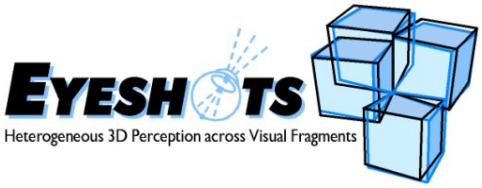


Project no.: FP7-ICT-217077
Project full title: Heterogeneous 3-D Perception across Visual Fragments
Project Acronym: EYESHOTS
Deliverable no: D2.1 (updated)
Title of the deliverable: Convolutional network for vergence control

Date of Delivery:	31 August 2010	
Organization name of lead contractor for this deliverable:	K.U.Leuven	
Author(s):	N. Chumerin, K. Pauwels, M. Van Hulle, A. Gibaldi, M. Chessa, F. Solari, S.P. Sabatini	
Participant(s):	K.U.Leuven, UG	
Workpackage contributing to the deliverable:	WP2	
Nature:	Report	
Version:	2.1 (updated)	
Total number of pages:	62	
Responsible person:	Marc Van Hulle	
Revised by:	Fred Hamker	
Start date of project:	1 March 2008	**Duration:** 36 months

Project Co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:
We present two neural models for vergence angle control of a robotic head, a simplified and a more complex one. Both models work in a closed-loop manner and do not rely on explicitly computed disparity, but extract the desired vergence angle from the post-processed (or raw) response of a population of disparity tuned complex cells, the actual gaze direction and the actual vergence angle. The first model assumes that the gaze direction of the robotic head is orthogonal to its baseline and the stimulus is a frontoparallel plane orthogonal to the gaze direction. The second model goes beyond these assumptions, and operates reliably in the general case where all restrictions on the orientation of the gaze, as well as the stimulus position, type and orientation, are dropped.
Note: This deliverable is based on Deliverable 2.1 (dated 2009-09-09), which was updated with new results and extended with a model for *Vergence-Version Control with Attention effects* (VVCA, see Section 6). Version control is a subtask of WP2 ("Voluntary exploration"), but the VVCA model is a new concept since it integrates version and vergence control. It was originally not planned to be part of Deliverable 2.1. This work is still under development.

Contents

1 Executive summary

One of the objectives of Workpackage 2 is to develop a network-based vergence control from a population of disparity-tuned complex cells. To this end, we investigated the specialization of these disparity detectors at different levels in a hierarchical network architecture to see the effect of learning specific coding and decoding strategies for active vergence control and depth vision. The extraction of binocular features occurs through a cortical-like population network, developed by partner UG. This network (referred in this work as a *disparity detector population* or simply as a *V1 population*) provides a distributed disparity representation to the vergence control network (VC-net).

Using the population responses the proposed VC-net is trained to produce angular vergence control, which in turn is further executed by the oculomotor plant. We propose two types of VC-net paradigms: a *linear* and a *convolutional* one. The conventional convolutional network (LeNet5) architecture has been extended to increase its flexibility by including new functionalities.

We conclude that:

1. The slow (closed loop) vergence eye movements can be controlled using convolutional network even in the case when the gaze is oriented arbitrary.

2. A strategy for reading-out binocular energy population codes for short-latency disparity-vergence eye movements can be devised. Specific features are: (i) wide working range with a reduced number of resources (single scale), (ii) linear servos with fast reaction times and satisfactory precision.

The further generalization of the network paradigm is explored, also with the aim of including (i) kinematic (*i.e.*, in terms of eye rotation velocity) vergence control, and (ii) attentional signals (based on object properties) that might guide intentional exploration of the selected object by performing version eye movements. To achieve the latter, we are currently developing a vergence-version control model, the current status of which we are reporting in Section 6 (note that, according to the Annex I, version control was not intended to be part of this Deliverable). The results of research discussed in this Deliverable were published in:

- N. Chumerin., A. Gibaldi, S.P. Sabatini, M.M. Van Hulle. Learning Eye Vergence Control from a Distributed Disparity Representation. *International Journal of Neural System*, vol. 20, no. 4, pp. 267–278, 2010.

- A. Gibaldi, M. Chessa, A. Canessa, S.P. Sabatini, and F. Solari. A cortical model for binocular vergence control without explicit calculation of disparity. *Neurocomp.*, 73:1065–1073, 2010.

- A. Gibaldi, N. Chumerin, M.M. Van Hulle, S.P. Sabatini. Two Neural Models for Instantaneous Vergence Control. *The 4th International Conference on Cognitive Systems*, Zurich, Switzerland, January 27–28, 2010.

- N. Chumerin, A. Gibaldi, S.P. Sabatini, M.M. Van Hulle. Convolutional Network for Vergence Control. *2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Bratislava, Slovak Republic, November 24–27, 2009.

- M. Chessa, S.P. Sabatini, and F. Solari. A fast joint bioinspired algorithm for optic flow and two-dimensional disparity estimation. In *Proc. International Conference on Computer Vision Systems (ICVS'09)*, Liege, Belgium, October 2009.

- A. Gibaldi, M. Chessa, A. Canessa, S.P. Sabatini, and F. Solari. A neural model for binocular vergence control without explicit calculation of disparity. In *Proc. European Symposium on Artificial Neural Networks (ESANN'09)*, Bruges, Belgium, April 2009.

2 Introduction

Vergence eye movements have the task to align both the left and the right eyes on the same object, in order to allow for the fusion of the binocular image, and thus to produce singleness of vision. Since the eyes are located in slightly different viewpoints, the image of an object in the world is projected on the retinas at different positions, and this difference is defined as retinal disparity, which is the cue used for vergence. In fact, both eyes rotate in opposite directions according to the retinal disparity, depending on the sign of the disparities either convergence or divergence is elicited (in Figure 1, by convention, a positive vergence leads to convergence, and *vice versa*), so as to achieve and/or maintain the singleness of vision. In this study, we consider a stereo setup consisting of a fixed robotic head with a pair of eyes (see Figure 1). The task is to estimate, and then to maintain the vergence angle that brings the fixation point, along the gaze direction, onto the surface of the observed object.

Experimental evidence shows that, although depth perception and vergence eye movements are based on the activity of complex cells of the primary visual cortex, the brain adopts specific and separate mechanisms to combine binocular information and carry out the two distinct tasks. Vergence control models that are based on a distributed population of disparity detectors, usually require first the computation of the disparity map, thus limiting the functionality of the vergence system inside the sensitivity range of the population of cells specialized for depth perception. As for the control of vergence, larger disparities have to be supported, while keeping a good accuracy around the fixation point to achieve a stable fixation, alternative strategies might be employed. In this work, we developed models that combine the population responses without taking a decision, but extracting, directly from the population responses, a disparity-vergence response that allows us to nullify the disparity in the fovea, even if the stimulus presented is far beyond the disparity sensitivity range. The disparity-vergence response is obtained by a weighted combination of the population response. First, the weights were computed in order to obtain the desired set of disparity-vergence responses on which to base a 'dual-mode' vergence control mechanism; then the weights were directly learned from examples of

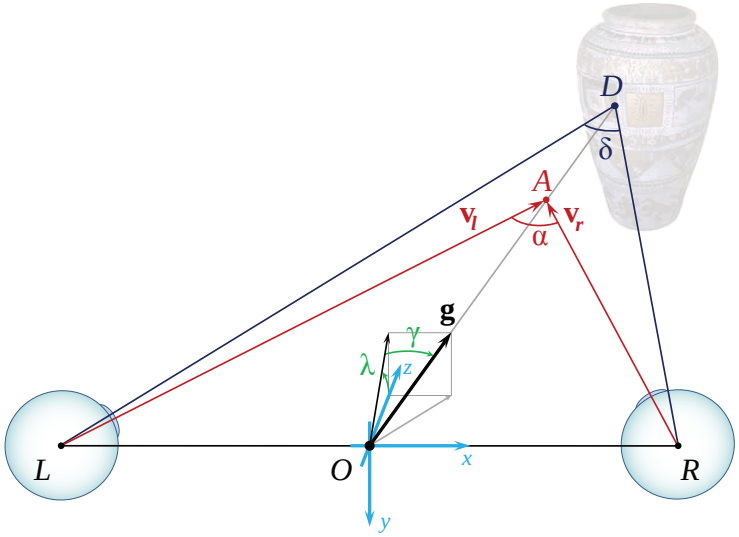


Figure 1: The geometry of the robotic head model. L and R are the nodal centers of the eyes, O is the middle-point of the baseline LR; A is the *actual fixation point*, $|OA|$ is the *actual distance* and α is *actual vergence angle*; D is the *desired fixation point*, $|OD|$ is the *desired distance* and δ *desired vergence angle*. The *gaze direction* is defined as the direction from point O to the fixation point A (and/or D), and depicted by a unit vector $\mathbf{g}$, which, in a headcentric coordinate system $Oxyz$, is specified by a pair of angles γ (pan/yaw) and λ (tilt/elevation). The orientation of the left and right eye visual axes is specified by the vectors $\mathbf{v}_l = \overrightarrow{LA}$ and $\mathbf{v}_r = \overrightarrow{RA}$ respectively.

the desired vergence behaviour. We tested the proposed model in a virtual environment achieving stable fixation and small response time to a wide range of disparities. The vergence movements produced are able bring and to keep the fixation point both on a steady and on a moving stimulus. Section 3 and Section 4, respectively, report on the basic population network of disparity detectors and the proposed 'dual-mode' strategy for binocular vergence, devised by UG. Section 5 reports on the two networks (linear and convolutional) developed by K.U.Leuven to learn disparity-vergence behaviours on the basis of the population responses.

3 Distributed representation of binocular disparity

3.1 Computational theory

3.1.1 Multichannel band-pass representation of the visual signal

An efficient (internal) representation is necessary to guarantee all potential visual information can be made available for higher level analysis. At an early level, feature detection occurs through initial local *quantitative* measurements of basic image properties (*e.g.*, edge, bar, orientation, movement, binocular disparity, colour) referable to spatial differential structure of the image luminance and its temporal evolution (*cf.*, linear cor-

tical cell responses). Later stages in vision can make use of these initial measurements by combining them in various ways, to come up with categorical *qualitative* descriptors, in which information is used in a non-local way to formulate more global spatial and temporal predictions. The receptive fields of the cells in the primary visual cortex have been interpreted as fuzzy differential operators (or local *jets* [1]) that provide regularized partial derivatives of the image luminance in the neighborhood of a given point $\mathbf{x} = (x, y)$, along different directions and at several levels of resolution, simultaneously. Given the 2D nature of the visual signal, the spatial direction of the derivative (*i.e.*, the orientation of the corresponding local filter) is an important "parameter". Within a local jet, the directionally biased receptive fields are represented by a set of similar filter profiles that merely differ in orientation.

Alternatively, considering the space/spatial-frequency duality [2], the local jets can be described through a set of independent spatial-frequency channels, which are selectively sensitive to a different limited range of spatial frequencies. These spatial-frequency channels are equally apt as the spatial ones. From this perspective, it is formally possible to derive, on a local basis, a complete harmonic representation (phase, energy/amplitude, and orientation) of any visual stimulus, by defining the associated analytic signal in a combined space-frequency domain through filtering operations with complex-valued band-pass kernels. Formally, due to the impossibility of a direct definition of the analytic signal in two dimensions, a 2D spatial frequency filtering would require an association between spatial frequency and orientation channels. Accordingly, for each orientation channel θ, an image $I(\mathbf{x})$ is filtered with a complex-valued filter:

$$f_A^\theta(\mathbf{x}) = f^\theta(\mathbf{x}) - i f_\mathcal{H}^\theta(\mathbf{x}) \tag{1}$$

where $f_\mathcal{H}^\theta(\mathbf{x})$ is the Hilbert transform of $f^\theta(\mathbf{x})$ with respect to the axis orthogonal to the filter's orientation. This results in a complex-valued *analytic image*:

$$Q_A^\theta(\mathbf{x}) = I * f_A^\theta(\mathbf{x}) = C_\theta(\mathbf{x}) + i S_\theta(\mathbf{x}) \ , \tag{2}$$

where $C_\theta(\mathbf{x})$ and $S_\theta(\mathbf{x})$ denote the responses of the quadrature filter pair. For each spatial location, the amplitude $\rho_\theta = \sqrt{C_\theta^2 + S_\theta^2}$ and the phase $\phi_\theta = \arctan(S_\theta / C_\theta)$ envelopes measure the harmonic information content in a limited range of frequencies and orientations to which the channel is tuned.

In the harmonic space, it is in general an important requirement to have both the spatial width of the filters and the spatial frequency bandwidth small, so that good localization and good approximation of the harmonic information is realized simultaneously. Gabor functions reaching the maximal joint resolution in space and spatial frequency domains are specifically suitable for this purpose and are extensively used in computational vision [2]. Different band-pass filters have been proposed as an alternative to Gabor functions, on the basis of specific properties of the basis functions [3–10], or according to theoretical and practical considerations of the whole space-frequency transform [11–16]. A detailed comparison of the different filters evades the scope of this report and numerous comparative reviews can be already found in the literature (*e.g.*, see [17–19]).

We have considered a discrete set of oriented Gabor filters with different angles θ. To generate a filter with orientation θ (measured from the positive horizontal axis), we can

rotate a vertically oriented filter by $\theta - 90°$ with respect to the filter's center (positive angle means counterclockwise rotation):

$$g(\mathbf{x}, \theta, \psi) = \eta \cdot \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x_\theta^2}{2\sigma_x^2} - \frac{y_\theta^2}{2\sigma_y^2}\right) \mathrm{cis}(k_0 x_\theta + \psi) \tag{3}$$

with

$$\begin{cases} x_\theta = x\cos(\theta - 90°) + y\sin(\theta - 90°) \\ y_\theta = -x\sin(\theta - 90°) + y\cos(\theta - 90°) \end{cases}$$

k_0 denotes the *radial peak frequency*, ψ relates to the filter symmetry, and σ's relates to the spatial filter extension. The parameter η is a proper normalization constant (*e.g.*, chosen to the unitary maximum condition or to the unitary energy condition). Equivalently, the set of Gabor filters can be defined by a quadratic form as:

$$g(\mathbf{x}, \theta, \psi) = \eta \cdot \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\right) \mathrm{cis}(\mathbf{k}_0^T\mathbf{x} + \psi) \tag{4}$$

where $\mathbf{k}_0 = (k_0\sin\theta, -k_0\cos\theta)^T$ is the oriented spatial frequency vector[1], and the matrix $\mathbf{A}$ can be derived from a diagonal matrix $\mathbf{D}$ (corresponding to a vertically oriented Gabor filter) by multiplication with the rotation matrix $\mathbf{\Theta}$:

$$\mathbf{A} = \mathbf{\Theta}^T\mathbf{D}\mathbf{\Theta} = \begin{pmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} \sigma_x^{-2} & 0 \\ 0 & \sigma_y^{-2} \end{pmatrix} \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix}. \tag{5}$$

It is worth noting that the peak radial frequency k_0 and the width σ_x of the Gaussian envelope in the Gabor function are linked by the constant relative bandwidth factor β (in octave)[2] as:

$$\sigma_x = \frac{1}{k_0}\left(\frac{2^\beta + 1}{2^\beta - 1}\right). \tag{6}$$

Typically, β is chosen around 1 ($\beta \in [0.8, 1.2]$). The relative bandwidth constancy yields self-similar filters across the scales: filters with different radial peak frequencies, but identical orientation angle are simply geometrically scaled version of each other. The aspect ratio σ_x/σ_y normally takes values between 0.25 and 1 and, together with the radial peak frequency, defines the orientation bandwidth of the filter[3]. In the following, to bind the orientation bandwidth of the filter to the presence of the sinusoidal term only, we fix the aspect ratio to 1 (*i.e.*, $\sigma_x = \sigma_y = \sigma$).

[1]The orientation of the Gabor filter in space and the orientation of the bandpass channel in the frequency domain are related by $\theta = \arg(\mathbf{k}_0) + \frac{\pi}{2}$.

[2]The relative bandwidth of a Gabor filter is defined as

$$\beta = \log_2\left(\frac{k_0 + \Delta k/2}{k_0 - \Delta k/2}\right) = \log_2\left(\frac{k_0\sigma_x + 1}{k_0\sigma_x - 1}\right)$$

when one chooses the cut-off frequency at one-standard-deviation of the amplitude spectrum of the Gabor function ($1/\sigma_x$) to define the absolute bandwidth Δk.

[3]The orientation bandwidth is the angle between two lines that pass through the frequency origin and are tangent to the one-standard-deviation contour of the amplitude spectrum of the Gabor function.

The values of all the design parameters have been chosen to have a good coverage of the space-frequency domain, to guarantee a uniform orientation coverage and to keep the spatial support to a minimum, in order to cut down the computational cost. Therefore, we determined the smallest filter on the basis of the highest allowable frequency without aliasing, and we doubled the sampling when the model analysis requires a higher precision in the filter's profile (or, from a different perspective, a larger spatial support in pixels). [Note: this design strategy reveals itself particularly effective for economic multi-scale analysis through pyramidal techniques [20]. Yet, for all the simulations conducted in this work we considered a single scale, only]. Accordingly, we fixed the maximum radial peak frequency (k_0) by considering the Nyquist condition and a constant relative bandwidth β around one octave, that allows us to cover the frequency domain without loss of information. The result was a minimal 11×11 filter mask capable of resolving sub-pixel phase differences. To satisfy the quadrature requirement all the even symmetric filters have been "corrected" to cancel the DC sensitivity. The filters have been expressed as sums of x-y separable functions to implement separate 1D convolutions instead of 2D convolutions in a similar way that [21], with a consequent further drop of the computational burden. For a detailed description of the filters used, see the Appendix A.

3.1.2 Phase-based disparity detection

Depth perception derives from the differences in the positions of corresponding points in the stereo image pair projected on the two retinas of a binocular system. When the camera axes are parallel, on the basis of a local approximation of the Fourier Shift Theorem, the phase-based stereopsis defines the disparity $\delta(\mathbf{x})$ as the one-dimensional (1D) shift necessary to align, along the direction of the horizontal epipolar lines, the phase values of bandpass filtered versions of the stereo image pair $I^R(\mathbf{x})$ and $I^L[\mathbf{x} + \delta(\mathbf{x})]$ [22]. In general, this type of local measurement of the phase results in stable, and a quasilinear behaviour of the phase vs. space is observed over relatively large spatial extents, except around singular points where the amplitudes $\rho(\mathbf{x})$ vanishes and the phase becomes unreliable [23]. This property of the phase signal yields good predictions of binocular disparity by

$$\delta(\mathbf{x}) = \frac{\lfloor \phi^L(\mathbf{x}) - \phi^R(\mathbf{x}) \rfloor_{2\pi}}{k(\mathbf{x})} = \frac{\lfloor \Delta\phi(\mathbf{x}) \rfloor_{2\pi}}{k(\mathbf{x})}, \tag{7}$$

where $k(\mathbf{x})$ is the average instantaneous frequency of the bandpass signal, measured by using the phase derivative from the left and right filter outputs:

$$k(\mathbf{x}) = \frac{\phi_x^L(\mathbf{x}) + \phi_x^R(\mathbf{x})}{2}. \tag{8}$$

As a consequence of the linear phase model, the instantaneous frequency is generally constant and close to the tuning frequency of the filter ($\phi_x \simeq k_0$), except near singularities

It is given by

$$B_\theta = \arctan\left(\frac{2^\beta - 1}{2^\beta + 1}\right).$$

where abrupt frequency changes occur as a function of spatial position. Therefore, a disparity estimate at a point $\mathbf{x}$ is accepted only if $|\phi_x - k_0| < k_0\mu$, where μ is a proper threshold [23].

