0;
pos->getAxes(&jnts);  //number of joints

// prepare new positions for all joints
Vector positions;
positions.resize(jnts);
for (int j=0; j<jnts; j++) {
    positions[j] = 0;
}
positions[3] = fRand(-35.0, 15.0); //tilt (vertical)
positions[4] = fRand(-50.0, 50.0); //version (horizontal)

// move to desired positions
```

```
pos->positionMove(positions.data());
bool done = false;
while (!done) {
    pos->checkMotionDone(&done);
}
```

Generating the data

We will now describe the process of data generation. The first step, moving iCub's eyes, is quite straightforward. We generated random vertical (tilt) and horizontal (version) angles and moved eyes into the corresponding position. Both eyes were rotated by the same angles and in this stage we omitted eyes vergence. Once the eyes were moved, we needed to determine the space where the iCub is able to see the object, that is, to determine iCub's field of view. Cameras in iCub simulator have resolution 320×240 points and use simple pinhole projection with the focal length 257.34. This means that a box with dimension 320×240 cm in the distance 257.34 cm before iCub's head fills the whole field of view. We first put the object within these limits and then rotated its position by the same angles as the eyes. One additional check was needed to ensure that the rotation did not put the object under the ground. The simulator has three predefined objects that can be put into the scene: box, sphere and cylinder. We have generated datasets that contained all types of objects and also datasets that contained only spheres. To make the data diverse enough, we generated objects with random sizes. However, we needed to prevent close objects from being too big and far objects getting too small, while still preserving this characteristic. Therefore we chose random sizes for objects based on their distance before iCub. After the object was put in the scene, we captured images from both cameras and saved eye position and global target position. We repeated this procedure about 1500 times to generate a sufficient number of training and testing patterns.

Processing data for neural network

In order to use the dataset as an input for the neural network, we processed each pattern into the set of real numbers in the interval $\langle 0, 1 \rangle$. The global object position was translated into two angle values that determined the direction to the object from iCub's chest (Fig. 31). These two slopes and values of eyes tilt and version were converted into the population codes as described in section 3.4. We used 11 neurons for eye tilt, 21 neurons for eye version, 19 neurons for horizontal slope and 19 neurons for vertical slope. Note that these numbers were

found experimentally and produced best results. Camera images from the left and right eyes were flipped in both directions and downscaled to 64×48 pixels. For better performance, we also removed the background. The processed image is illustrated in Fig. 30 (white ball on black background). The value of each pixel was scaled to interval $\langle 0, 1 \rangle$. The image input was therefore represented by $2 \times 64 \times 48 = 6144$ neurons.

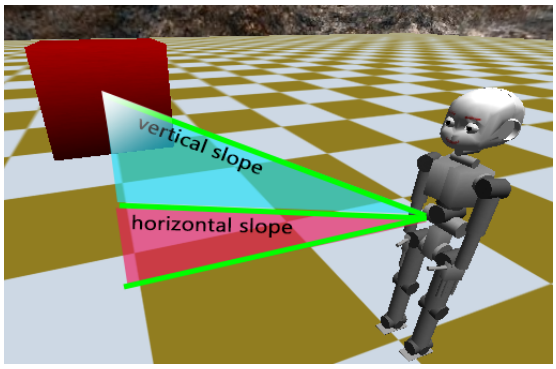


Figure 31: Object position represented by horizontal and vertical angles

4.2 Network architecture

We used three-layer feed-forward artificial neural network. The input layer consisted of four units: two units represented object retinal images from the left and right eye and two units were used for encoding horizontal and vertical eye positions. Retinal images with resolution 64×48 pixels were encoded row by row in population of 6144 neurons. Eye tilt was encoded by 11 neurons with preferred directions uniformly distributed over the interval $\langle -35°, 15° \rangle$ and eye version by 21 neurons distributed over the interval $\langle -50°, 50° \rangle$. Best results were achieved for tuning curves with width $\sigma = 5$ for tilt neurons and $\sigma = 7$ for eye-version neurons (see 3.4). The total number of input units was $6144 + 11 + 21 = 6176$.

The hidden layer consisted of 64 neurons. We experimented with several numbers of hidden neurons ranging from 50 to 300 without any significant change in network performance. The performance of the networks with less than 40 neurons was quite limited. The output layer contained two units of 19 neurons that encoded horizontal and vertical direction to the object from iCub's chest. The preferred directions of output neurons were in both cases uniformly distributed over the interval $\langle -90°, 90° \rangle$ and the width of tuning curves was $\sigma = 10$. The activation functions for neurons in hidden and output layers were sigmoids, with the gain[1] for hidden layer 0.05 and for the output layer 0.1. The activation function

[1] For clarification, this is not a gain field. In FANN, this parameter is called steepness.

thus had form:

$$f(net) = \frac{1}{e^{-steepness*net}}$$

The network was fully connected, meaning that there was full connectivity between the input and hidden layer and between the hidden and output layer. However, there were many more input units that encoded retinal image than units encoding eyes version and tilt (6144 vs. 32 neurons). We therefore artificially changed the weights of connections from input units to the hidden units in a way that we could change the ratio between the numbers of neurons encoding these two modalities. This can be expressed as follows:

$$net = r \cdot \sum_{i}^{N_r} w_i r_i + e \cdot \sum_{j}^{N_e} w_j e_j$$

where *net* is the input to a hidden neuron; r and e are the coefficients used for balancing retinal inputs r_i and eyes-positions inputs e_j; N_r and N_e are the numbers of units encoding given modality. We calculated r and e to correspond to the desired ratio $R : E$, where R is the desired size of retinal inputs contribution and E is the desired contribution of eyes-positions inputs. The equations for calculating e and r were:

$$r = \frac{R \cdot (N_e + N_r)}{N_r \cdot (R + E)} \quad , \quad e = \frac{E \cdot (N_e + N_r)}{N_e \cdot (R + E)}$$

The chosen ratio $R : E$ was $2 : 1$, but there was no significant difference in network performance for slightly different ratios. Even the original network was able to successfully perform the transformations. However, by setting this ratio we achieved faster training, better accuracy and weight profiles that were nicer for visualisation purposes.

4.3 Training and testing

We trained our neural network using Fast Artificial Neural Network Library (FANN). It is a free open source library for training multilayer feed-forward networks using several variations of backpropagation algorithm. FANN is written in C, but there are more than 15 bindings to other programming languages including C++. We have modified the library in order to be able to change the ratio between the contributions of input modalities, as we described in previous section.

The training dataset consisted of 1000 patterns. Training was performed by

various modifications of backpropagation algorithm and stopped when the MSE reached the threshold $5 \cdot 10^{-4}$. This value was found experimentally, we did not observe any significant improvement in the network accuracy for smaller values. In the first trials, the best training performance was achieved by RPROP algorithm. It performed approximately 8 times faster than standard BP and about 10 times faster than quickprop algorithm. Because quickprop appeared to be very ineffective, we didn't use it in later trials. Interestingly, adding momentum to standard BP dramatically increased the speed of training. We experimentally found values of the learning rate and the momentum term, with which the incremental BP outperformed RPROP. These values are $\alpha = 1.5$, $\mu = 0.9$ (see 3.2.2). The additional disadvantage of RPROP and quickprop algorithms was that they often generated weights that reached either top or bottom limits and thus were not suitable for visualisation purposes. This is an interesting point, because it indicates that the right choice of the training algorithm may be very important for the purposes of studying the internal structures of the network.

The accuracy of the trained network was tested on another 500 patterns and varied according to the complexity of the dataset. For the dataset that contained various types of objects (box, sphere, cylinder) at various sizes, the mean errors in horizontal and vertical directions were around 4° and the standard deviation of error was 3.5°. In the dataset that contained only spheres at various sizes, the mean and standard deviation of error was around 3°. In the dataset with spheres at fixed size, the mean error and deviation was around 2°. We did not find any significant correlation between the size of error for particular pattern and position of eyes or object in this pattern. The typical distribution of accuracy errors over testing patterns is shown in Fig. 32.

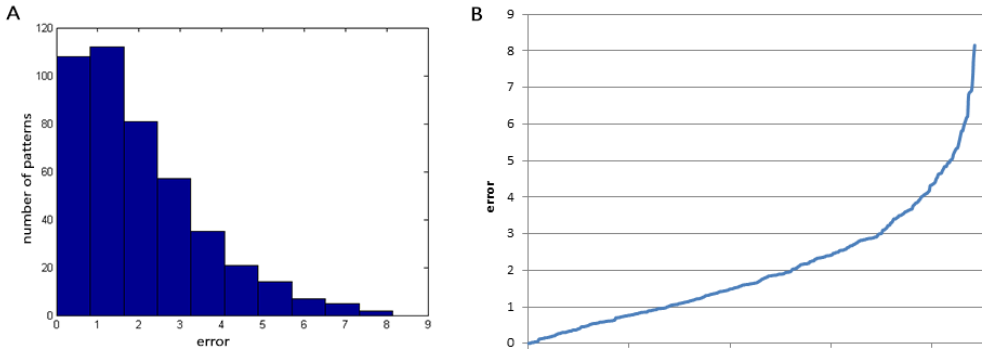


Fig. 32: Distribution of errors over testing patterns. A) Histogram of errors. B) Errors sorted by size.

4.4 Results

4.4.1 Receptive fields

After the network learned to accurately perform the transformation from eye-centered to body-centered coordinates, we examined the hidden layer for the effect of gain modulation and shifting receptive fields. For this purpose, we first visualised the receptive fields of hidden units. This visualisation was done by plotting the weights between the hidden unit and the neurons representing visual input. We found a wide variety of receptive fields, but we were able to divide them into three groups as illustrated in Figure 33. In the first group (A), we can distinguish continuous area with positive weights contrasting with an area of smaller or negative weights. Note that this is the receptive field of hidden neuron #4 in our network. The second group (B) has receptive field divided into two parts, usually with stronger weights on the sides. In the third group (C), we were not able to find any continuous area and the receptive field was hard to interpret without further investigation. Quantitatively, in the network with 64 hidden units we have found 41 units of type A, 15 of type B and 8 of type C. Note that these numbers are specific for the given network and would be slightly different if we repeated the training process. We may conclude that the majority of units have developed continuous receptive fields for particular area(s) in the visual space.