Equivalently, the principal part of the interocular phase difference necessary to estimate the binocular disparity can be obtained directly, without explicit manipulation of the left and right phase and thereby without incurring the 'wrapping' effects on the resulting disparity map [24] (see also [25, 26]):

$$\begin{aligned} \lfloor \Delta\phi \rfloor_{2\pi} &= \arg(Q^L Q^{*R}) && (9)\\ &= \text{atan2}\left(\text{Im}(Q^L Q^{*R}), \text{Re}(Q^L Q^{*R})\right) && (10)\\ &= \text{atan2}\left(C^R S^L - C^L S^R, C^L C^R + S^L S^R\right) && (11) \end{aligned}$$

where $Q^L = Q^L(\mathbf{x}) = I^L * g(\mathbf{x}, 0°, \psi)$, $Q^R = Q^R(\mathbf{x}) = I^R * g(\mathbf{x}, 0°, \psi)$ and Q^* denotes complex conjugate of Q.

When the camera axes are moving freely, as it occurs in a binocular active vision system, stereopsis cannot longer be considered a 1D problem and the disparities can be both *horizontal* and *vertical*. Therefore, the 1D phase difference approach must be extended to the 2D case.

Still relying upon the local approximation of the Fourier Shift Theorem, the 2D local vector disparity $\boldsymbol{\delta}(\mathbf{x})$ between the left and right images can be related/detected as a phase shift $\mathbf{k}^T(\mathbf{x})\boldsymbol{\delta}(\mathbf{x})$ in the local spectrum, where $\mathbf{k}(\mathbf{x})$ is the local (*i.e.*, instantaneous) frequency vector defined as the phase gradient:

$$\mathbf{k}(\mathbf{x}) = \nabla\phi(\mathbf{x}) = \left(\frac{\partial\phi(x,y)}{\partial x}, \frac{\partial\phi(x,y)}{\partial y}\right)^T \tag{12}$$

with

$$\phi(\mathbf{x}) = \frac{\phi^L(\mathbf{x}) + \phi^R(\mathbf{x})}{2}.$$

Given the 1D character of both the local phase and the instantaneous frequency, their measures strictly depend on the choice of one reference orientation axis, thus preventing the determination of the full disparity vector by a punctual single-channel measurement. We will see that only the projected disparity component on the direction orthogonal to the dominant local orientation of the filtered image can be detected.

Let us distinguish two cases. When the image (stimulus) structure is intrinsically 1D, with a dominant orientation θ_s (let us think of an oriented edge or of an oriented grating with frequency vector $\mathbf{k}_s = (k_s \sin\theta_s, k_s \cos\theta_s)^T$, as extreme cases), the aperture problem [27] restricts detectable disparity to the direction orthogonal to the edge (*i.e.*, to the direction of the dominant frequency vector $\mathbf{k}_s$):

$$\boldsymbol{\delta}_{\theta_s}(\mathbf{x}) = \frac{\mathbf{k}_s}{k_s}\frac{\lfloor \Delta\phi_{\theta_s}(\mathbf{x}) \rfloor_{2\pi}}{k(\mathbf{x})} \simeq \frac{\mathbf{k}_s}{k_s}\frac{\lfloor \Delta\phi_{\theta_s}(\mathbf{x}) \rfloor_{2\pi}}{k_s} \tag{13}$$

where $k(\mathbf{x})$ is the magnitude of the instantaneous frequency. That is, only the projection $\boldsymbol{\delta}_{\theta_s}$ of the disparity $\boldsymbol{\delta}$ onto the direction of the stimulus frequency $\mathbf{k}_s$ is observed. A

spatial disparity in a direction orthogonal to $\mathbf{k}_s$ cannot be measured. For an intrinsic 1D image structure, indeed, the spectrum energy is confined within a very narrow bandwidth and it is gathered by the bandwidth $(\Delta k, B_\theta)$ of a single activated channel. This is a realistic assumption for a relatively large number of orientation channels. Moreover, in these condition, when the dominant frequency of the stimulus $\mathbf{k}_s$ is unknown, it can be approximated by k_0, and thus Eq. (13) becomes:

$$\boldsymbol{\delta}_{\theta_s}(\mathbf{x}) \sim \frac{\mathbf{k}_0}{k_0} \frac{\lfloor \Delta\phi_{\theta_s}(\mathbf{x}) \rfloor_{2\pi}}{k_0}. \tag{14}$$

When the image structure is intrinsically 2D (let us think of a rich texture or a white noise, as an extreme case), the visual signal has local frequency components in more than one direction and the dominant direction is given by the orientation of the Gabor filter. Similarly, the only detectable disparity by a band-pass oriented channel $(\Delta k, B_\theta)$ is the one orthogonal to the filter's orientation θ, *i.e.*, the projection in the direction of the filter's frequency:

$$\boldsymbol{\delta}_{\theta}(\mathbf{x}) = \frac{\mathbf{k}_0}{k_0} \frac{\lfloor \Delta\phi_{\theta}(\mathbf{x}) \rfloor_{2\pi}}{k(\mathbf{x})}. \tag{15}$$

Again, $k(\mathbf{x})$ can be derived by Eq. (12) or approximated by the peak frequency of the Gabor filter $\mathbf{k}_0$.

By considering the whole set of oriented filters, we can derive the projected disparities in the directions of all the frequency components of the multi-channel band-pass representation, and obtain the full disparity vector by intersection of contraints [28], thus solving the aperture problem. Without measurement errors, the vector disparity determined by each orientation channel consists of projection $\boldsymbol{\delta}_\theta(\mathbf{x})$ in $\mathbf{k}_0$-direction and unknown orthogonal component (see Figure 2). The full disparity vector $\boldsymbol{\delta}(\mathbf{x})$ can be recovered from at least two projections $\boldsymbol{\delta}_\theta(\mathbf{x})$, which are not linearly dependent. The end points of the vectors $\boldsymbol{\delta}_\theta(\mathbf{x})$ for fixed $\mathbf{k}_0$ are located on a circle through the origin and the end point of $\boldsymbol{\delta}_\theta(\mathbf{x})$. Taking into account measurement errors of $\Delta\phi_\theta$ and , the redundancy of more than two projections can be used to minimize the mean square error for $\boldsymbol{\delta}(\mathbf{x})$:

$$\boldsymbol{\delta}(\mathbf{x}) = \underset{\boldsymbol{\delta}(\mathbf{x})}{\operatorname{argmin}} \sum_\theta c_\theta(\mathbf{x}) \left(\delta_\theta(\mathbf{x}) - \frac{\mathbf{k}_0^T}{k_0} \boldsymbol{\delta}(\mathbf{x}) \right)^2. \tag{16}$$

where the coefficent $c_\theta(\mathbf{x}) = 1$ when the component disparity along direction θ for pixel $\mathbf{x}$ is a *valid* component on the basis of a confidence measure, and is null otherwise. In this way, the influence of erroneous filter responses is reduced.

3.2 Distributed models

The phase-based disparity estimation approach presented in Section 3.1.2 implies *explicit* measurements, for each spatial orientation channel θ (and for any given scale) of the local phase difference $\Delta\phi$ in the image pairs, from which we obtain the *direct* measure of the binocular disparity component δ_θ. Similarly, we can consider a distributed approach in which the binocular disparity δ is never measured but implicitly coded by the population

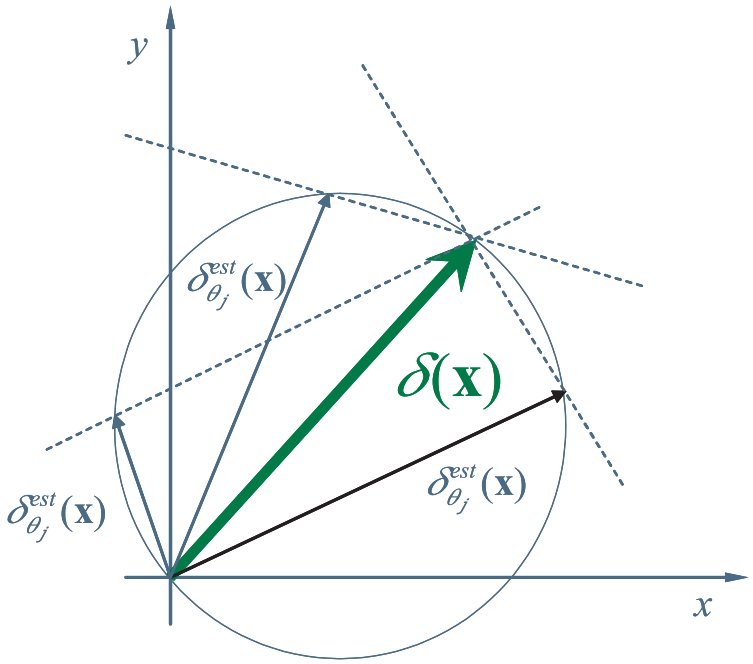


Figure 2:

activity of cells that act as "disparity detectors" - over a proper range of disparity values. Such models are inspired by the experimental evidences on how the brain and, specifically, the primary visual cortex (V1), implements early mechanisms for stereopsis. Using such a distributed code it is possible to achieve a very flexible and robust representation of binocular disparity for each spatial position in the retinal image.

3.2.1 Phase-shift and binocular energy models

An abundance of neurophysiological evidences report that the cortical cells' sensitivity to binocular disparity is related to interocular phase shifts in the Gabor-like receptive fields of V1 simple cells [22, 29–33]. It is worth noting that models based on a difference in the position of the left and right RFs (position-shift models) or hybrid approaches have been proposed (we will discuss the consequences of this model extensions at the end of this Section). The phase-shift model posits that the center of the left and right eye RFs coincides, but the arrangements of the RF subregions are different. Formally, the response of a simple cell with RF center in $\boldsymbol{x}$ and oriented along θ, can be written as:

$$ {}^{\theta}_{\Delta\psi}r_{s,\psi_0}(\mathbf{x}) = I^L * h^L(\mathbf{x};\theta,\psi_0+\psi^L) + I^R * h^R(\mathbf{x};\theta,\psi_0+\psi^R) \tag{17} $$

where

$$ h(\mathbf{x}) = h(\mathbf{x};\theta,\psi) = \eta \exp\left(-\frac{1}{2\sigma^2}\mathbf{x}^T\mathbf{x}\right)\cos(\mathbf{k}_0^T\mathbf{x}+\psi) \tag{18} $$

is a real-valued RF (*cf.* Eq. (4)), ψ_0 is a "central" value of the phase of the RF, and ψ^L and ψ^R are the phases that characterize the binocular RF profile.

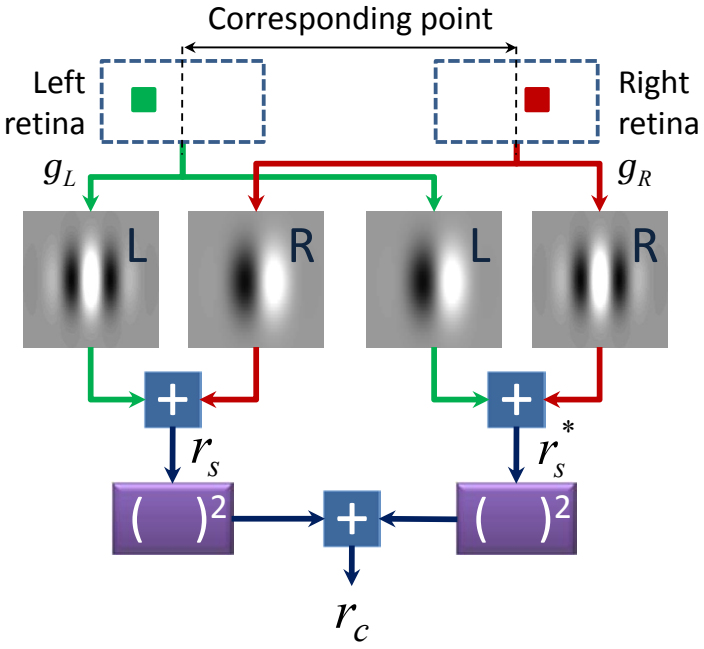


Figure 3: The complex cell response is constructed as the squared sum of a quadrature pair of simple cells. The green and red pathways relate to the monocular "quadrature pair" of simple cell RFs, g^L and g^R, respectively.

In order to make the disparity tuning independent of the monocular local Fourier phase of the images (but only on the interocular phase difference), binocular energy complex cells play the role. Such "energy units" are defined as the squared sum of a quadrature pair of simple cells (see Figure 3) and their response is defined as:

$$ {}^{\theta}_{\Delta\psi} r_c(\mathbf{x}) = {}^{\theta}_{\Delta\psi} r^2_{s,0}(\mathbf{x}) + {}^{\theta}_{\Delta\psi} r^2_{s,\pi/2}(\mathbf{x}) \tag{19} $$

Linking phase-based and energy-based models For any fixed orientation, if we characterize a "quadrature pair" of simple cells by a complex-valued RF (*cf.* Equation 4):

$$ \mathrm{h}(\mathbf{x}) \stackrel{\triangle}{=} h_C(\mathbf{x}) + j h_S(\mathbf{x}) = g(\mathbf{x};\psi) \tag{20} $$

then we can write the expression of the response of the "quadrature pair" as:

$$ \begin{aligned} Q(\mathbf{x}) &= I^L * g^L(\mathbf{x}) + I^R * g^R(\mathbf{x}) = I^L * g(\mathbf{x}) e^{j\psi^L} + I^R * g(\mathbf{x}) e^{j\psi^R} = \\ &= Q^L(\mathbf{x}) e^{j\psi^L} + Q^R(\mathbf{x}) e^{j\psi^R}. \end{aligned} $$

The response of a complex "energy" cell is then

$$ \begin{aligned} {}^{\theta}_{\Delta\psi} r_c(\mathbf{x}) &= \left| {}^{\theta}_{\Delta\psi} r_{s,0}(\mathbf{x}) + {}^{\theta}_{\Delta\psi} r_{s,\pi/2}(\mathbf{x}) \right|^2 = \left| Q^L(\mathbf{x}) e^{j\psi^L} + Q^R(\mathbf{x}) e^{j\psi^R} \right|^2 = \\ &= \left| e^{j\psi^L} \left(Q^L(\mathbf{x}) + Q^R(\mathbf{x}) e^{j\Delta\psi} \right) \right|^2 = \left| Q^L(\mathbf{x}) + Q^R(\mathbf{x}) e^{j\Delta\psi} \right|^2 \end{aligned} \tag{21} $$

where $\Delta\psi = \psi^L - \psi^R$. Therefore, complex cells' responses depend on $\Delta\psi$ only, instead of on ψ^L and ψ^R individually.

Equation 21 formally establishes the equivalence between phase-based techniques and energy-based models [34]. Indeed, the maximum of r_c responses is obtained when the two phasors Q^L and Q^R are aligned in the complex plane, that is when $\Delta\psi$ compensates for the different Fourier phases of the right and left image patches within the cell's RF (*cf.* [22]).

Notwithstanding the formal equivalence between phase-based techniques and energy-based models, the latter prove themselves more robust to noise and more flexible, since they can intrinsically embed adaptive mechanisms both at coding and decoding levels of the population code. From algebraic and trigonometric manipulation we can derive the tuning curve of the complex cell:

$$\overset{\theta}{\underset{\Delta\psi}{}}r_c(\mathbf{x}) = |Q^L(\mathbf{x})|^2 + 2|Q^L(\mathbf{x})Q^{*R}(\mathbf{x})|\cos(\delta^\theta k_0 - \Delta\psi) + |Q^R(\mathbf{x})|^2. \tag{22}$$

Accordingly, the stimulus disparity, along direction θ, to which the cell is tuned is:

$$\delta^\theta_{pref}(\mathbf{x}) = \frac{\lfloor \Delta\psi(\mathbf{x}) \rfloor_{2\pi}}{k_0}. \tag{23}$$

Including position shift: hybrid models The position-shift model posits that there is a population of energy neurons with different receptive field position shifts. Accordingly we can consider a family of binocular energy neurons whose right monocular subfield is shifted by a set of distances d compared to the retinal position of the left monocular subfield. Usually position-shift are used in combination with phase-shift models to overcome the restriction on the maximum disparity detectability stemmed by the fact that the phase shifts are unique only between $-\pi$ and π. These hybrid models posit that there is a population of binocular energy neurons with different RF positions and different RF phase shifts. In the following we will restrict our analysis to phase-shift model only, and we will deserve a model extension for future work.

3.2.2 Characterization of the population of disparity detectors

Coding Disparity information is extracted from a sequence of stereo image pairs by using a distributed cortical architecture that resorts to a population of simple and complex cells. The population is composed of cells sensitive to $N_p \times N_o$ vector disparities $\boldsymbol{\delta} = (\delta_H, \delta_V)$ with N_p magnitude values distributed in the range $[-\Delta, \Delta]$ pixels and along N_o orientations uniformly distributed between 0 and π (see Figure 4). For each simple cell we can control the ocular dominance of the binocular receptive field $h(\mathbf{x})$, its orientation θ with respect to the horizontal axis and the interocular phase shift $\Delta\psi$ along the rotated axis, which confers to the cell its specific tuning to a disparity $\delta^\theta_{pref} = \Delta\psi_\theta/k_0$, along the direction orthogonal to θ. The spatial frequency k_0 and the spatial envelope are fixed on the basis of the design criteria described in Section 3.1. The complex cell inherits the spatial properties of the simple cells, and its response $r^{ij}_c(\mathbf{x})$ is given by Equation 21: For each orientation, the population is, in this way, capable of providing reliable disparity

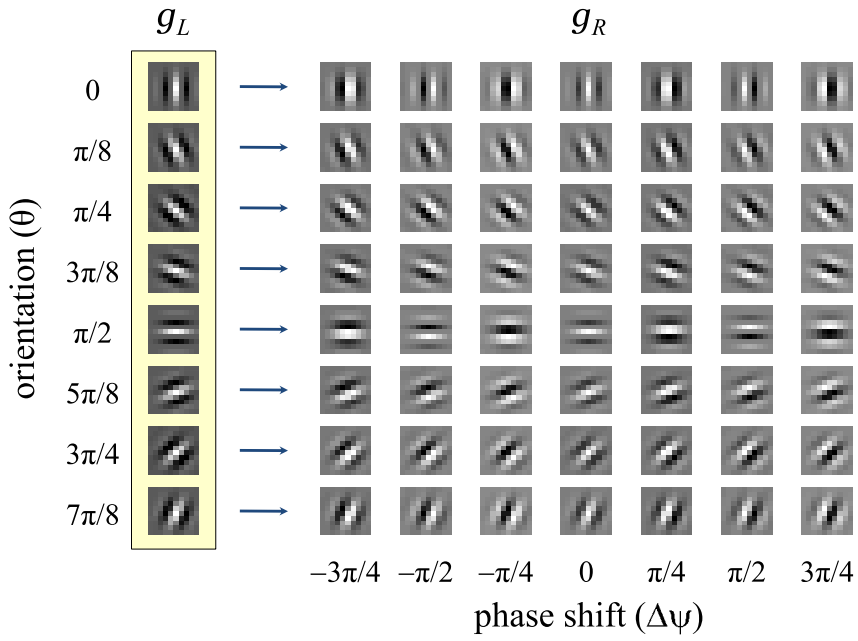


Figure 4: The population of binocular receptive fields for each retinal location.

estimates in the range between $-\Delta$ and Δ, where $\Delta = \Delta\psi_{max}/k_0$ can be defined as the maximum detectable disparity of the population.

Figure 6 shows examples of tuning curves obtained from the population network, compared to the variety of tuning curves observed experimentally in V1 cortical cells [33].