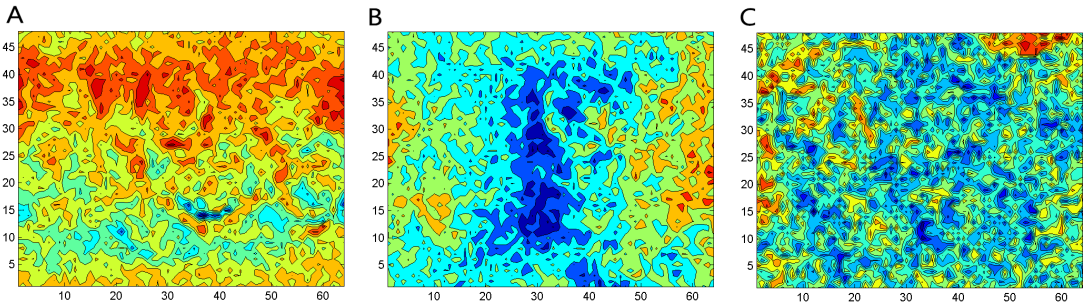


Fig. 33: Examples of receptive fields. A) Receptive field is continuous (neuron #4). B) Receptive field is divided into two parts. C) Uncertain receptive field.

4.4.2 Gain modulation

In the next step we examined the effect of gain modulation. In section 2.2 we explained that gain modulation is revealed when a modulatory input changes the response amplitude of a neuron to the other input, without modifying its selectivity. To observe this effect we recorded the responses of hidden units to

the fixed visual stimuli and different eye positions, which were changed in a systematic fashion with 10° step in vertical direction and 20° step in horizontal direction. Together there were 25 different eye positions arranged in a grid 5×5.

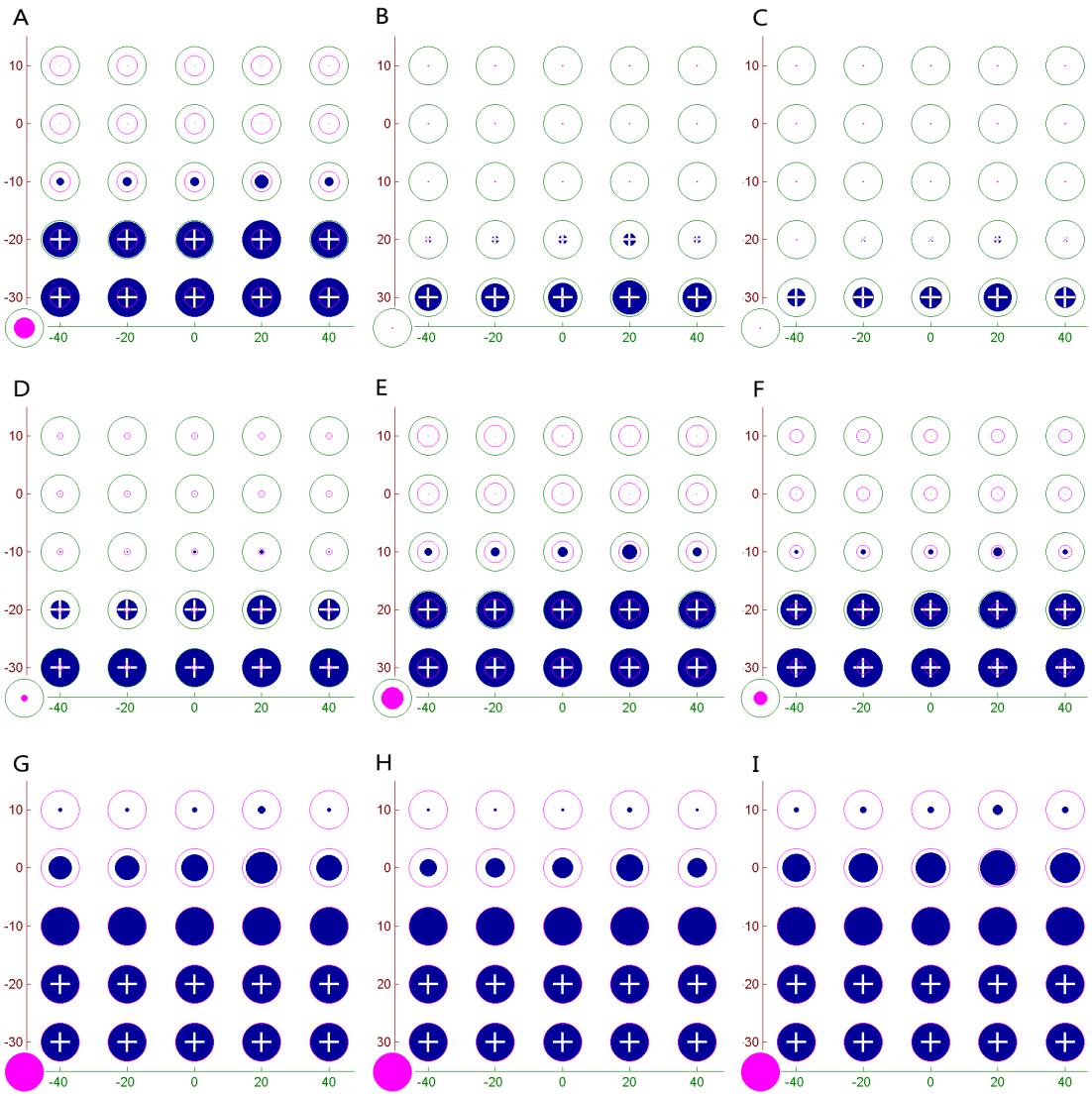


Fig. 34: Gain modulation of neuron's response (neuron #4). The arrangement of panels A-I indicates the locations of objects relative to iCub gazing straight ahead. In panel A, the object was on the left and above the iCub; in panel I, the object was located on the right side and close to the ground. (Note that objects in panels A-C were projected on the bottom part of retina, because the retinal image is flipped. Other panels likewise). The magenta circles represent how would unit response only to the corresponding visual stimuli without the influence of eye-position. In every panel, blue circles represent unit responses to the visual stimuli modulated by corresponding eye position. The top left circle denotes response when gazing up and left, the bottom right circle gazing down and right. The white plus sign indicates that the effect of modulation was excitatory.