Decoding Once the disparity along each spatial orientation have been coded by the population activity, it is necessary to read out this information, to obtain a reliable estimate. The decoding strategy, the number of the cells in the population and their distribution are jointly related. To decode the population by a winners-take-all strategy, a large number of cells along each spatial orientation would be necessary, thus increasing the computational cost and the memory occupancy of the approach. To obtain precise feature estimation, while keeping the number of cells as low as possible, thus an affordable computational cost, a *weighted sum* (*i.e.*, a center of gravity) of the responses for each orientation is calculated. The *component disparity* $\delta^{est}_{\theta_j}$ is obtained by:

$$\delta^{est}_{\theta_j} = \frac{\sum_{i=1}^{N_p} \frac{\Delta\psi_i}{k_0 \cos\theta_j} r_c^{ij}}{\sum_{i=1}^{N_p} r_c^{ij}} \tag{24}$$

Other decoding methods [35], such as the *maximum likelihood* estimator, have been considered, but the center of gravity of the population activity is the best compromise between simplicity, low computational cost and accuracy of the estimates.

Confidence values, based on local energy, are used to provide a reliability measure for each disparity estimate.

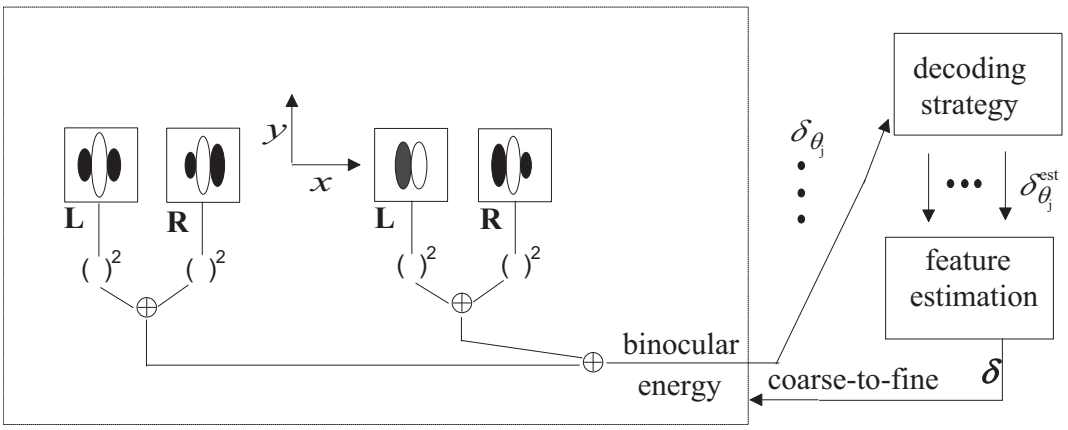
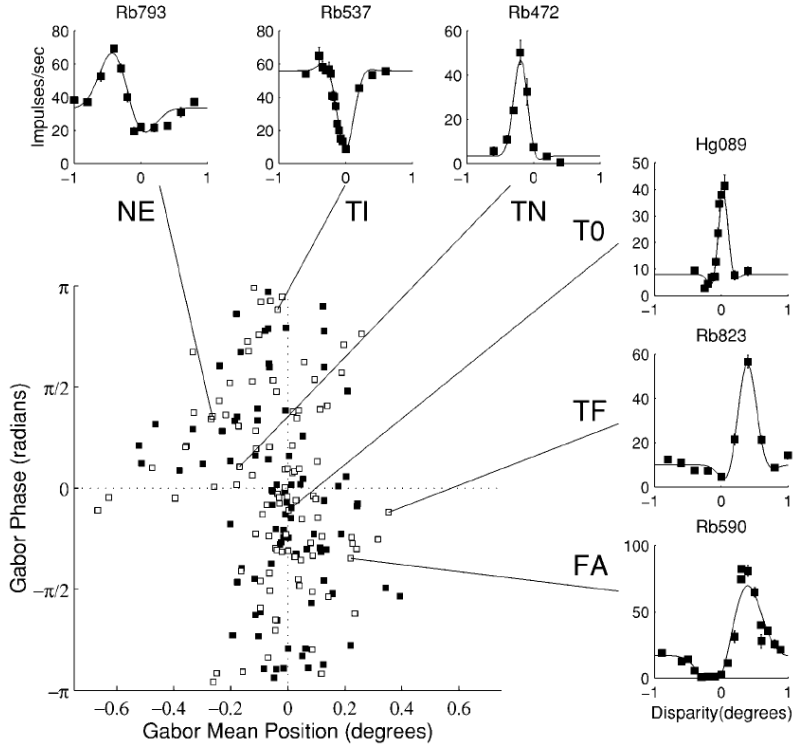


Figure 5: Basic scheme of the neuromorphic architecture for the computation of the 2D disparity.

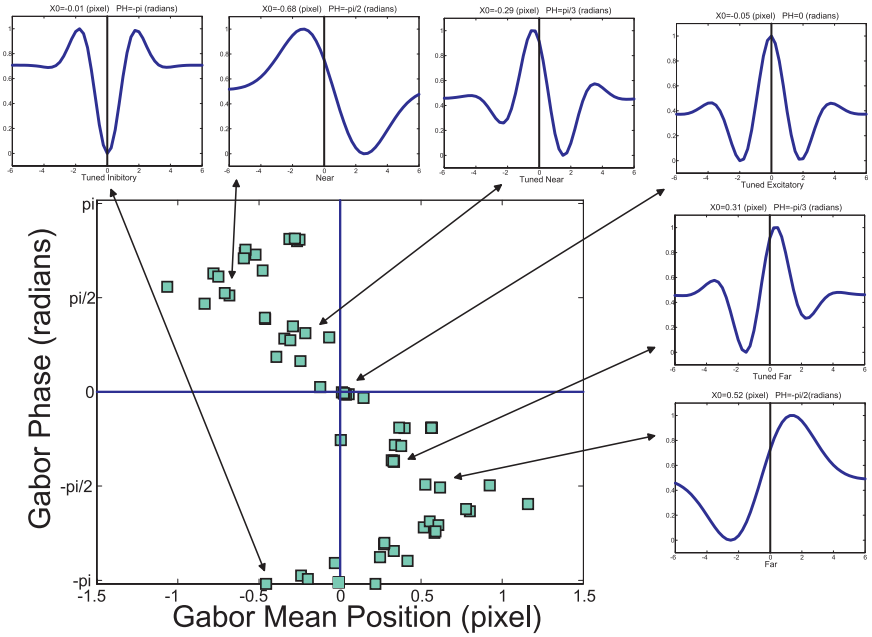
To decode the full (horizontal and vertical) disparity we can still rely on the intersection of constraints (channel interaction) introduced in Section 3.1.2 that combine the population estimates for each orientation channel.

Summarizing, on the basis of these principles, a cortical-like architecture for disparity estimation can be devised. The overall scheme of the proposed architecture is shown in Figure 5. Three distinct levels of processing can be distinguished: (1) the distributed coding of disparity across different orientation channels, (2) the decoding stage for each channel, and (3) the estimation of the full disparity through channel interaction. If one wants to consider several scales, coarse-to-fine strategies can be straightforwardly embodied, *e.g.*, by including in the scheme a refinement loop as re-entrant connections in the filtering stage (see [36, 37]).

Toward a generalized architecture for active stereopsis In active stereopsis, besides handling horizontal and vertical disparities, we have to explicitly consider vergence mechanisms in the processing loop. From this perspective, in the next Section, we address the problem of the refinement of vergence, which does not necessarily implies first a refinement of the estimation of the disparity map. Indeed, experimental evidences (see *e.g.*, [38–40]) pointed out that mechanisms guiding eye movements are in general different from those supporting depth perception. We will see that, by specializing disparity detectors for vergence control, we can obtain linear servos with fast reaction and precision that work over a wide range of disparities with a reduced number of resources single scale).



(a)



(b)

Figure 6: (a) Distribution of the tuning curves obtained from the population network, compared to (b) the observed for real V1 cortical cells [33].

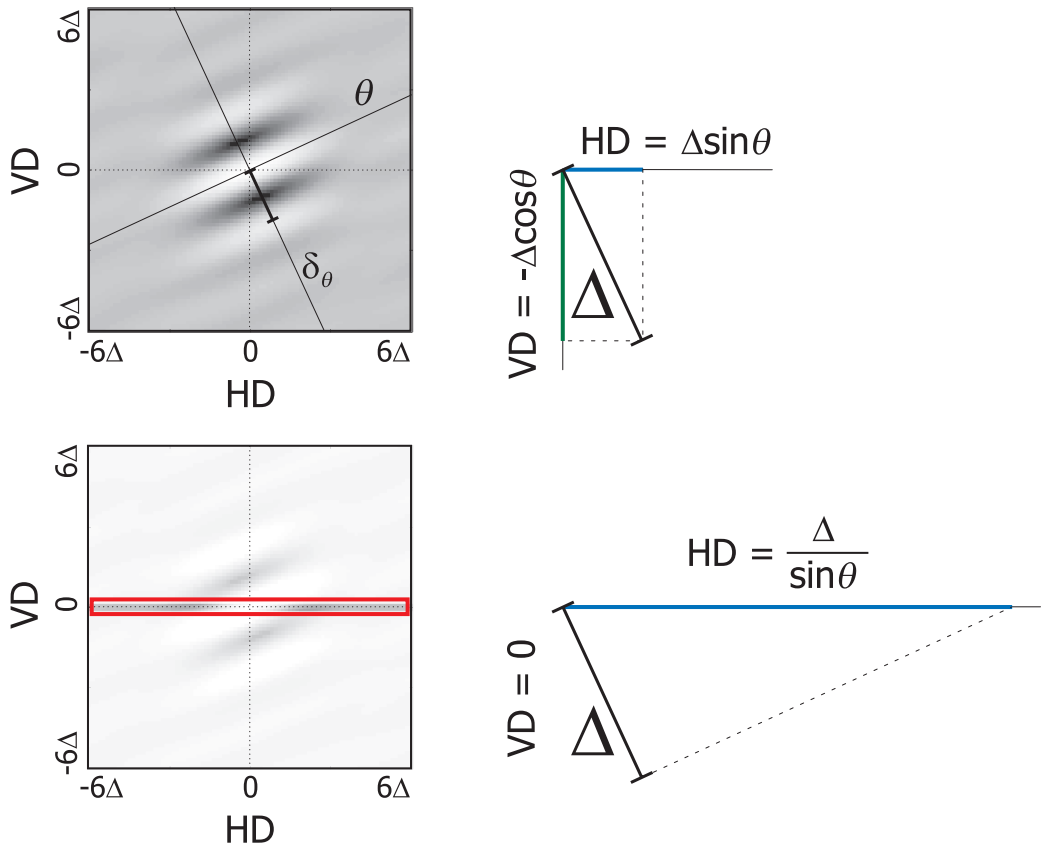


Figure 7: Each complex cell is, by construction, tuned to an oriented disparity δ_θ, *i.e.*, each cell is jointly tuned to horizontal (HD) and vertical (VD) disparities. (Top): For each oriented disparity, its contribution to the HD and VD is calculated by projections on the horizontal and vertical lines. (Bottom): By assuming VD= 0, the orientation of the RF is used as a degree of freedom to extend the sensitivity range of the cell to horizontal disparity stimuli (HD).

4 Strategies for vergence without explicit calculation of disparity

4.1 Reading binocular energy population codes for short-latency disparity-vergence eye movements

As described in Section 3, the population of complex cells are, by construction, tuned to oriented disparities, *i.e.*, jointly tuned to horizontal (δ_H) and vertical disparities (δ_V). In general, indeed, the retinal disparity is a two-dimensional (2D) feature and the full decoding of the population response would require the proper solution of the aperture problem [27]. This can be achieved, by example, through the intersection of the constraints provided by the different orientation channels (*cf.* [28]). If one proceeds in such a way, that is by recovering the full disparity vector, the disparity detectability range

would still be limited to $\pm\Delta$, and the horizontal (vertical) component of the full disparity vector will then used for the control of horizontal (vertical) vergence. Unless one uses computationally expensive multiscale techniques for widening the disparity detectability range, this approach would considerably limit the working range of the vergence control. As for the control of vergence, larger disparities have to be discriminated while keeping a good accuracy around the fixation point for allowing finer refinement and achieving stable fixations, alternative strategies might be employed to gain effective vergence signals directly from the complex cell population responses, without explicit computation of the disparity map. To this end, we can map the 2D disparity feature space into the 1D space of the projected horizontal disparities, where the orientation θ plays the role of a parameter. More precisely, by assuming $\delta_V = 0$, the dimensionality of the problem of disparity estimation reduces to one, and the orientation of the receptive field is used as a degree of freedom to extend the sensitivity range of the cells' population to horizontal disparity stimuli (see Figure 7). In this way, each orientation channel has a sensitivity for the horizontal disparity that can be obtained by the projection of the oriented phase difference on the (horizontal) epipolar line in the following way:

$$\delta_H^{\theta} = \frac{\Delta\psi}{2\pi k_0 cos\theta} \tag{25}$$

Figure 8a shows the horizontal disparity tuning curves obtained of the population for different orientations of the receptive fields. To decode the horizontal disparity at a specific image point, the whole activity of the population of cells, with receptive fields centered in that location, is considered. By using a center-of-mass decoding strategy, the estimated horizontal disparity δ_H^{est} is obtained by:

$$\delta_H^{est} = \frac{\sum_{i=1}^{N_p}\sum_{j=1}^{N_o}\frac{\Delta\psi_i}{2\pi k_0\cos\theta_j}r_c^{ij}}{\sum_{i=1}^{N_p}\sum_{j=1}^{N_o}r_c^{ij}} \tag{26}$$

where r_c^{ij} denotes the response of the complex cell characterized by the i-th phase difference and by the j-th orientation. The dashed line plots in Figure 8b-c show the resulting disparity curves obtained by population decoding. The estimate of the disparity can be considered correct when the stimulus disparity is within $\pm\Delta$.

By analyzing the tuning curves of the population (see Figure 8a) we observe that the peak sensitivity of cells that belong to a single orientation channel is uniformly distributed in a range that increases with the orientation angle θ of the receptive field, as the horizontal projection of the frequency of the Gabor function declines to zero. Thus, applying the center of mass decoding strategy, separately for each orientation, we can obtain j different estimates of the disparity:

$$\delta_{H,\theta_j}^{est} = \frac{\sum_{j=1}^{N_p}\frac{\Delta\psi_i}{2\pi k_0\cos\theta_j}r_c^{ij}}{\sum_{i=1}^{N_p}r_c^{ij}} \tag{27}$$

It is worthy to note that the increase of the sensitivity range, as the orientation of the receptive fields deviates from the vertical, comes at the price of a reduced reliability and

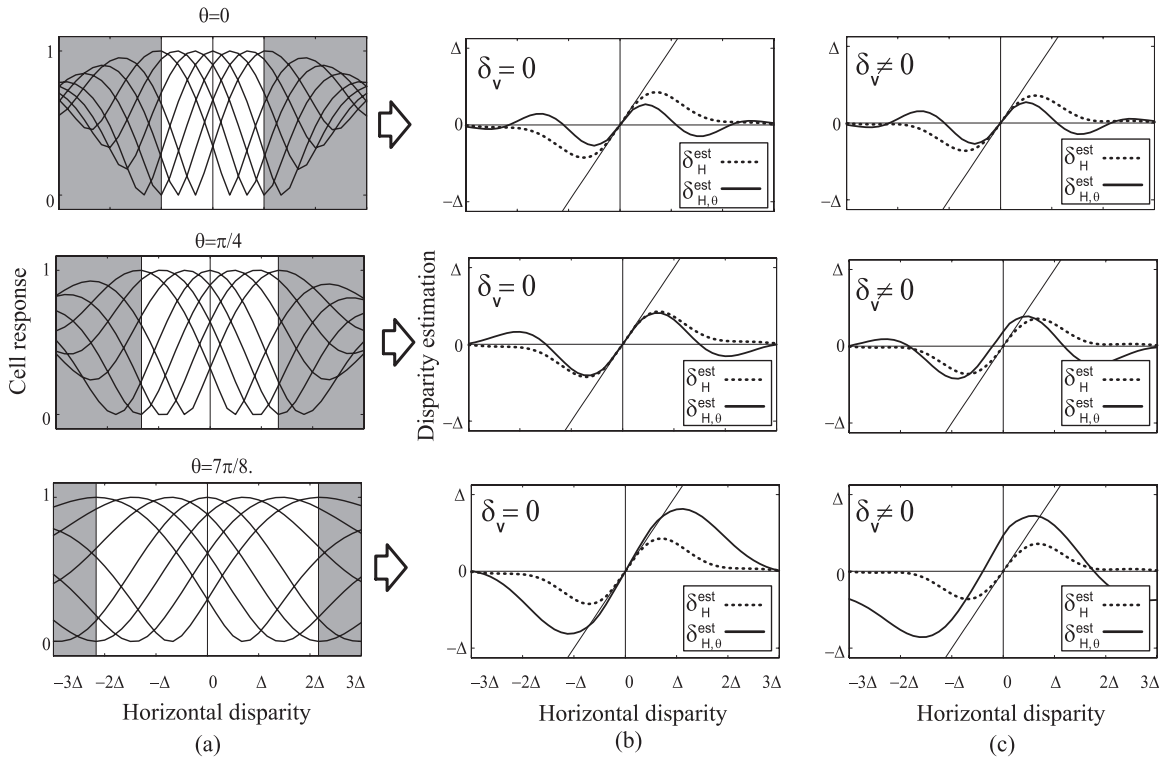


Figure 8: (a) Disparity tuning curves of complex cells at different orientations. (b) Estimated horizontal disparity using single orientation channels in presence of horizontal disparity only ($\delta_V = 0$). (c) Estimated horizontal disparity using single orientation channels in presence of a fixed amount of vertical disparity ($\delta_V \neq 0$). Dashed line plots refer to the horizontal disparity estimates obtained by combining all the orientation channels

accuracy of the measure (as an extreme case, horizontal receptive fields are unable to detect horizontal disparities, *i.e.*, $\delta_H^{\theta=0} \to \infty$). In any case, the estimate of the disparity can be considered correct in a range around $[-\Delta, \Delta]$, only.

Moreover, since the 1D tuning curves of the population were obtained under the assumption of horizontal disparity only, when the vertical disparity in the images differs from zero, the correctness of estimate of the actual component of the horizontal disparity has to be verified. We observe that (see Figure 8b and Figure 8c, top row), the disparity estimated by the whole population is unaffected by non null vertical disparities, as well as the estimate obtained by the orientation $\theta = 0$ (vertically oriented cells are indeed , by definition, sensitive to horizontal disparity only). On the contrary, the estimated disparity obtained for $\theta \neq 0$ shows a dependence on vertical disparity, that increases with θ (see Figure 8c, middle and bottom row), and leads to a systematic error response.

4.1.1 Control signal extraction

A desired feature of disparity-vergence curves is an odd symmetry with a linear segment passing smoothly through zero disparity, which defines critical servo ranges over which

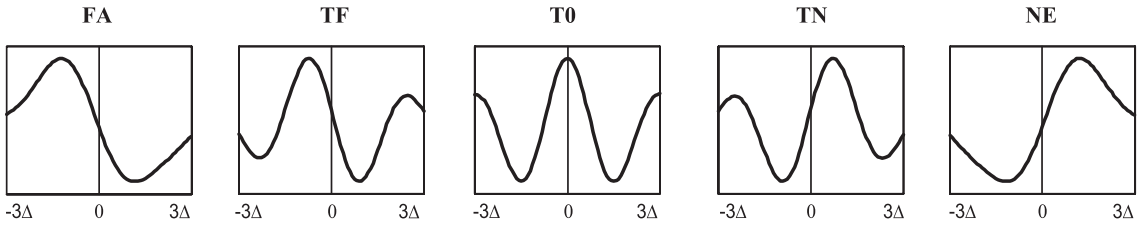


Figure 9: The v_H^k target curves to be approximated by the LMS minimization. Each of them is designed to have a tuning to disparities of differen magnitude.

changes in the stimulus horizontal disparity elicit roughly proportional changes in the amount of horizontal vergence eye movements, $\Delta\alpha = p\delta_H$, where α is the vergence angle. Starting from the estimated disparity curves shown in Figure 8b, we can exploit the responses at different orientations to design linear servos that work outside the reliability range of disparity estimation. Yet, we have to cope with the attendant sensitivity to vertical disparity, which is an undesirable effect that alters the control action. Hence, given a stimulus with horizontal and vertical disparity δ_H and δ_V, we want to combine the population responses in order to extract a vergence control proportional to the δ_H to be reduced, regardless of any possible δ_V. We demonstrate that such disparity vergence response can be approximated by proper weighting of the population cell responses where disparity tuning curves act as basis functions. Due to these considerations, the population responses are combined with two very specific goals: (1) to obtain signals proportional to horizontal disparities, (2) to make these signals be insensitive to the presence of vertical disparities. The disparity vergence response curves r_v^k are obtained by a weighted sum of the complex cell responses (see Figure 10):

$$r_v^k = \sum_{i=1}^{N_p} \sum_{j=1}^{N_o} w_{ij}^k r_c^{ij} \tag{28}$$

where the index k denotes the different kind of the desired vergence response curves. Referring to a common classification [41] we divide the V1 cells in five categories: near (NE) and far (FA) dedicated to coarse stereopsis, and tuned near (TN), tuned far (TF) and tuned zero ($T0$) for fine stereopsis. The weights w_{ij}^k are obtained through a recursive LMS algorithm. From the control point of view, we assume that small values of vertical disparities do not affect the disparity-vergence curves. Moreover, to mildly constraint the solution of the problem and, in the meantime to ensure a good control stability, we pose the VD independence constraint for HD $\simeq$ 0, only. Under this assumption, we can design the disparity-vergence curves that define the visual servos by considering the tuning curves obtained separately for VD=0 and HD=0 (*i.e.*, the orthogonal cross-section of the oriented 2D disparity tuning curves of the binocular energy model). More precisely, the profile of the desired vergence curve v_H^k (see Figure 9) is approximated by a weighted sum of the tuning curves for horizontal disparity $r_c(\delta_H; \theta, \Delta\psi)$.

To gain the insensitivity to vertical disparity we add a constraint term in the minimization formula. This term ensures that the sum of the vertical disparity tuning curves $r^c(\delta_V; \theta, \Delta\psi)$, weighted with the same $\mathbf{w}^k$, approximates v_V^k. To overcome the difficulties

of approximating a constant with a combination of a limited number of periodic basis functions, we impose v_V^k to have a profile that is mildly constant as the one that can be obtained by summing the tuning curves all together ($v_V^k = \sum_{i=1}^{N_p} \sum_{j=1}^{N_o} r_c^{ij}(\delta_V)$). Hence, the weights $\mathbf{w}^k$ are obtained by minimizing the following functional:

$$E(\mathbf{w}^k) = \left\| \sum_{i=1}^{N_p} \sum_{j=1}^{N_o} r_c^{ij}(\delta_H) w_{ij}^k - v_H^k \right\|^2 + \lambda \left\| \sum_{i=1}^{N_p} \sum_{j=1}^{N_o} r_c^{ij}(\delta_V)(w_{ij}^k - 1) \right\|^2 \tag{29}$$

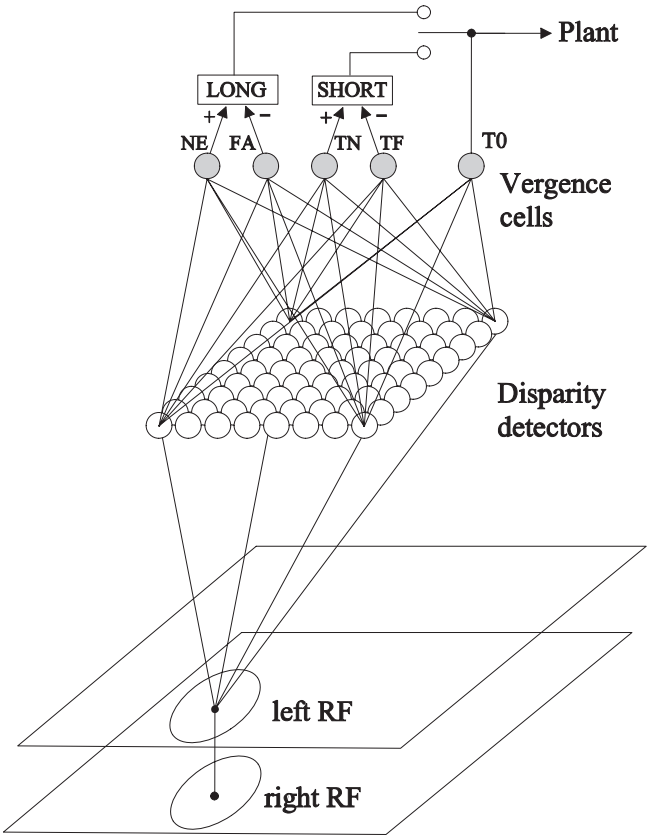
where $\lambda > 0$ balances the relevance of the second term over the first. In our simulations we fixed $\lambda = 1$ in order to give the same relevance to both δ_H and δ_V. To test the functionality of the model, at this stage, we used the same kind of stimuli adopted to compute the disparity tuning curves of the cells, so that we expect the disparity vergence tuning curve to be the same we drew from the minimization. The stimuli have a disparity varying in the same range used for the tuning curves, and the control computed has the same shape of the desired curves (Figure 10b). A drawback that arises is that if the image contrast is lowered, disparity vergence tuning curves hold the same shape, but their gain is consequently lowered, with the effect that the speed of the vergence movements is modulated by the image contrast. The estimated disparity does not show this effect because the center of mass decoding strategy means to take a decision on the disparity value, regardless to the contrast of the stimulus (*cf.* [30]). By analogy with the formula used to decode the disparity, we can introduce the same normalization term to let the system work in the proper way independently of the image contrast:

$$r_v^k = \frac{\sum_{i=1}^{N_p} \sum_{j=1}^{N_o} w_{ij}^k r_c^{ij}}{\sum_{i=1}^{N_p} \sum_{j=1}^{N_o} r_c^{ij}} \tag{30}$$

4.1.2 Signal Choice

With reference to the five categories of the disparity-vergence curves, it is plausible to think that the first two generate the fast and coarse component and the others the slow and fine component of the vergence movements. In practice the fast-coarse control is given by LONG= $r^{NE} - r^{FA}$, while the slow-fine is given by SHORT= $r^{TN} - r^{TF}$ (see Figure 10). The SHORT control signal is designed to proportionally generate, in a small range of disparities, the vergence to be achieved, and allows a precise and stable fixation (Figure 10b). Out of its range of linearity, the SHORT signal decreases and loses efficiency to the point where it changes sign, thus generating a vergence movement opposite to the desired one. On the contrary for small disparities the LONG control signal yields overactive vergence signal that make the system to oscillate, whereas for larger disparities it provides a rapid and effective signal.

The role of the r^{T0} signal, is to act as a switch between the SHORT and the LONG controls. When the binocular disparities are small, r^{T0} is above a proper threshold *TH*, and it enables the SHORT control (see white regions in Figure 10b). On the contrary, for



(a)

Figure 10: Extraction of the vergence control signals: each location of the left and right image is filtered with a population of disparity detectors whose response r_c is combined with five different families of weights w^k, in order to extract five signals $r^{FA}, r^{TF}, r^{T0}, r^{TN}$ and r^{NE}, tuned to disparities of different magnitudes. These signals are combined in a differential way, in order to extract the LONG and SHORT controls, used to drive the vergence eye movements, while the r^{T0} works as a switch between them.

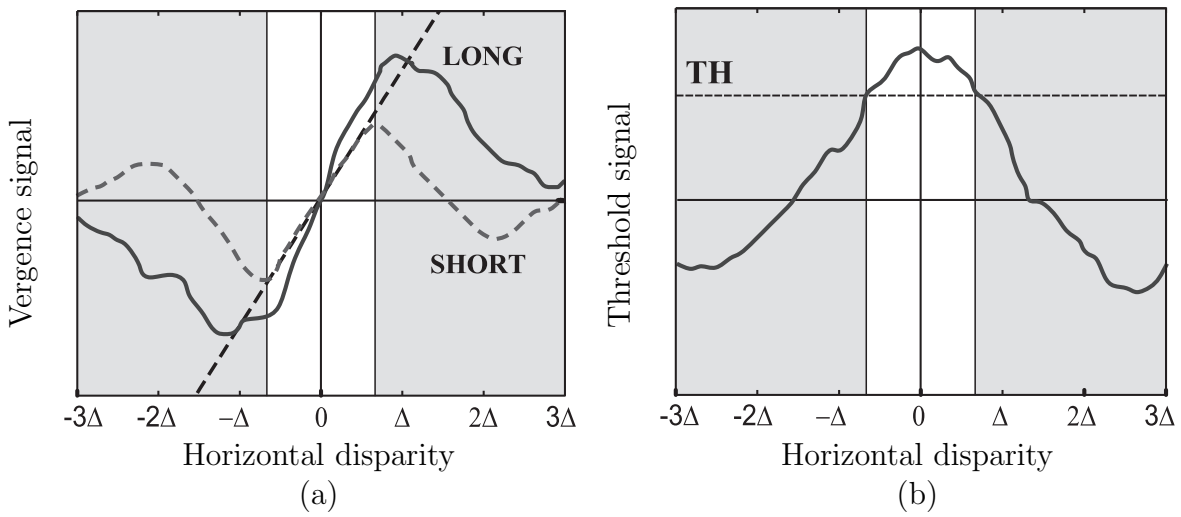


Figure 11: The effective LONG, SHORT (a), and T0 (b) signals computed by the model stimulated with a random dot stereograms (RDS). The SHORT control is able to work in a linear and precise manner for small disparities, while the LONG one works in a coarse but effective way for larger disparities. Since the $T0$ signal is high for small disparities, it is able to act like a switch between the two controls.

large stimulus disparities, r^{T0} is below the threshold and it enables the LONG control (see grey regions in Figure 10b).

A straightforward but meaningful effect that arises from calculating the SHORT and the LONG controls in a differential way is a strong robustness to noise. If we add a Gaussian white noise to the population response, both the decoding of the disparity and the computation of the r_v^k signals, would be affected. Since the weights $\mathbf{w}^k$ are normalized, it is easy to demonstrate that the noise terms on r^{NE} and r^{FA} cancel each other while differentiating to compute the LONG control, and so it happens for the SHORT one. Simulation results evidenced that, when one adopts the differential SHORT and LONG control signals, the S/N ratio is $\sim$ 6dB higher than the input S/N ratio for the complex cell responses.

4.2 Effects of vertical disparity

The optimized control we want to obtain from the proposed technique is a control of the horizontal vergence that would be able to yield the same movement for a given horizontal disparity δ_H, without suffering any effect from the vertical disparity δ_V. Indeed if the δ_V constraint is not taken into account in the minimization process used to obtain the weights w (see Equation 29), the resulting control shows a strong dependence on vertical disparity, as it appears evident in the disparity-vergence tuning curves shown in Figure 8 right column. The control loses the zero crossing and its odd symmetry, which are instrumental features to ensure that at the steady state, the eyes fixate on the closest surface along the axis of fixation, not before, nor beyond.

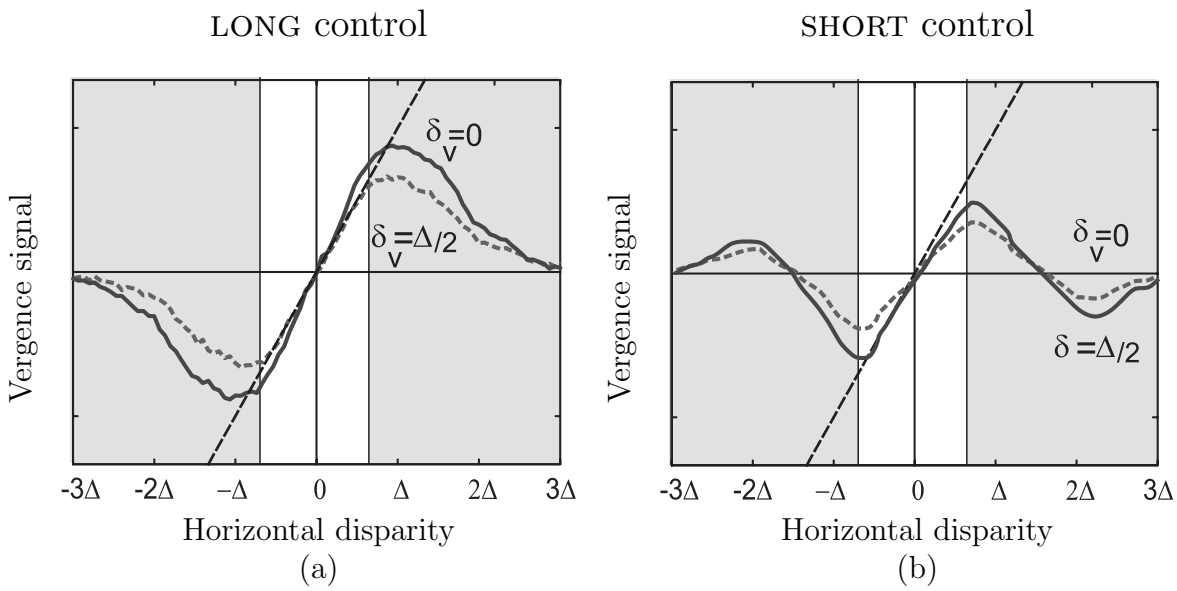


Figure 12: The effective LONG (a), and SHORT (b) signals computed by the model stimulated with a random dot stereograms (RDS) in the absence (solid line) and in presence (dashed line) of a vertical disparity pedestal.

Although, the regularization term we introduced in Equation 29 has the sake of forcing the control to be insensitive to the vertical disparity, simulations with RDSs showed that the behaviour is different from the expected one.

The problem of this approach is due to the fact that the minimization is computed by considering, for every complex cell, its response r_c^{ij} to the horizontal and the vertical disparity only, for the first and the second term of the functional, respectively. As a matter of fact, in the functional in Equation 29, what we consider are the cross-sections for $\delta_H = 0$ and $\delta_V = 0$ of the two-dimensional (2D) tuning profile that characterizes each complex cell. Though, the 2D disparity tuning profile of a binocular energy unit can be oriented by any angle, depending on the orientation channel we consider, and it is separable for $\theta = 0$ and $\theta = \pi/2$ only.

Hence, the problem arises if a vertical disparity pedestal is added to the stimulus, the section of the 2D profile one should consider, is the one at the imposed δ_V. Otherwise, the more the filter is tilted from the vertical and the more the δ_V is, the most the tuning curves change, producing the effect of making the decoding for vergence unreliable.

Thus if the δ_V is set to 0, the control is working in the conditions it is designed for, and its effectiveness is the highest. In Figure 15a we show the evolution in time of the actual horizontal disparity, when the vergence control is active. The value of each plot at the first time step is the initial horizontal disparity step we imposed. The system is able to cope with disparity values ranging from -3Δ to 3Δ and the control of vergence reduces to zero the stimulus disparity. In the figure it is also highlighted on each trace when the system relies upon the SHORT control (filled circles) and the LONG control (open circles). As expected, for small disparities the working mode is the former, while for larger disparities is the latter, depending on the threshold of the $T0$ signal (see Figure 11b). At

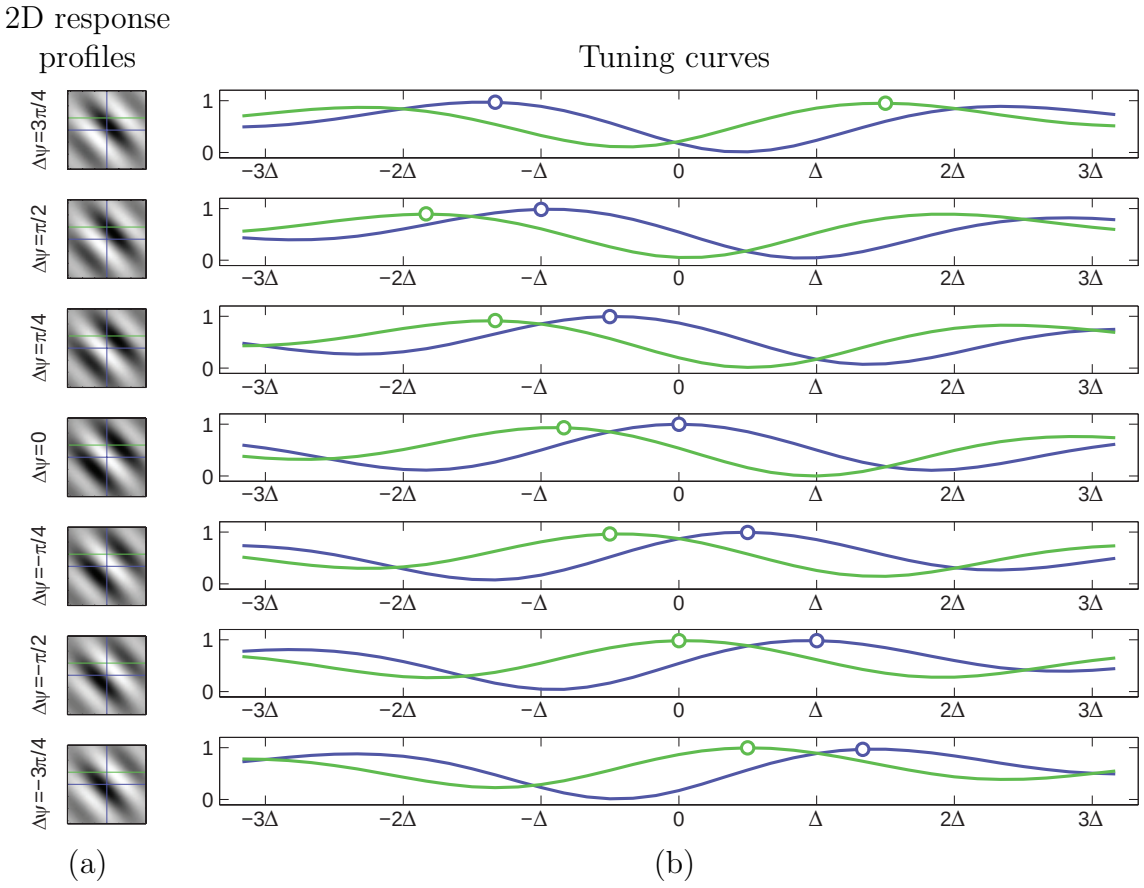
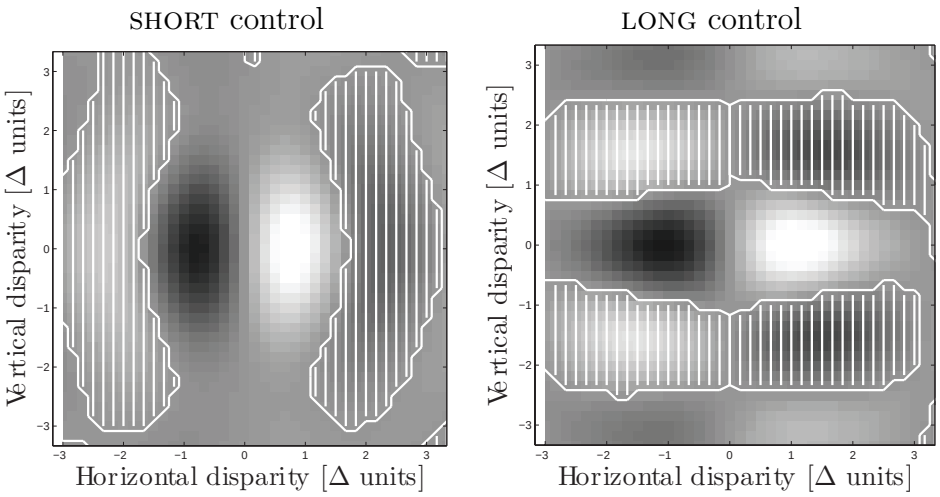


Figure 13: (a) The profile of the response of the complex cell defined by $\theta = \pi/4$, tested with a RDS with δ_H and δ_V ranging from -3Δ and 3Δ. (b) Tuning curves for the same complex cells, taken at different fixed vertical disparity, the blue one is for $\delta_V = 0$ and the green one is for $\delta_V = \Delta$. The empty circles highlight the position of the peak for each curve.