We repeated this process with 9 different retinal images that depicted the object at particular locations. These locations were chosen systematically, starting with top left region and ending with bottom right region of the image. Thus, all together there were $9 \times 25 = 225$ different configurations of visual stimuli and eye positions. The response profile of one typical hidden unit (neuron #4) is illustrated in Fig. 34. Panels A-I represent the position of object relative to the iCub. In panel A, the object was located on the left and above iCub; in panel I, the object was on the right side and close to the ground. The filled magenta circles depict neuron's responses only to a visual stimulus without the information about eye position. In other words, all input units representing eye position had a value set to 0. We can observe that the location of largest magenta circles (panels G-I) reflect the unit's receptive field (compare Fig. 33A and 34). The largest magenta circles are on the bottom because the retinal image of object was flipped. The blue filled circles correspond to the different eye positions. The effect of gain modulation is evident in all panels. For instance, in panel D, blue bottom circles illustrate that the unit is active even though its response to purely visual input is weak. We may notice that the modulation has the same direction as the receptive field, meaning that the receptive field is sensitive to the object at the bottom and the effect of gain modulation is highest when gazing down. We hypothesise that this may be a desired behaviour.

Imagine an output neuron whose activity indicates that the object is located close to the ground. This neuron is fed by the population of hidden neurons. We may think about each hidden neuron as a small unit whose activity indicates some specific position of the object. The output neuron thus collects these indications and decides if it means that the object is located on the ground. We will now look at the hidden neuron #4. The receptive field suggest that the neuron may indicate objects located on the ground. Let's consider various combinations of visual stimuli and eye position. The object cannot be on the ground when iCub looks up and sees an object. In case when iCub gazes straight ahead, the object is on the ground only when its projection falls on the top part of the retina. In the last case, when iCub fixates its eyes down, the object is always on the ground. We can see that the response profile in Fig. 34 accurately corresponds with this assumption. Neuron #4 is thus active only when the object is on the ground. We would expect, that the output neuron indicating this position will have strong connection to this hidden neuron. This is actually what we found when we looked at the weights from neuron #4 to the output units (Fig. 35A). The preferred directions of the output neurons were uniformly distributed over

the interval $\langle -90°, 90° \rangle$ and iCub could move its eyes only $-35°$ down, therefore the strongest connections are on the left side close to the centre. In the light of the experiment from section 2.3.1 and Fig. 12A-D, we could say that the RF-GF direction difference was zero. We performed numerical analysis of this assumption and computed RF-GF direction difference for each hidden units. The histogram with results is shown in Fig. 35B. Because for the majority of hidden units is RF-GF difference close to zero, we have a reason to believe that our hypothesis may be one of the core principles of gain modulation.

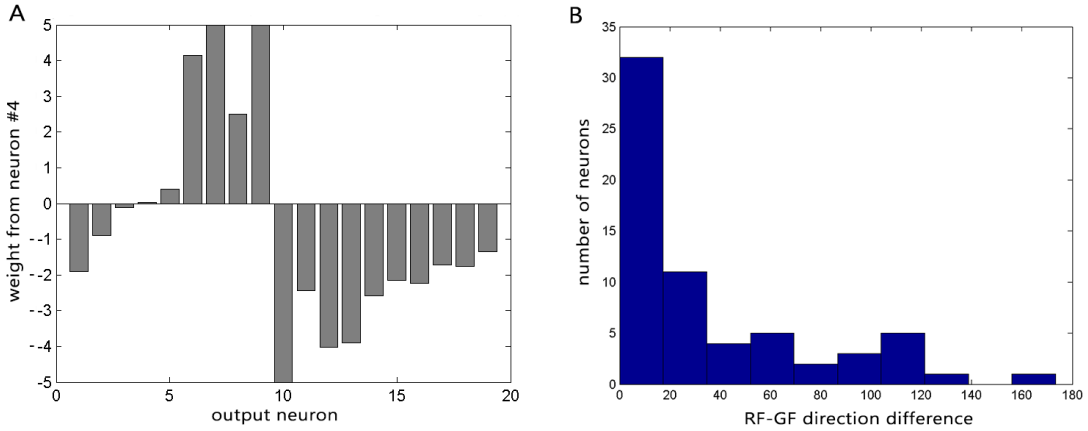


Fig. 35: A) Weights of connections between neuron #4 and output neurons encoding vertical object position. B) Histogram of RF-GF direction differences

The response profiles of hidden units were often much more complex than in the Fig. 34. To explore if there are any characteristic profiles or clusters of profiles, we used a visualisation method based on star plots. For illustration, visualisation of neuron #4 response profile is shown in Fig. 36.

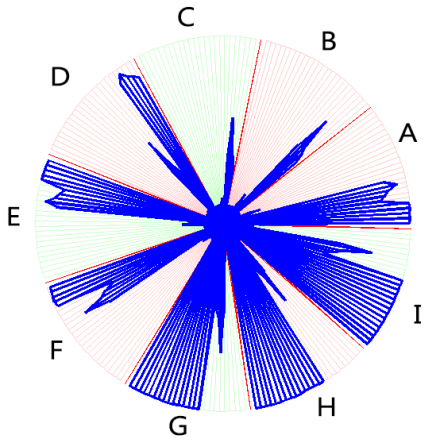


Figure 36: Star plot visualisation of response profile (neuron #4). Each neuron response is plotted on one axis. See Fig. 34 for comparison.

We then trained one-dimensional self-organizing map to organize profiles in topographic order. Organized profiles of all 64 hidden units are depicted in Fig. 37.

At the first glance the figure is hard to interpret, but we were able to conclude that the center of response profile is pulsing from the bottom part to the top through the stages when it pulses from the right to the left and vice versa. This progress itself is not very important, the point here is that the response profiles appear to be specialized in similar manner as the receptive fields thanks to the effect of gain modulation.

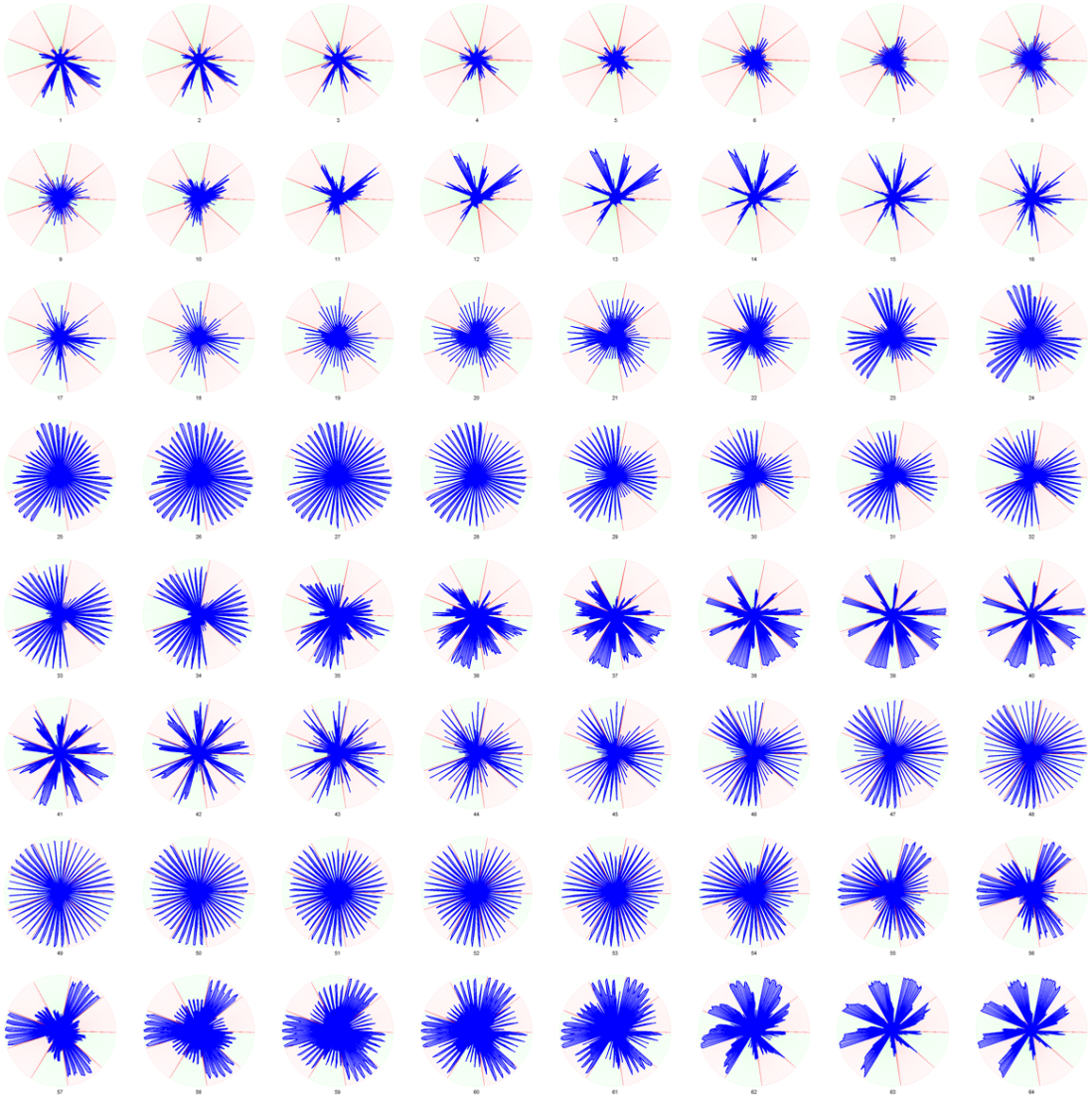


Fig. 37: Response profiles organized by SOM. Every circle represent response profile for one hidden unit.