(a) (b)

Figure 14: Response profile of the SHORT and LONG vergence controls to a 2D vector disparity, ranging in the interval $[-\Delta, \Delta]$ for both the horizontal and vertical components. The profile is obtained as a weighted sum of the 2D disparity tuning profiles of the complex cells, using the weights derived by Equation 29 and by Equation 30.

the same way if vertical disparity is small (see Figure 15b), the tuning of the population responses is almost unaffected by δ_V, and the only visible effect is a slight slow down of the vergence control. Increasing the value of δ_V (see Figure 15c-d), besides a more consistent slow down of the control, another drawback is the reduction of the range of δ_H the control is able to cope with. This effect is particularly evident on the LONG mode, because it resort mainly on the cells whose orientation tuning largely deviates from the vertical, thus being more sensitive to δ_V. Figure 14 shows the SHORT and LONG controls obtained as weighted sums of the 2D disparity tuning profiles of the complex cells, and in particular haw the two control are slowed down by δ_V. Moreover the areas where the controls are unreliable is highlighted with white lines.

4.3 Results

4.3.1 Test with Random Dot Stereograms

We tested the proposed model with synthetic stimuli consisting of random dot stereograms (RDS) in which the stereo image pairs are shifted horizontally. Specifically, we applied horizontal disparity steps varying from -3Δ to 3Δ. The model works in a perception-action loop in which the vergence movements are simulated reducing step by step the disparity between the left and right images by an amount proportional to the vergence control, computed both through the estimation of the disparity δ_H^{est} and through the vergence signals r_v^k. Figure 17 shows the percentage of vergence movement accomplished by the two mechanisms for different time steps. Because of the behaviour of the two mechanisms is symmetric with respect to zero disparity, we show the positive semiaxis only. A percentage value higher than 100 indicates an overshoot of the

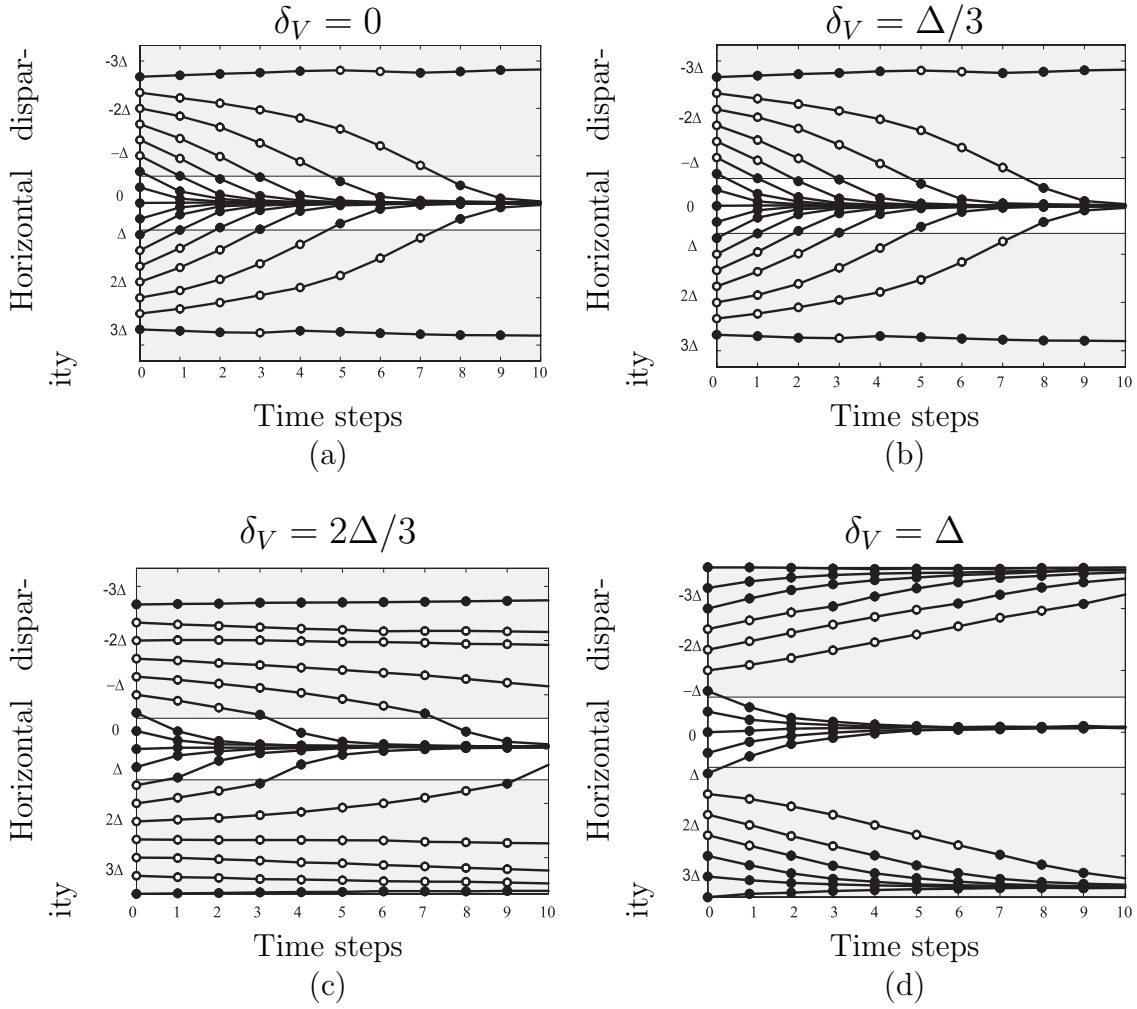


Figure 15: Evolution in time of the vergence control testes with a RDS. Fixed a vertical disparity pedestal, varying from 0 to Δ, each trace represent the evolution of the vergence starting from a different value of horizontal disparity. It is clear that considering a small vertical disparity (b), its effect on the horizontal vergence is negligible. Increasing it above a certain value (c) and (d), the vergence control is slowed down and its range of effectiveness is reduced. Filled and opend circles denote the action of the SHORT and LONG controls, respectively.

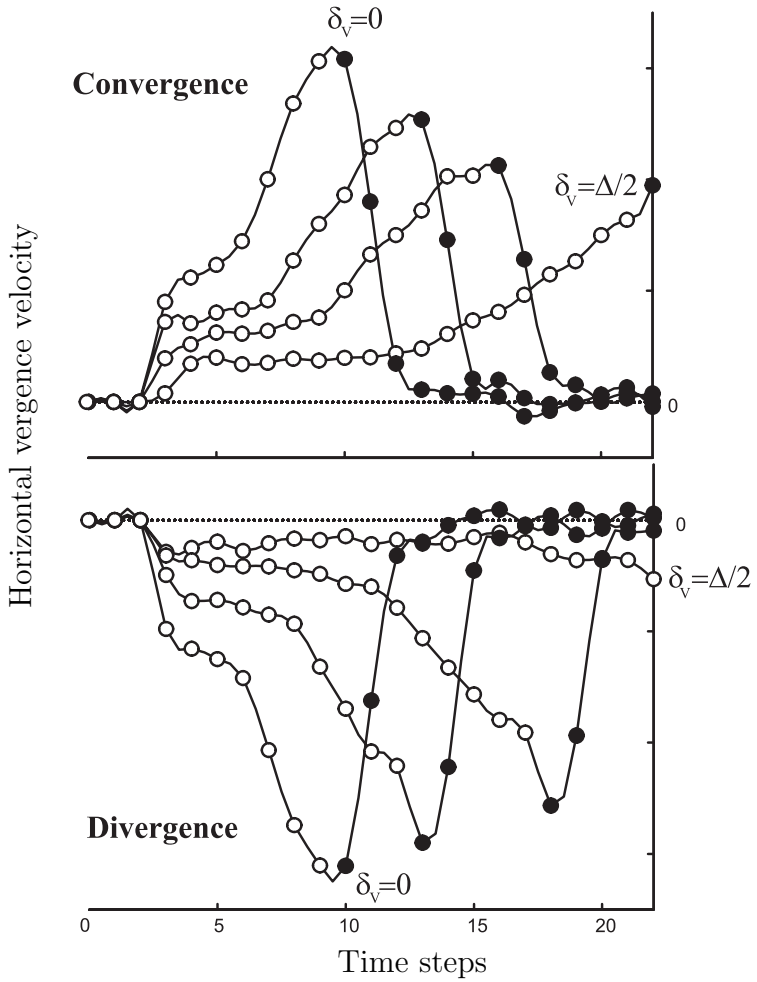


Figure 16: Horizontal vergence velocity (deg/timestep) in presence of a vertical disparity pedestal of increasing magnitude.

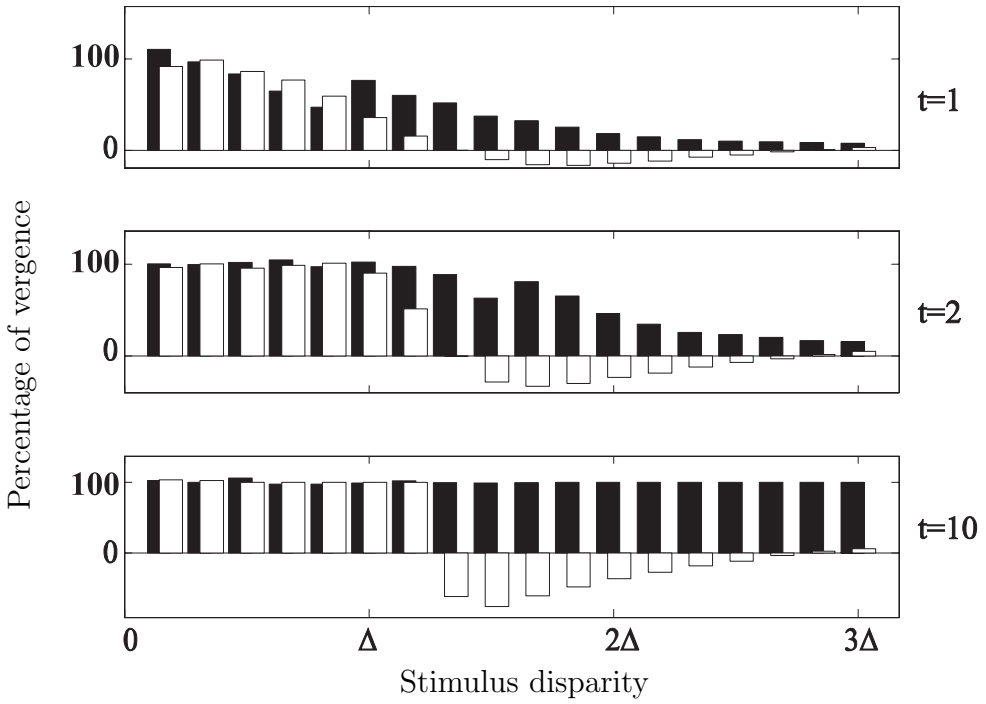
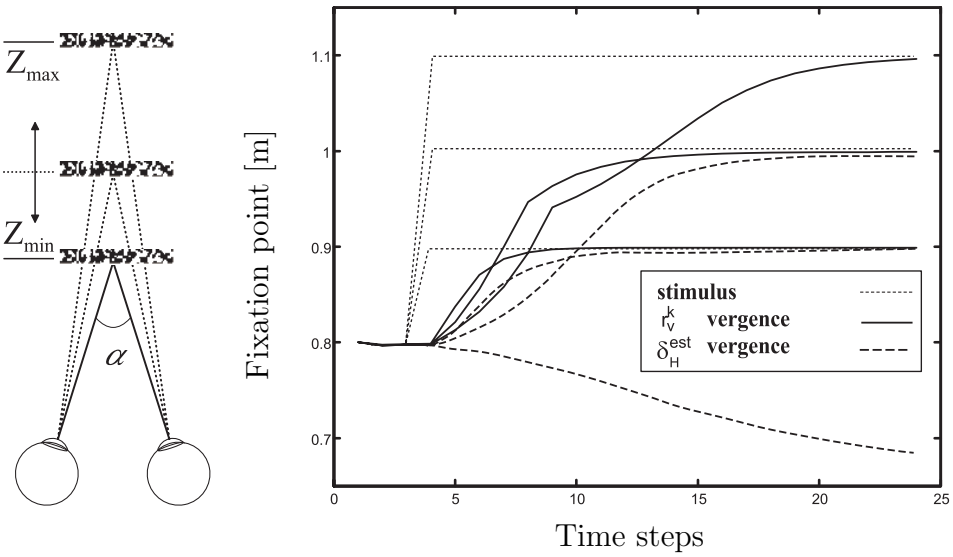


Figure 17: Comparison of percentage of vergence achieved by the model using the estimated disparity δ_H^{est} (white bars), and the r_v^k signals to control the vergence. The stimulus used is a RDS with a disparity step in the range from -3Δ to 3Δ. Only the positive axis is represented because the response is symmetric around zero disparity. The graphs represent the status of the system after 1, 2 and 10 time steps. At each step the r_v^k signals are able to reach the target in the whole range tested, while δ_H^{est} yields a wrong control for disparities larger than Δ.



(a) (b)

Figure 18: (a) Simulated experimental setup, consisting of the eyes looking at a plane characterized by a RDS pattern, and perpendicular to the binocular line of sight. (b) Behaviour of the vergence control using r_v^k vs δ_H^{est} in case of a diverging step. The r_v^k control (solid line) is able to reach the depth of the plane (dotted line) in all the cases presented, while the δ_H^{est} control (dashed line) produce a wrong movement for a depth step above a certain threshold.

movement, whereas a value lower than zero indicates a movement in the opposite (*i.e.*, wrong) direction. After the first time step (Figure 17 top row), if the stimulus disparity is within Δ, the behaviour is slightly better for δ_H^{est} (white bars), whereas outside this range it produces a vergence movement that is the opposite of the one requested. The r_v^k signals (black bars) produce almost the same movement of δ_H^{est} for small disparity steps, but they are able to achieve slow but effective vergence movements up to the limit of the tested range. At the second time step (Figure 17 middle row), for disparity steps smaller than Δ, both the mechanisms reach the target, and for higher disparities the behaviour is similar to the previous time step. After 10 time steps (Figure 17 bottom row), we observed that δ_H^{est} was able to work in the proper way only for disparities within Δ, whereas r_v^k was able to reach the target in all the tested range.

4.3.2 Test with a frontoparallel plane

Considering a virtual environment in which the eyes, characterized by null version and elevation angle, and by a vergence angle α, look at a plane with a random dot texture (Figure 18a). The plane is at a depth Z with respect to the cyclopic position, and perpendicular to the binocular line of sight. The interocular distance is $b = 70mm$, the nodal length is $f_0 = 17mm$, and the stimulus is projected onto the retinal plane, with a size of $6mm$, thus considering a field of view of almost 20 degree. At the first time step the plane and the fixation point are at the same Z, then the plane is moved to a new depth,

and the vergence angle starts to change step by step, until the fixation point reaches the depth of the plane. Considering the position of the eyes, the vergence variation is applied symmetrically: $\Delta\alpha_R = -\Delta\alpha_L = -\arctan(\frac{r}{2f_0})$, where r is computed by considering the weighted average of the vergence responses r_v^k or of the estimated disparities δ_H^{est}. The area where the average is computed, is a neighborhood of the fovea of 5°, and its size is based on physiological experiments [42] that show that it is the maximum extent of the retina where the disparity stimulus is integrated to drive vergence eye movements in humans.

The first test considers a fixed frontoparallel plane at a given distance, while the eyes are fixating on the surface of the plane. The plane steps back and forth by an amount that varies from trial to trial. Figure 18b shows that a control based on the disparity computation $\delta_e st$ produces the correct change of the vergence angle (dotted lines), when the size of the step is restrained. On the other hand, the implemented model is able to produce a faster change of the fixation point (solid lines), and, even for larger depth steps, the model is able to ensure a reliable vergence control. Moreover, once the fixation point has reached the plane in depth, the disparity in the fovea is approximately zero and the system is able to ensure a stable fixation.

The second test considers a frontoparallel plane whose position in depth varies continously in time as a ramp and a sinusoid. The slope of the ramp is varied from 0.5 cm per time step to a pure step (Figure 19a). While for small values only the SHORT control is enabled, in the other cases the initial part of the vergence is produced by the LONG one, and the interplay between the two controls is very similar to the one observed in the transient and sustained components of the physiological responses [43]. In support of this hypothesis, in case of both a divergent and a convergent ramp, the simulated vergence movements are qualitatively very similar to the results obtained in physiological experiments. In the same way, the frequency of the sinusoid that controls the depth of the plane was varied between 7 and 38 time steps, and again the simulated results (Figure 19b) are qualitatively similar to the experimental data [43]. Increasing the frequency, it is evident a transition from a slow and smooth tracking of the plane, due to the SHORT control, to a combination of the LONG and SHORT controls. When the frequency becomes too high, the system is no more able to follow the stimulus in depth. To demonstrate that the vergence control is independent of the image contrast, the first and the second test were repeated with the same RDS texture, but with different levels of contrast. The denominator term in Equation 30 has the effect of a divisive normalization, thus the response of each disparity detector is rescaled with respect to the stimulus contrast (*cf.* [44]). Thereby the vergence control, being derived from a linear summation of the responses of the disparity detectors, is not affected by the stimulus contrast too. It is worth noticing that an RDS image has constant energy in the frequency domain, thus it produces an optimal response of the filters; for a real image it is not guaranteed. The contrast normalization allows us to maintain the vergence control effective even if the texture of the plane is a real image.