4.4.3 Reference frames

Last but not least, we performed the reference frame analysis. We proceeded in similar manner as we stated elsewhere in this thesis, which means that we examined if the centre of mass of the receptive field shifts for different gaze directions. To determine the centre of RF, we reorganized units' response profiles in a way, that for every particular gaze direction we collected the information about units responses to 9 different visual stimuli. Instead of a grid containing 9 panels with 25 responses, we thus get grid containing 25 panels with 9 responses. Three such panels for neuron #4 are illustrated in Fig. 38. The centre of mass depicted by red dot clearly shifts for three different gaze directions. We computed the centres of mass for each of 25 different eye positions and visualised them in one plot. Fig. 39 illustrates all shifts of RF for three hidden units. Two first examples have shifting receptive fields, indicating that these units encode in body-centered coordinates. In the third case, the receptive field remains close to the centre, which is interpreted as using eye-centered coordinates. In order to determine the common reference frame used by the population of hidden neurons, we computed the absolute shifts and the standard deviations for all units and put them into histograms in Fig. 40. If the majority of units had absolute shifts close to zero, we would interpret it as encoding in eye-centered reference frame. The body-centered reference frame would be signalized by the numerous absolute shifts close to one. Since none of these situations happened, we concluded that the hidden layer encodes the object position in an intermediate coordinates between eye- and body-centered reference frames.

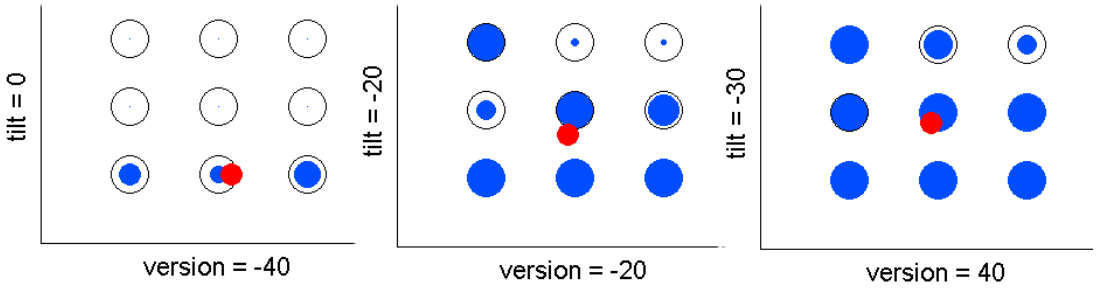


Fig. 38: Shifts of the receptive field of neuron #4. Each panel illustrates the responses for 9 different visual stimuli at one fixed gaze direction given by tilt and version value. The red dot indicates the centre of mass of the receptive field.

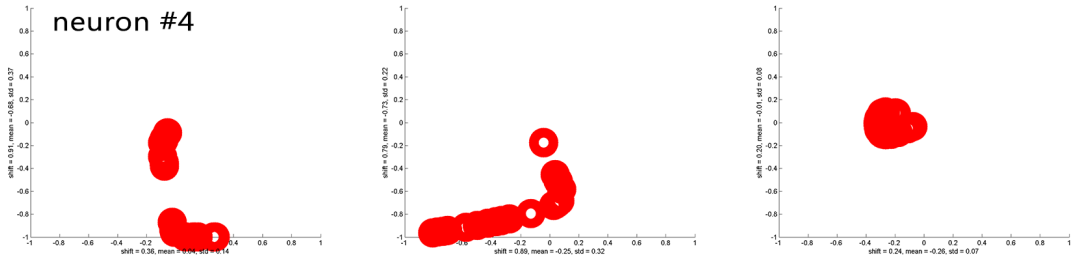


Fig. 39: Shifts of the centres of mass of RF for three hidden units. Left panel depicts all centres of mass of RF of neuron #4 for 25 different eye positions. The centres of mass are represented by red circles.