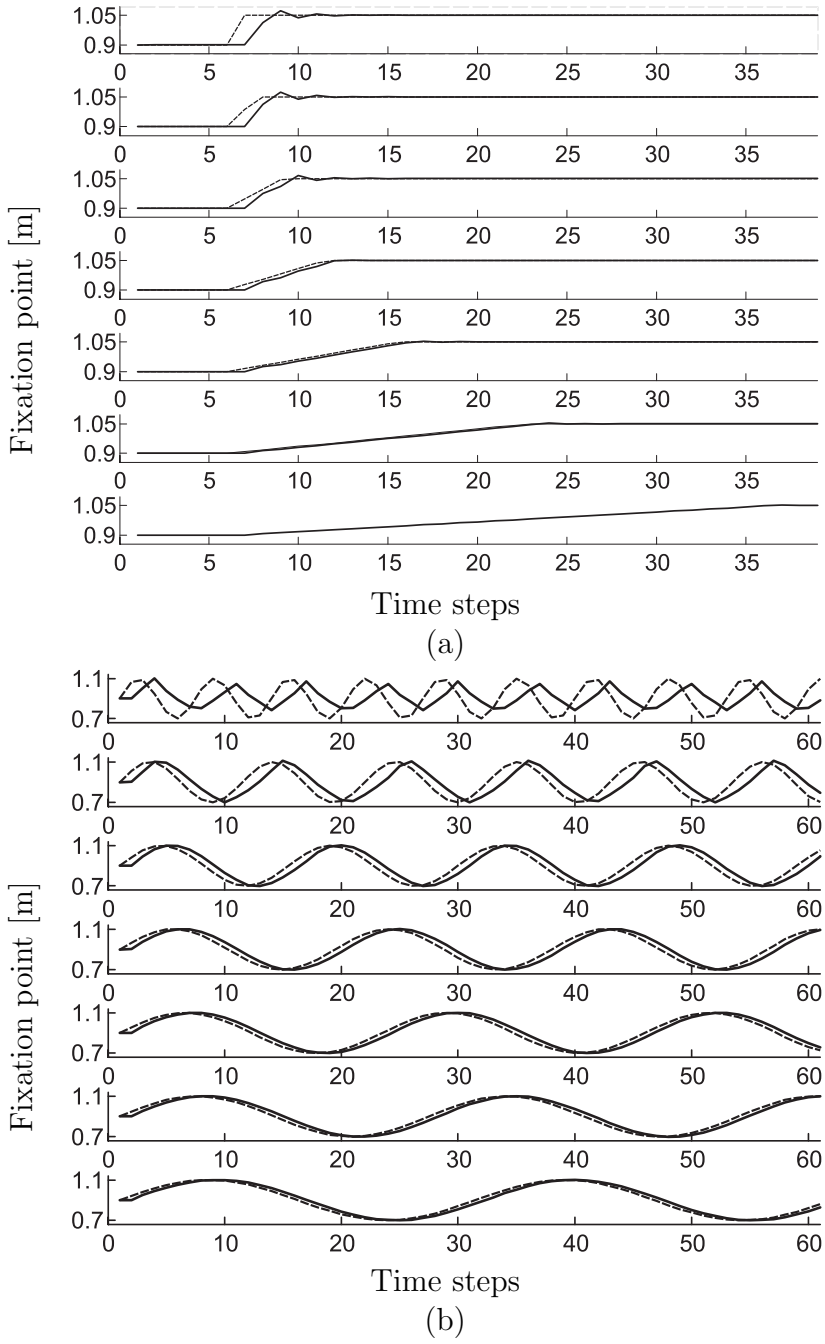


Figure 19: (a) Vergence response in time to diverging ramps with different slopes, and (b) to sinusoids characterized by different periods. The dotted line represents the depth of the stimulus and the solid one is the depth of the fixation point.

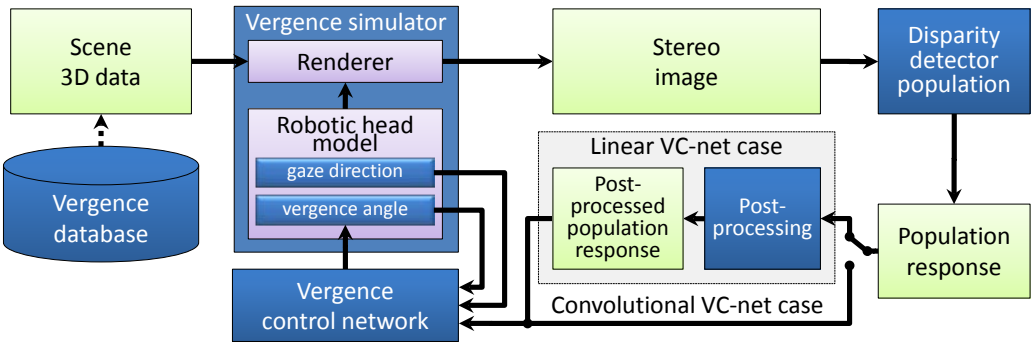


Figure 20: The block diagram of the framework used in vergence control model training and testing (see text).

5 Network Paradigms for vergence control

In this section, we present a modular architecture (see Section 5.1) and two networks for learning vergence control: a linear (Section 5.3.2) and a convolutional one (Section 5.3.3). The training of the proposed vergence control networks is briefly discussed in the same sections (5.3.2 and 5.3.3). In more details we discuss the evaluation experiments (Section 5.5) and the results (Section 5.6).

5.1 Vergence control framework

For the vergence control paradigm modeling, we have used the framework shown in Figure 20. This setup consists of the *vergence simulator* module, the *disparity detector population* module, the *population response post-processing* module and the *vergence control network* (VC-net) module.

The main goal of the vergence simulator is to generate a stereo image (left and right eye views) based on the actual state of the robotic head: the *vergence angle* and the *gaze direction* (see Figure 1), and information about the 3D environment.

The stereo image generated by the simulator is processed by the *disparity detector population*, to produce the population response. Depending on which vergence control network is used, the population response is then directed to either the *population response post-processing* block, which is producing the *post-processed population response* (the linear VC-net case), or directly to the *vergence control network* module (the convolutional VC-net case). The (raw/post-processed) *population response*, together with the actual values of the *gaze direction* and the *vergence angle*, are fed into the *vergence control network* module, the main module of the model. The goal of the VC-net is to produce a new vergence angle, to get the fixation point onto the surface of the object of interest, without changing the gaze direction.

5.1.1 Vergence database

For training the VC-net, we have prepared a *vergence database*. The database consists of two tables: a table of synthetic scenes, and a table of vergence samples (see Figure 21). For efficient memory usage, the scenes were allowed to be reused in several vergence samples. There are two types of synthetic scenes in the vergence database, which correspond to the simplified- and general case scenarios, respectively. The simplified case scenes contain only one type of object-stimulus, a fronto-parallel rectangular patch perpendicular to the gaze direction in the primary position (see Figure 22a). The stimulus in this case is large enough to completely cover the field of view of both cameras.

The general case scenes consist of several simple (plane rectangular patch, cube, pyramid, tetris-like etc.) textured objects, randomly placed into a room-like virtual environment with several light sources (see Figure 22b). The object sizes are chosen randomly allowing for depth discontinuities.

Vergence samples consist of the *gaze direction*, the *actual vergence angle*, the *stereo pair* (left and right eyes' images), the *population response* for the stereo pair and the *desired vergence angle*. The actual vergence angle is a distorted (with Gaussian noise) version of the desired one. The actual vergence angle is expected to become as close to the desired vergence angle as possible, when running the control model.

Each vergence sample in the database can be considered as a training pair. The input part is constructed from the post-processed (or raw) population response, the gaze direction and the actual vergence angle; the output consists of only one scalar parameter – the desired vergence angle. The vergence database used for the VC-net training consists of 1000 synthetic scenes and 5000 samples. The balance between general and simplified scenes (as well as for the samples) has been set to 50/50%. Real-world images were used as textures for the objects. To reduce the influence of a possible overfitting to particular textures on the results of the evaluation, we have used non-overlapping sets of textures for the training- and test experiments. An early stopping technique (with 10% of the training data for validation) was used to prevent overfitting during training. To achieve a fair comparison, both VC-nets were trained using the same training data.

5.1.2 Vergence simulator

The vergence *simulator* module consists of the *renderer* and the ideal *robotic head model* (RHM) with fixed neck. In this model, the robotic head is assumed to be fixed and the eyes to rotate around their nodal points. We selected this model because it is easy to implement, and eventually to replace by a real tilt-pan stereo setup.

Given a RHM baseline b, *i.e.*, the distance between the nodal points, the gaze direction is defined by γ and λ, the pan/yaw and tilt/elvation angles, considering the coordinates centered in the cyclopean point O in the middle of the baseline b, as in [45]. The actual vergence angle α is defined as the angle between the left and right visual axes $\mathbf{v}_l$ and $\mathbf{v}_r$ (see Figure 1). In this study, we used the same parameters of the RHM as in [46] (the baseline $b = 70\,\text{mm}$, the focal length $f_0 = 17\,\text{mm}$ and field of view $\approx 20°$). The RHM, developed by UG-Dist module takes as input the vergence angle and the gaze direction to produce the exact position and orientation of both eyes/cameras, needed for the renderer.

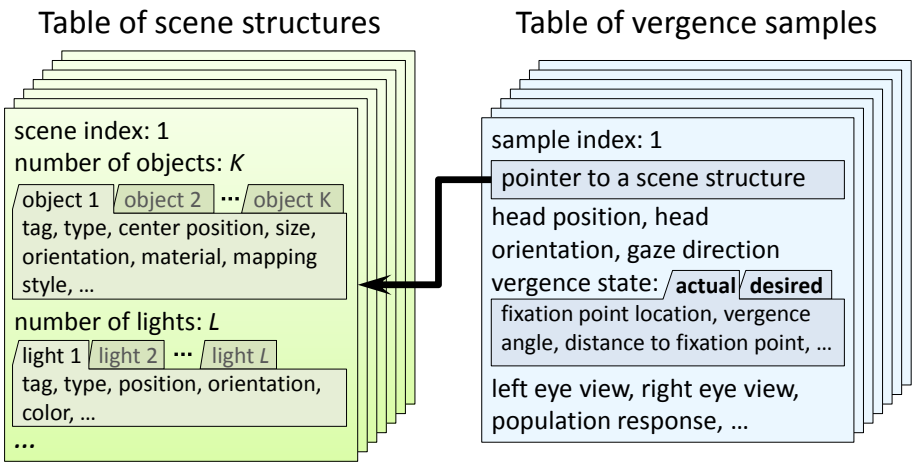


Figure 21: Schematic structure of the vergence database.

The renderer, in turn, produces the stereo image, observed by the left and right eyes (see Figure 22), using the position/orientation of the eyes, and the geometric description of the scene, provided by the *scene 3D data* block.

To make sure that the disparities are not too large and can be properly handled by the disparity detector population, we decided to render the retinal projections with low resolution *i.e.*, we obtain images of 41×41 pixels for a field of view of $\approx 20°$. Note that the resolution could be higher, but consequently to allow the population to cope with the same range of disparities, the receptive fields of the disparity detectors should be larger, which would significantly increase the computational cost and, thus, slow down the simulations.

5.2 Post-processing module

5.2.1 Disparity detectors population module

Disparity information can be extracted from a stereo image pair by using a distributed cortical architecture discussed in great details in Section 3. In this work, we consider only a single-scale architecture of the disparity detector population, but the population can be readily extended to the multiscale mode, without conceptually changing our framework, but which will be computationally much more expensive.

5.3 Post-processing module

ssec:post-processing The post-processing of the population response is used only for the linear VC-net, and comprises a two-dimensional convolution over the first two (spatial) dimensions of the population response, using a two-dimensional Gaussian kernel G_σ:

$$P_{ij} = G_\sigma * r_c^{ij}, \tag{31}$$

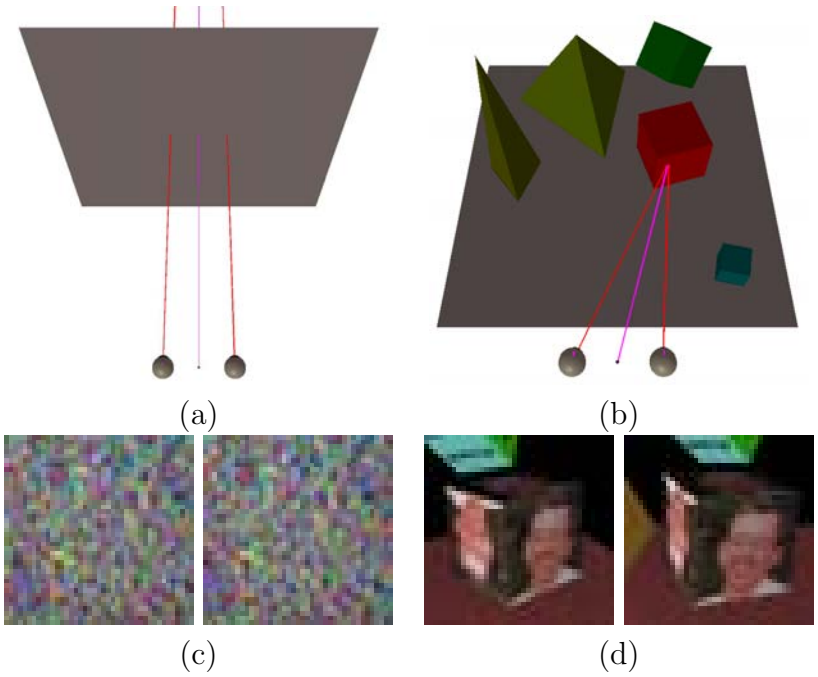


Figure 22: Examples of simplified (a) and general case (b) synthetic scenes used by the simulator to render the corresponding stereo images (c,d).

where r_c^{ij} is the population response map for the i-th orientation and the j-th phase shift. The kernel G_σ has the same size $n_r \times n_c$ as the size of a population response map r_c^{ij}, so the result of the convolution is a scalar value P_{ij}.

On the one hand, this step drastically reduces the amount of data to further process. Indeed, after pooling, the network has to process only a two-dimensional $(N_o \times N_p)$ pooled population response instead of a four-dimensional $(n_r \times n_c \times N_o \times N_p)$ array, where N_p is the number of phase shifts, and N_o the number of orientations. But, on the other hand, the pooling has a major drawback as it discards the spatial information about the disparity encoded in the population response. The results of simulations (see Section 5.6 revealed that, in the general case scenario, this discarding could lead to a degraded vergence accuracy.

The convolutional network works directly on the population response, and the post-processing is done in the first two layers of the convolutional network.

5.3.1 Vergence control module

This module is the main module of the model. The purpose of it is to convert the post-processed population response together with the actual vergence, and the gaze direction, into a new vergence angle. Virtually, this module can be represented by any kind of paradigm, but in this workpackage we discuss only a linear network and a convolutional network.

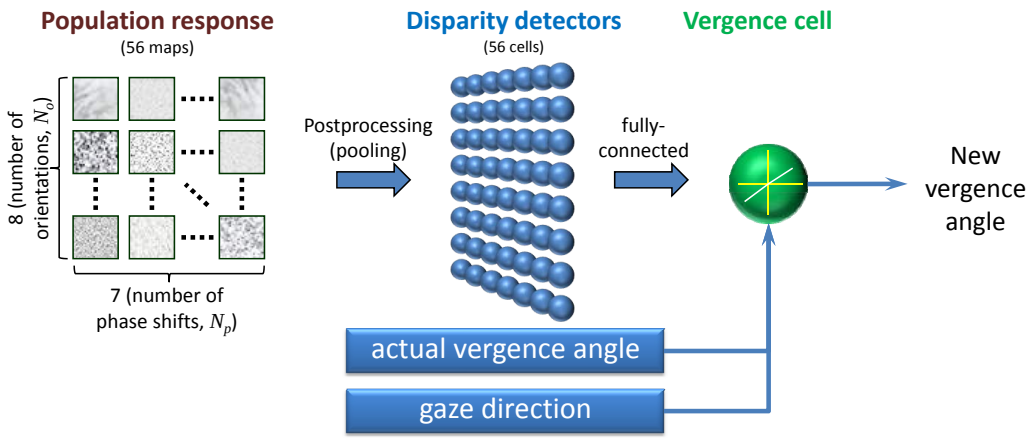


Figure 23: Linear VC-net and its inputs.

5.3.2 Linear network

The first attempt in developing a network model for vergence control was with the simplest possible solution consisting of only a single linear unit (see Figure 23).

The simulations revealed (see Section 5.6) that even this simple network is able to produce accurate angular vergence control in some restricted situations (*e.g.*, in the simplified case). The input vector for the linear VC-net was constructed as a concatenation of the pooled population response (56 values), the gaze direction (2 values) and the actual vergence (1 value), so its dimensionality is 59. The output is a prediction of the vergence angle, which is a scalar value. Due to the linearity of the network, there was no reason to introduce any hidden layers, so the linear VC-net consisted of only one linear unit. This simplest possible vergence control network has only 60 parameters (including bias), which can be learned either directly (using linear regression or its robust modification), or iteratively (using gradient descent), from the training database.

5.3.3 Convolutional network

Convolutional networks (CNs) appeared in the 80s and became popular in Computer Vision [47–49] mainly due to efforts of Yann LeCun and co-workers [50]. All CNs have common architectural features: *local receptive fields*, *shared weights*, and spatial or temporal *subsampling*, which allow them to achieve some degree of shift- and deformation invariance and, at the same time, reduce the number of training parameters.

A typical convolutional network is a feed-forward network of layers of three types: *convolutional* (C-layer), *subsampling* (S-layer) and *fully-connected* (F-layer). The C-layers and S-layers usually come in pairs and are interleaved, and F-layers come at the end (see Figure 24). The output of a C-layer is organized as a set of *feature maps*. Each feature map contains the output of a set of neurons with local receptive fields. All neurons in the feature map share the same weights, so the feature map is responsible for a particular local visual feature, encoded in the weights of these neurons. The computation

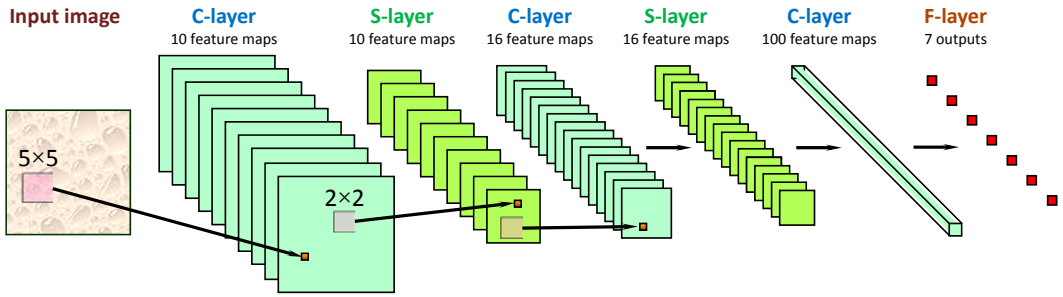


Figure 24: Typical architecture of a convolutional neural network.

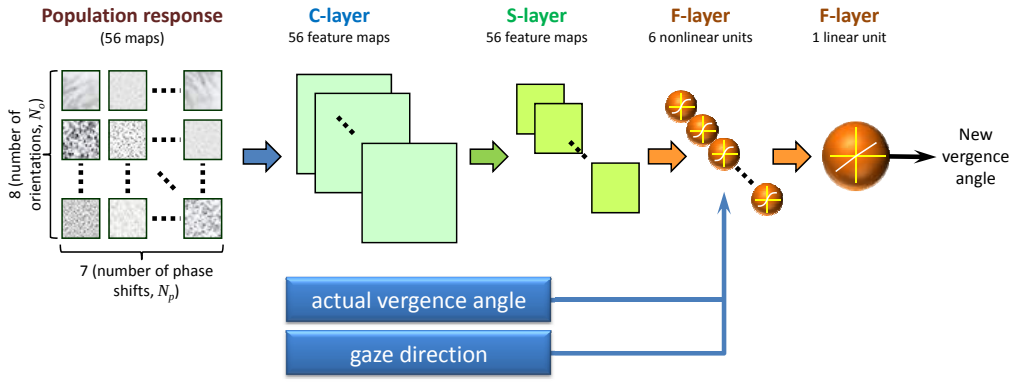
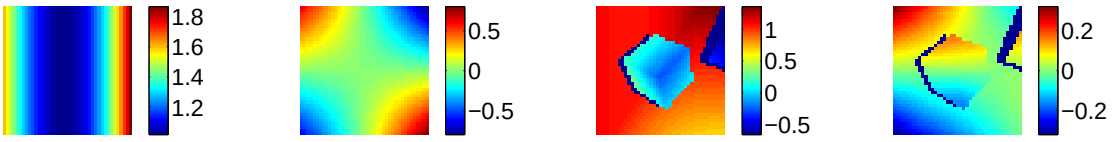


Figure 25: Convolutional VC-net and its inputs.

of a feature map starts with a 2D convolution of the input with a fixed kernel defined by the neuron's weights. A feature map can have inputs from several feature maps of the previous layer. In order to condense the extracted features, and to make them more invariant with respect to spatial deformations, the C-layer is typically followed by an S-layer which performs a local averaging and subsampling. Each neuron in a F-layer just adds a bias to the weighted sum of all inputs and then propagates the result through a nonlinear transfer function (RBF or sigmoid).

The network is trained in a supervised manner using backpropagation. For the efficient training of large CNs, LeCun and colleagues proposed a modification of the Levenberg-Marquardt algorithm [51].

The architecture of the convolutional network, used for our experiments is shown in Figure 25. The main challenge in this approach was the amount of data: the population response consists of 56 (8×7) maps of resolution 41×41 (rendered image resolution), so the input of the network has 94136 ($41 \times 41 \times 8 \times 7$) components. In order to be able to train the network with such high dimensional input data, we had to reduce the number of training parameters. The first (convolutional) layer is a fixed set of (nontrainable) Gaussian kernels of size 19×19 with standard deviation 6. The second (subsampling) layer has also 56 feature maps size of which was set to 3×3.



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

Figure 26: Typical examples of horizontal (a,c) and vertical (b,d) disparity maps for the simplified (a,b) and general case (c,d) synthetic scenes. In the simplified case (a,c), the disparity maps have the same symmetrical patterns, and differ only by the magnitude of the disparity. In the general case (b,d), the disparity maps usually have discontinuities, and are not symmetrical.

5.4 Vergence performance measures

Given the RHM, from the gaze direction vector $\mathbf{g}$, ($\|\mathbf{g}\| = 1$), it is possible to infer the actual distance $d = |OA|$ to the fixation point A from the middle of the head's baseline O using the actual vergence angle α (see Figure 1):

$$\begin{aligned} d &= \frac{b}{2}\left(s + \sqrt{s^2+1}\right), \text{ where} \\ s &= \cot\alpha\cos\gamma \end{aligned} \tag{32}$$

and *vice versa*:

$$\begin{aligned} \alpha &= \arccos\left(\frac{\mathbf{v}_l^T\mathbf{v}_r}{\|\mathbf{v}_l\|\cdot\|\mathbf{v}_r\|}\right), \text{ where} \\ \mathbf{v}_l &= d\cdot\mathbf{g} + (b/2,0,0)^T, \text{ and} \\ \mathbf{v}_r &= d\cdot\mathbf{g} - (b/2,0,0)^T. \end{aligned} \tag{33}$$

where $\mathbf{v}_l$ and $\mathbf{v}_r$ are the visual axes of respectively the left and right eye. From the equations (32) and (33), one can see that, by considering a fixed gaze direction $\mathbf{g}$ and a fixed baseline b, the vergence angle $\alpha \in (0;\pi)$ can be diffeomorphically mapped into the distance to the fixation point d (nevertheless the mapping is nonlinear).

In our experiments $\alpha \in (4°, 10°)$ and, it follows from (32), even for a small vergence angle α (more distant stimulus), the deviation leads to a significant change of d. In this case, the deviation of the actual distance to the fixation point d from the desired one, more adequately reflects the accuracy of the vergence model, than the deviation of the corresponding vergence angles. Due to this anisotropy of the distance uncertainty, we prefer the distance-based measure for the assessment of the model performance over the angular-based.

5.5 Experiments

To evaluate both VC-nets, as already mentioned, we consider two cases for the experiment: a *simplified* and a *general* case. An example of the simplified case is shown in Figure 22(a,c): the gaze direction of the robotic head is orthogonal to its baseline, and the stimulus is in the frontoparallel plane which is also orthogonal to the gaze direction. In the general case, all restrictions on the orientation of the gaze, as well as the stimulus position, type and orientation, are dropped. One of the examples is shown in Figure 22(b,d).

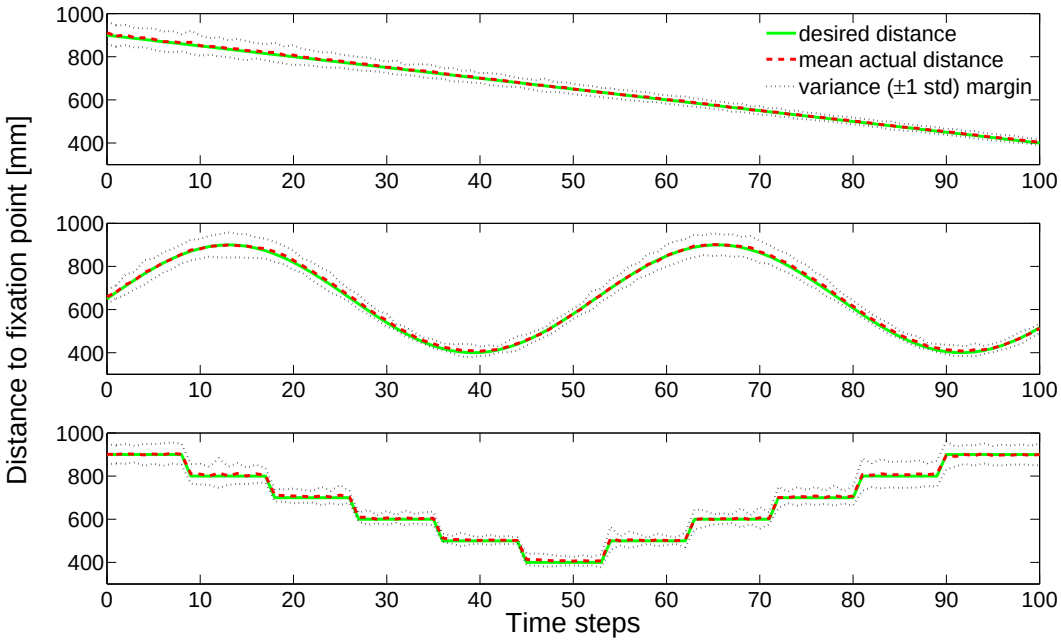
A series of 100 vergence maintenance experiments have been carried out for both VC-nets, for both scenarios. Each experiment consisted of 100 steps during which the randomly generated stimulus was moving along the gaze direction, changing its distance (from 400 mm to 900 mm) to the head in a particular manner. We have considered three patterns of the stimulus motion-in-depth: ramp, sinusoid and staircase. Pretrained VC-nets were allowed to control the actual vergence angle to keep the fixation point as best as possible on the surface of the stimulus. During each experiment, the actual and the desired values of the vergence angle, and the distance to the stimulus were stored for each time step, for further analysis.

5.6 Results

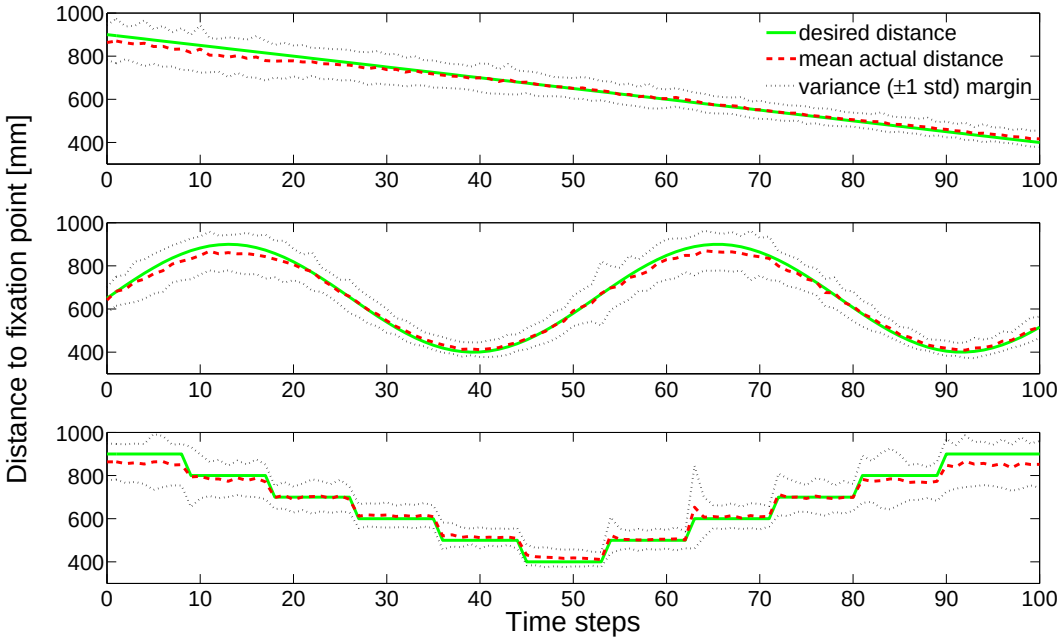
The results of the evaluation experiments described in Section 5.5 of both VC-networks in both considered scenarios are presented in Figure 27, Figure 28 and Table 1. Each panel of Figure 27 and Figure 28 contains: 1) the desired (ground truth) distance to the stimulus curve depicted by the solid green curve, 2) the mean (averaged across all experiments) actual distance to the stimulus curve depicted by the dashed red curve, and 3) the variance (standard deviation across all experiments) of the actual distance margins depicted by the dotted black curve.

The performance of the VC-net can also be assessed using the ratio of the distance-based error variance to the corresponding desired distance. The smaller this ratio is, the lower the relative (distance) error is produced by the network. Table 1 contains the minimal, mean, median and maximal values of this ratio (in percent) for each experiment type and each stimulus.

From Figure 27 and Table 1, it can be clearly seen that both networks perform relatively well in the simplified scenario: the mean actual distance curve almost coincides with the desired one, and the variance in both cases is relatively small. For the general case scenario, the situation is different. The linear VC-net (Figure 27b) shows a much larger variance and a general tendency to over(under)shoot towards the "average" depth of the scene (at approximately 600 mm). The convolutional VC-net (Figure 28b) also shows a relatively larger variance, but the mean actual distance is closer to the ground truth than in the linear VC-net case. The effect of the anisotropy of the distance uncertainty, mentioned in Section 5.4, is noticeable in Figure 27 and Figure 28: the further the stimulus is, the larger mistakes made by the VC-net.

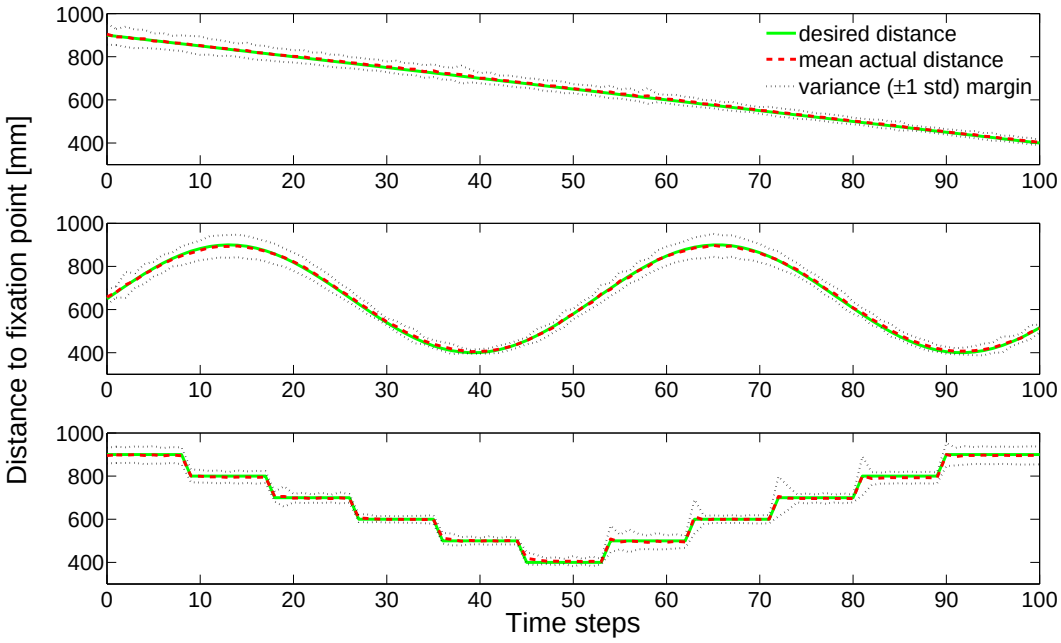


(a) Linear VC-net, simplified scenario.

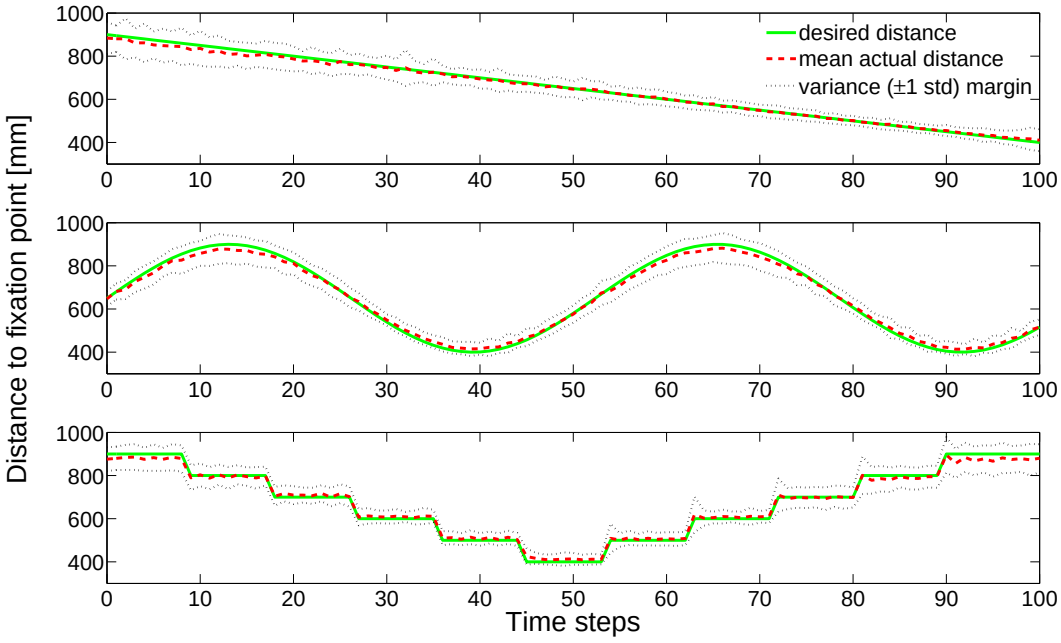


(b) Linear VC-net, general case scenario.

Figure 27: Results of the depth-based performance plots for linear VC-net in both scenarios.



(a) Convolutional VC-net, simplified scenario.



(b) Convolutional VC-net, general case scenario.

Figure 28: Results of the depth-based performance plots for convolutional VC-net in both scenarios.

Table 1: Variance of distance-based error relatively to desired distance.

VC-net	Experiment scenario	Stimulus type	Error variance ratio (%)			
			min	mean	median	max
Linear	Simplified case	Ramp	2.6828	3.8921	3.6172	6.2715
		Sinusoid	2.8904	5.2275	5.2840	7.5772
		Staircase	2.6566	5.4345	5.2589	10.6765
	General case	Ramp	6.4996	8.4466	8.1057	12.9288
		Sinusoid	6.2622	10.0322	9.9448	22.1558
		Staircase	7.1387	10.6848	9.9420	31.9260
Convolutional	Simplified case	Ramp	2.4841	3.7045	3.6237	5.8980
		Sinusoid	2.2913	4.8870	4.9034	8.6034
		Staircase	2.1722	4.3578	3.6502	13.2121
	General case	Ramp	4.0339	6.4378	5.7930	12.7739
		Sinusoid	4.8622	6.9828	6.8680	10.6242
		Staircase	3.9617	6.5304	6.2880	13.7682

5.7 Discussion

The larger magnitude of the vergence error of the linear network Figure 27 in the general case, compared to the simplified case, can be explained, from our point of view, mainly by the disparity discontinuities, and possibly by the presence of the vertical disparity asymmetric patterns. The disparity discontinuities are usually caused by the limited size of the stimuli, which do not entirely cover the field of view in both eyes, or by the non-convex shape of the stimuli (*e.g.*, tetris-like objects). The horizontal and vertical disparities in the simplified case (see Figure 26(a,b)) have very simple symmetrical patterns, while in the general case, these patterns are not so simple and usually not symmetrical (see Figure 26(c,d)). This irregularity of the disparity is caused by the arbitrary orientation and location of the object surface, as well as by the depth discontinuities, and the not always convex shape of the stimulus-object.

For the linear VC-net, in the simplified scenario, the vertical disparity is symmetrically spread over the spatial dimensions of the population response, and is discarded in the preprocessing stage, due to spatial pooling. This does not always happen in the general case, so the pooled population response is biased by the residual vertical disparity, which in turn leads to a bias in the vergence angle, at convergence. This situation motivated us to investigate a more complex paradigm for vergence control, one which should be able to recognize particular patterns in the population responses in the general case, and produce a proper vergence control signal. The idea behind the use of the convolutional network, as a vergence controller, relies on the assumption that this powerful network, after proper training, will be able to recognize disparity patterns directly from the population responses, and convert them into the desired vergence angle.

There is also an interesting phenomenon that we discovered during testing: when the stimulus is too far from the actual fixation point, leading to too large disparities for the population to handle, both networks, in the majority of the cases, choose the *proper*

direction of the vergence control. In this case, the fixation point will not land on the surface of the stimulus-object after the first iteration, but after a few more iterations. This effect can explain the larger error variation for the staircase stimulus (see Figure 27–28, third plot in each panel) compared to the other stimuli: the VC-net is not able to handle large distance steps in one-shot manner and extra iterations are needed. The first iteration, in this case, systematically undershoots, causing a larger distance error, which explains the "spikes" in the error variance curves around the depth steps (in staircase experiments).

We also should discuss some limitations of the proposed models. As our models heavily depend on the quality of the population response, all the limitations of the local distributed disparity methods (poor performance on homogenous textures, short range of the disparity) apply to the proposed models as well. To reduce the effect resorted by these limitations, we suggest to avoid large objects with homogenous textures, and to replace the filters in the population by ones with a larger support, in order to tackle large disparities.

Both proposed models use iterative training based on input images and, therefore, there is a possibility that training will overfit the models on representing the textures used during training. Unfortunately, completely eliminating this possibility, as well as to prove the opposite statement (about the independence from textures), is not possible. One way to avoid this problem, is to consider a large variety in textures in the training set.

6 Vergence-Version Control with Attention Effects (work in progress)

In order to integrate the efforts of WP1, WP2 and WP3 into one model working on one simulation platform (SIMULINK), the partners from T.U.Chemnitz, University of Genoa and K.U.Leuven have developed the *Vergence/Version Control with Attention effects* (VVCA) model. The purpose of the VVCA model is to simulate vergence and version control in the presence of an attention signal.

As it shown in Figure 29, the model consists of:

1. Environment simulator, that generate the image stereo pair.
2. Robotic head model, a kinematic model of the eye movement for a pan-tilt and a tendon-driven binocular head.
3. Disparity representation, a model of area V1 for obtaining a distributed representation of retinal disparity.
4. Object-recognition system (ORS) that generates a saliency map (*FEFmovement*) to drive the version on an object.
5. Eye movement system (EMS) that generates the control signals for the robotic head in order to produce version (based on saliency) and vergence (based on disparity information) eye movements.