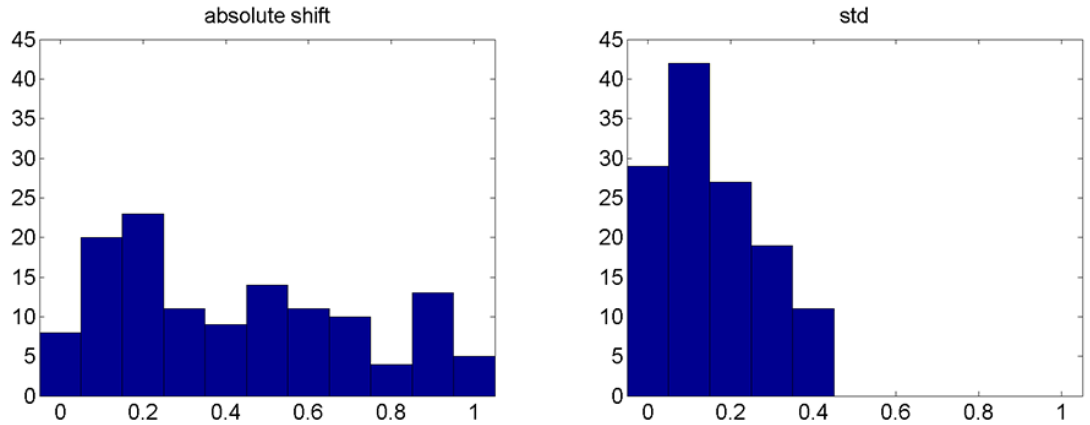


Fig. 40: Histogram of absolute RF shifts (left) and histogram of the standard deviations of RF shifts (right). The vertical axis denotes the number of units with the given value of the shift. Both horizontal and vertical shifts were used in the histograms, so the sum of bars in each histogram is $2 \times 64 = 128$.

5 Conclusions

Spatial transformations performed by neural networks are crucial components of cognitive processes. Central to the spatial representations in neural networks is the concept of a frame of reference. Our thesis explores the computational principles of spatial transformations between various frames of reference by examining the hidden structures of the neural network constructed for this purpose. Before we designed our own experiment, we studied past and recent network models able to perform multimodal integration. The first influential model of spatial transformations comes from Zipser and Andersen (1988). They reported the effect known as gain modulation, that was later closely studied and is now considered to be one of the core principles of multimodal integration in neural networks. More advanced studies of visuomotor transformations comes from Blohm et al. (2009), who trained a four-layer feed-forward network to compute a reach plan from the visual information about the target and hand position and posture signals about eye and hand positions. Their model accounts for the real geometry of the human body in 3D. Pitti and Blanchard (2012) recently constructed a robot-head to study a multimodal integration and spatial cognition in neonates. Their preliminary experiments with the basis function network revealed the presence of gain fields. In order to study spatial transformation in realistic environment, we have used the robotic simulator iCub. It served as a data generator for the three-layer feed-forward neural network, that we trained by several modifications of BP algorithm to transform eye- to body-centered coordinates using the information about the eyes position. The best results were achieved with the standard version of BP with momentum term. The network was able to successfully perform the transformation task with the accuracy within 2°. We examined the hidden layer of the network by the means of several visualisation techniques that revealed the effect of gain modulation and shifting receptive fields. We explained why we consider the effect of gain modulation to be perfectly suitable for the spatial transformations. The results of the reference frame analysis indicate that the hidden layer of the network encodes object position in the intermediate reference frame between eye- and body-centered coordinates. The advantage of using an iCub simulator lies in the fact that it accounts for full body geometry in 3D without the need for additional mathematical models. Our network works in 2.5D, meaning its output encodes only the direction to the observed object, not the distance. We believe that our work have prepared the ground for further studies of spatial transformations in neural systems, including the experiments in full 3D.

References

Andersen, R., L. Snyder, C. Li, and B. Stricanne (1993). Coordinate transformations in the representation of spatial information. *Current opinion in Neurobiology 3*(2), pp. 171–176.

Andersen, R. a. and V. B. Mountcastle (1983). The influence of the angle of gaze upon the excitability of the light-sensitive neurons of the posterior parietal cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience 3*(3), pp. 532–48.

Batista, A. (2002). Inner space: Reference frames. *Current biology 12*(11), pp. 380–383.

Blohm, G. and J. D. Crawford (2007). Computations for geometrically accurate visually guided reaching in 3-d space. *Journal of Vision 7*(5), pp. 4.1–22.

Blohm, G. and J. D. Crawford (2009). Fields of gain in the brain. *Neuron 64*(5), pp. 598–600.

Blohm, G., G. Keith, and J. Crawford (2009). Decoding the cortical transformations for visually guided reaching in 3D space. *Cerebral Cortex 19*(6), pp. 1372–1393.

Blohm, G., A. Khan, and J. Crawford (2008). *Spatial transformations for eye–hand coordination*, Volume 9, pp. 203–211. Elsevier Inc.

Buneo, C. and R. Andersen (2006). The posterior parietal cortex: sensorimotor interface for the planning and online control of visually guided movements. *Neuropsychologia 44*(13), pp. 2594–2606.

Chang, S., C. Papadimitriou, and L. Snyder (2009). Using a compound gain field to compute a reach plan. *Neuron 64*(5), pp. 744–755.

Chinellato, E., M. Antonelli, B. J. Grzyb, and A. P. D. Pobil (2011). Implicit sensorimotor mapping of the peripersonal space by gazing and reaching.

Dayan, P. and L. F. Abbott (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.

Deneve, S., J. Duhamel, and A. Pouget (2001). A new model of spatial representation in multimodal brain areas. *Advances in Neural Information Processing Systems 13* .

Duhamel, J. R., F. Bremmer, S. Ben Hamed, and W. Graf (1997). Spatial invariance of visual receptive fields in parietal cortex neurons. *Nature 389*(6653), pp. 845–8.

Fahlman, S. E. (1988). An empirical study of learning speed in backpropagation networks. Technical Report CMU-CS-88-162, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Fahlman, S. E. and C. Lebiere (1990). The cascade-correlation learning architecture. Technical Report CMU-CS-90-100, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Fitzpatrick, P., L. Natale, and G. Metta (2013). Yet another robot platform - a thin middleware for humanoid robots and more... http://eris.liralab.it/yarpdoc/what_is_yarp.html. online; accessed 31-January-2013.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.

Khan, A., L. Pisella, and G. Blohm (2012). Causal evidence for posterior parietal cortex involvement in visual-to-motor transformations of reach targets. *Cortex* , pp. 1–10.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*(1), pp. 59–69.

Kormushev, P., S. Calinon, R. Saegusa, and G. Metta (2010). Learning the skill of archery by a humanoid robot icub. In *Humanoids*, pp. 417–423. IEEE.

Kvasnička, V., L. Beňušková, J. Pospíchal, I. Farkaš, P. Tiňo, and A. Král' (1997). *Úvod do teórie neurónových sietí*. Iris.

Metta, G., L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano (2010). The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks 23*(8-9), pp. 1125–1134.

Metta, G., G. Sandini, D. Vernon, L. Natale, and F. Nori (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, New York, NY, USA, pp. 50–56. ACM.

Nissen, S. (2007). Large scale reinforcement learning using q-sarsa and cascading neural networks. Master's thesis, University of Copenhagen, Denmark.

Pitti, A. and A. Blanchard (2012). Gain-field modulation mechanism in multimodal networks for spatial perception. *2012 12th IEEE-RAS Int. Conf. on Humanoids Robots* , pp. 297–302.

Pouget, A., S. Deneve, and J.-r. Duhamel (2002). Opinion: A computational perspective on the neural basis of multisensory spatial representations. *Nature Reviews Neuroscience 3*(9), pp. 741–747.

Pouget, A., S. Deneve, and T. Sejnowski (1999). Frames of reference in hemineglect: a computational approach. *Progress in Brain Research* .

Pouget, A. and T. J. Sejnowski (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience 9*(2), pp. 222–237.

Pouget, A. and L. H. Snyder (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience 3 Suppl*, pp. 1192–1198.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks 12*(1), pp. 145 – 151.

Riedmiller, M. (1994). Rprop - description and implementation details.

Riedmiller, M. and H. Braun (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE International Conference on Neural Networks*, pp. 586–591.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1988). Neurocomputing: foundations of research. Chapter Learning internal representations by error propagation, pp. 673–695. Cambridge, MA, USA: MIT Press.

Salinas, E. and L. Abbott (2001). Coordinate transformations in the visual system: How to generate gain fields and what to compute with them. *Progress in brain research* , pp. 175–190.

Salinas, E. and L. F. Abbott (1996). A model of multiplicative neural responses in parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America 93*(21), pp. 11956–11961.

Salinas, E. and L. F. Abbott (1997). Invariant visual responses from attentional gain fields. *Journal of Neurophysiology 77*(6), pp. 3267–3272.

Salinas, E. and T. J. Sejnowski (2001). Gain modulation in the central nervous system: Where behavior, neurophysiology, and computation meet. *The Neuroscientist 7*, pp. 430–440.

Salinas, E. and P. Thier (2000). Gain modulation: A major computational principle of the central nervous system. *Neuron 27*, pp. 15–21.

Society for Neuroscience (2012). *Brain facts: a primer on the brain and nervous system*. Society for Neuroscience.

Soechting, J. and M. Flanders (1992). Moving in three-dimensional space: frames of reference, vectors, and coordinate systems. *Annual review of neuroscience 15*, pp. 167–191.

Tikhanoff, V., A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori (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, New York, pp. 57–61.

Wilson, D. R. and T. R. Martinez (2003). The general inefficiency of batch training for gradient descent learning. *Neural Netw. 16*(10), pp. 1429–1451.

Xing, J. and R. A. Andersen (2000). Models of the posterior parietal cortex which perform multimodal integration and represent space in several coordinate frames. *Journal of Cognitive Neuroscience 12*(4), pp. 601–614.

Zhang, J. and L. Abbott (2000). Gain modulation of recurrent networks. *Neurocomputing 32-33*(1-4), pp. 623–628.

Zipser, D. and R. A. Andersen (1988). A backpropagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature 331*, pp. 679–684.

A DVD supplement

We programmed the practical part of this thesis in C++ and our IDE was Microsoft Visual Studio 2010. We used the iCub simulator for generating datasets for training and testing ANN. The training was performed using C++ wrapper for the Fast Artificial Neural Network Library. All visualisations presented in this thesis were made in Matlab 2010. The DVD supplement contains all of these source codes and Matlab scripts, as well as several other visualisations of neural networks. The digital version of this document and readme-file are also included.