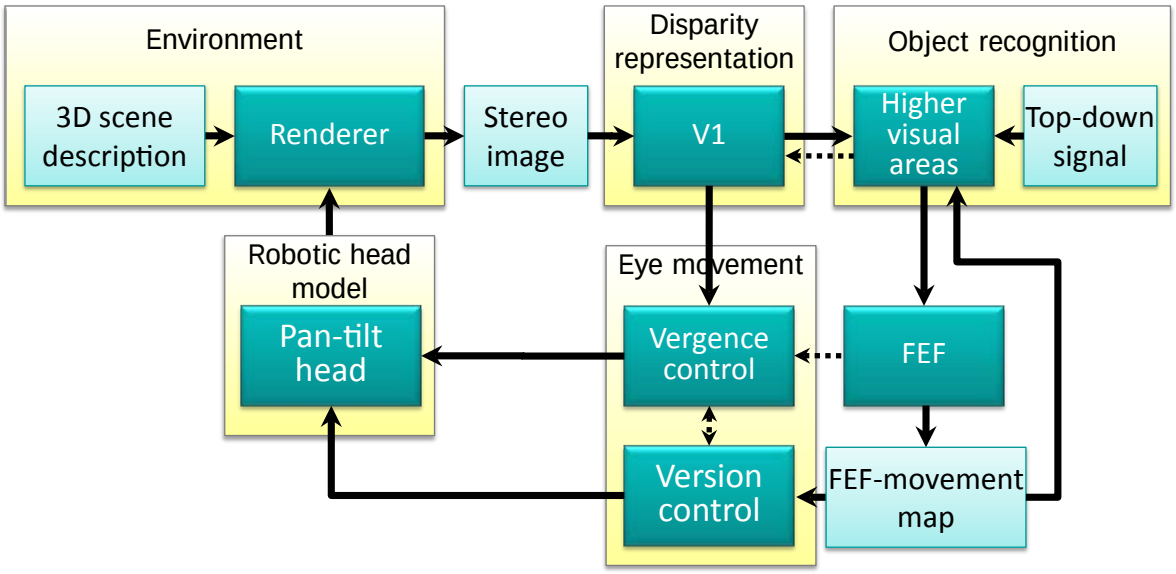


Figure 29: The block-diagram of the proposed VVCA model.

6.1 Image processing workflow

The image processing workflow is schematically shown in Figure 30. The renderer produces a stereo image, with resolution 384×384 pixels, which corresponds to the angular field of view of $40° \times 40°$. Then, the image processing flow is split into two streams:

- **Narrow view** stream: the image is *cropped* to the only central (foveal) part of the input image. This foveal part has a resolution of 192×192 pixels, which approximately corresponds to a $20° \times 20°$ of the original field of view. This stream is involved in the slow (closed loop) stage of vergence.
- **Broad view** stream: the input image is *downsampled/resized* to the resolution of 192×192 pixels, which corresponds to a $40° \times 40°$ field of view (the same as the original image). This stream is used by the ORS, the version control system and the fast (open loop) stage of the vergence control system.

Images processed in such a way are fed into the V1 block, which in turn produces inputs for the EMS and ORS blocks. In this case, the V1 processing technically remains the same for both streams, while the meaning of the results is a bit different. On the one hand, in the *broad* stream, the angle of view is relatively wide, which is beneficial for the planning and execution of the saccadic and/or large vergence movements. On the other hand, in the *narrow* stream, the angular resolution is relatively high, which is important for a fine tuning of the eye orientation during the slow (closed loop) vergence. This double-scale strategy can be considered as the first step towards space-variant visual processing.

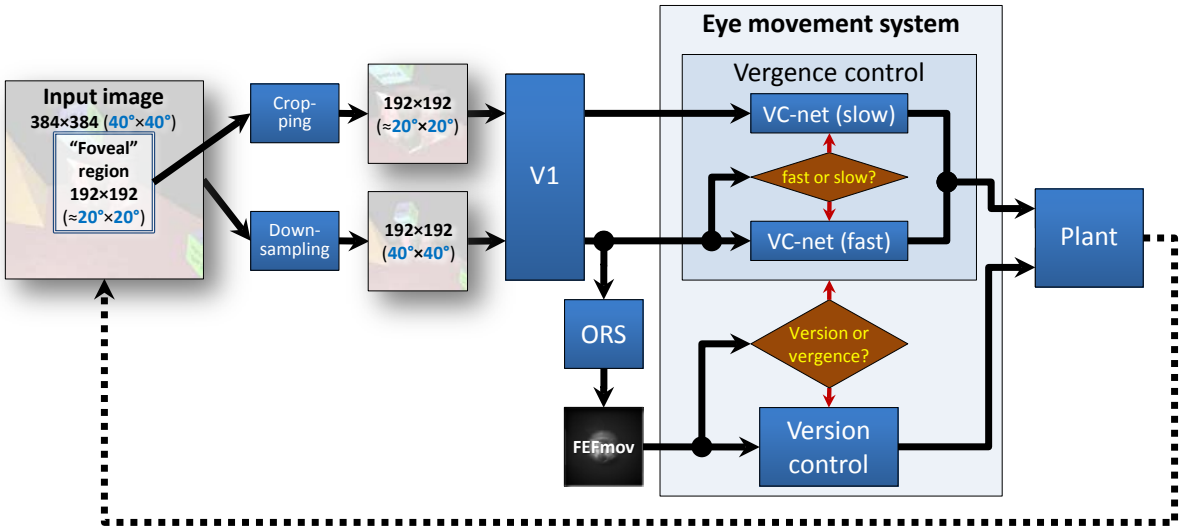


Figure 30: Image processing pipeline scheme in VVCA model.

6.2 Environment

The Environment module consists of a 3D scene description module and a Renderer block that, given the position, orientation and optical characteristics of the cameras, is able to render the image seen by them.

3D scene description block. The 3D scene description block contains the information about the peripersonal space observed by the robot (or its model). Depending on the renderer, this information can be represented in different formats (*i.e.*, MATLAB structure, VRML data). The 3D scenes used are:

- real-world scenes in VRML format, generated from the acquisitions of a 3D laser scanner (UG-Dibe);
- synthetic word scenes, generated by a simulator that considers a peripersonal space populated by synthetic textured objects (cubes, pyramids,...), that can be used both for vergence eye movements and for the object-based saliency (K.U.Leuven);
- vergence test scenes, made up with a simple textured plane that moves along the line of sight at different gaze direction, in order to verify the effectiveness of the vergence control (UG-Dibe).

Renderer block. The main goal of the renderer is to produce stereo images for both eyes according to the actual state of the head (position and orientation of the eyes). The Environment module benefits from the possibility to change the renderer which at the moment is a ray-tracing engine provided by K.U.Leuven, but in future can be replaced by any other renderer, *i.e.*, the stereo-generator being developed by UG-Dibe partner, capable to handle more complex scenarios, such as the 3D laser scanner acquisitions. To

this aim, the C++ virtual reality generator developed by UG-Dibe, has been modified in order to facilitate the intraprocess communication between the C++ and the MATLAB modules. The position of the eyes/cameras generated by the robotic head model can be passed to the renderer block that generate the stereo images and the ground truth disparity maps. Eventually, when the model is tested, the renderer can be replaced by the real stereo camera setup.

6.3 Robotic head model (RHM)

The *Robotic Head Model* (RHM), developed by UG-Dist, takes as input rotational velocities for both eyes and provides the exact position and orientation of the both cameras (eyes) to the renderer.

The robotic head is composed of a bio-inspired ocular model and a pan-tilt platform commonly used in robot vision. The oculomotor plant is composed of:

- Head block;
- Eye block;
- Extra Ocular Muscles (EOMs) block.

The pan tilt system is composed of:

- Head block;
- Pan-tilt block;
- Joint velocities block.

For each block, a custom graphical interface (with the MATLAB GUI) has been developed to configure the parameters of the block.

Head block. The Head block models the human head. Here, we assume that the head is fixed with respect to the reference frame called world. The head is modeled like a rigid body regardless the mass, the dimensions and the inertia of the body. The outputs of this block are:

- The position of the head with respect to the world frame;
- The rotation matrix of the head with respect to the world frame;
- The left and right eye position and orientation with respect to the head frame.

With the graphical interface the user can configure a set of parameters:

- The initial orientation and position of the head with respect to the world frame;
- The initial position and orientation of the two cameras (eye or pan-tilt) with respect to the head frame (fixation point, camera angles, vergence and version elevation angles).

Pan-tilt block. The pan-tilt block, models a pan-tilt system, which can be represented as a kinematic chain with two degrees of freedom. This system is composed of a revolute joint with one rotational degree of freedom about the x axis (tilt joint), a second revolute joint with one rotational degree of freedom about the y axis (pan joint) and the end-effector. For each joint and for the end-effector, a frame that identifies the position and orientation of the joints and of the end-effector in the space is defined. In the case of an ideal pan-tilt system the joints and the end-effector are coincident and there is no translational movement of the end-effector, for a given rotation, with respect to the tilt joint. Conversely, in the case of a real pan-tilt system, the end-effector has a translational movement with respect to the tilt joint. This block takes as inputs velocities for the two joints, the rotation matrix of the head with respect to the world reference frame, and the position vector of the head with respect to the world reference frame. The outputs of this block are the position and orientation matrix of the end-effector with respect to the world reference frame and the Jacobian matrix of the pan tilt system. The parameters of this block are the initial positions of the pan joint and of the end effector.

Joint velocities block. This block is used to compute the SVD (Singular Value Decomposition) of the pan-tilt kinematic chain. Thus, from the Jacobian matrix and from the desired angular velocities of the end-effector the joint velocities are computed for an ideal pan-tilt system.

Oculomotor plant The oculomotor plant is composed of the head, the two eyeballs and the extra ocular muscles that drive each eye in a particular position. The Head block is the same as the one used in the pan-tilt system. The Eye block models the human eye and takes as inputs the four recti muscle forces and the output is the orientation matrix of the eye with respect to the head reference frame. With the graphical interface it is possible to configure the mechanical (inertia, mass, elasticity and viscosity) and the geometrical (muscle's insertion point, soft pulley's positions) parameters of the eye. The EOMs block models the four recti-muscles. Each muscle is modeled as a parallel combination of an active state tension generator, an elastic element and a viscosity element connected to a series elastic element. This block takes in inputs the neurological control signals and the outputs are the four muscle forces. Through the graphical interface the user can configure the mechanical parameters of the four recti muscles.

6.4 Disparity representation (V1)

Primary Visual Area (V1). In the V1 block the disparity is represented by the response of a population of binocular energy complex cells, and is used both for the object-based saliency and for vergence eye movements. The population of disparity detectors is created by Gabor filters characterized by a spatial frequency that defines the range of detectable disparity, the orientation of the filter, which defines the orientation of the disparity, and the phase shift between the left and right filters, which defines the specific disparity tuning of each cell along its orientation. These parameters can be changed to adapt the population to the purpose to be achieved.

6.5 Object Recognition System (ORS)

The main function of the ORS (in the context of VVCA) is to process stereo images produced by the renderer (or by the cameras) and compute the position of the object of interest (in the form of saliency maps), which then can be used for saccade planning. The detailed description of the ORS can be found in Deliverable 3.2.

6.6 Eye Movement System (EMS)

The *Eye Movement System* (EMS) part consists of two subsystems that produce the kinematic (*i.e.*, in terms of rotation velocity) control for version and for the vergence eye movements.

The work of the EMS can be split into several stages:

1. Scene (re)analysis.
2. Version control.
3. Vergence control.
   (a) Fast (open loop) vergence.
   (b) Slow (closed loop) vergence.

6.6.1 Scene analysis stage

In this stage, the EMS waits for the input from the ORS in terms of the FEFmovement map. As soon as the FEFmovement map indicates a target position, the EMS estimates the vertical (δ_v) and the horizontal (δ_h) angular dislocations of the object of interest (represented in the FEFmovement map as a blob) from the center of the binocular view. If $|\delta_v| > \Theta_v$ or $|\delta_h| > \Theta_h$ (where Θ_v and Θ_h are preestimated thresholds), then the actual gaze direction is too far from the object of interest, therefore, a decision about a saccade is made and the EMS proceeds to the version stage. Otherwise, the object of interest is positioned approximately in the center of the (binocular) view, and therefore only the vergence is needed to refine the fixation, the EMS proceeds to the vergence stages starting from the fast vergence stage.

6.6.2 Version control stage

In this stage, the version control subsystem plans and executes (in an open loop manner) a saccade towards the object of interest.

In the VVCA version control subsystem, we assume that the rotation speed dynamics during the saccade is encoded by a special neural mechanisms *saccade related burst neurons* (SRBNs), which are triggered by the initial vertical (δ_v) and horizontal (δ_h) angular dislocations of the object of interest, estimated in the previous stage (see Section 6.6.1). Depending on the amplitudes and signs of δ_v and δ_h, the SRBNs generate rotational speed profiles, which, in turn, are further executed by the oculomotor system. In the

VVCA, we consider a pan-tilt setup, and thus at least three SRBNs are needed for the saccade planning:

- Left eye pan SRBN;
- Right eye pan SRBN and
- Common tilt SRBN.

All of these SRBNs produce speed profiles of approximately equal duration and, therefore, must have some interactions during the saccade planning phase. These interactions are implemented in the so-called *stretching mechanism*, which introduces an appropriate rescaling (in the amplitude as well as in the time domain) of the speed profiles in order to ensure that the saccade will land as close as possible on the planned location and with a trajectory that will not be too curved.

The saccade is considered to be finished when the rotational speed of each eye drops below a threshold $\Theta_\omega \approx 1\,^\circ\,\mathrm{s}^{-1}$. After the saccade is finished, the EMS proceeds to the first stage to re-analyze the scene and to decide whether a second (corrective) saccade is needed.

6.6.3 Vergence control stage

This stage is reached when the object of interest is located approximately in the center of the binocular view, but the fixation point could still be too far from the surface of the object. In this case, the disparity detector population response (broad stream) is analyzed to make a decision which type of vergence is needed: fast or slow? The fast vergence is selected if the pooled energy of the (broad stream) of V1 output is above the energy threshold (Θ_E), otherwise EMS proceeds with the slow vergence stage.

Fast vergence. It is commonly believed that the vergence system switches to the fast mode when it is stimulated by a relatively fast moving stimulus or by a stimulus with a large target disparity. Like in the case of saccades, during the fast vergence phase, the eye rotation speed is still too high to involve the visual feedback into the vergence control. That is why we use a similar (to version) approach to model the vergence related rotation speed, by also employing *vergence related burst neuron* (VRBN). VRBN is triggered by the initial vergence error estimated by the VC-network, which works on top of the V1 broad view stream response.

Depending on the amplitude and sign of the input, the VRBN generates a vergence speed (ω_α) signal which, in turn, is translated into the left and right eye rotation speed signals ($\omega_{LE/h}$, $\omega_{RE/h}$), and which executed further by the oculomotor system. Two possible mechanisms of translating ω_α into ($\omega_{LE/h}$, $\omega_{RE/h}$) are discussed in Section 6.6.4. A few examples of the vergence velocity profiles generated by the proposed VRBN are shown in Figure 31.

After the fast vergence stage is finished (if it applies), the execution control resorts to the slow vergence stage.

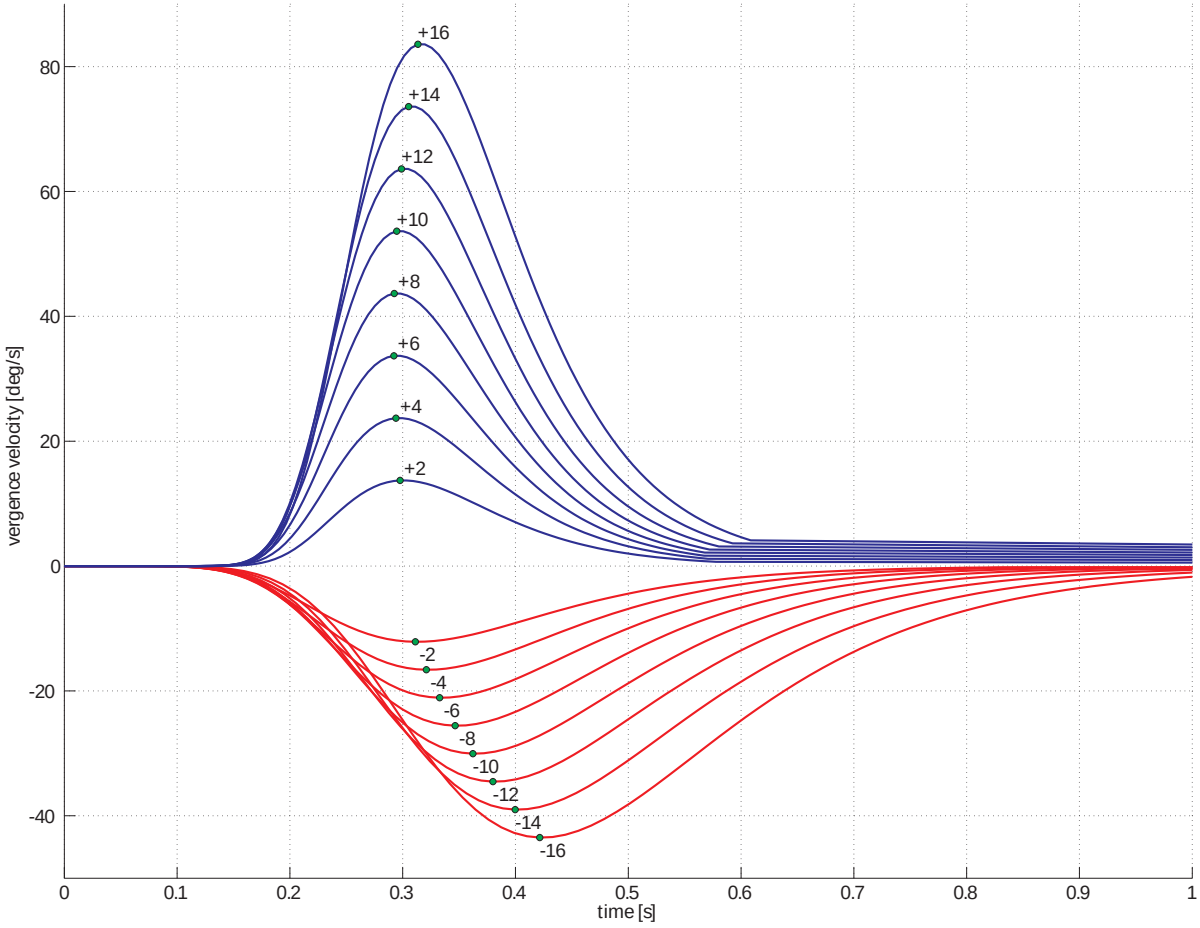


Figure 31: Examples of VVCA fast vergence velocity profiles generated by VRBN. Each profile is generated as a response on the input vergence error (depicted near the velocity peaks and measured in degrees). The vergence velocity peaks are indicated by green bullets on each curve. Convergence profiles are shown in blue, divergence are shown in red. For the sake of clarity, the velocity evolution is shown for a time span from 0 to 1 s, but for the large input vergence errors, the generated velocity profile could be up to 3 s long.

Slow vergence. Relying on the fact that the slow vergence velocity highly correlates with the vergence error [52], we model the slow vergence response as a linearly scaled vergence error: $\omega_\alpha = k(\delta - \alpha)$, where the scaling coefficient k is estimated based on the experimental data from [52].

For the vergence error estimation, we use a vergence control network similar to the one of the fast vergence stage. The difference between these two VC-nets is that the fast VC-net operates on the broad view stream, and the slow VC-net uses the narrow view stream (see Section 6.1 for details). This strategy allows the VVCA vergence subsystem to tackle large disparities for the fast vergence and to be more precise in the slow vergence stage. Due to the well established (and fixed) interfaces with the other modules (V1, RHM), the slow VC-net can easily be represented by any of the full feedback vergence networks discussed in Section 4 and Section 5.

6.6.4 Parameterization of the binocular gaze direction

The definitions of gaze direction and vergence angle depend on the parameterization of binocular eye movements, thus the vergence control signal depends on the choice of the parameterization, too. If we define the cyclopean coordinate center O in the middle point of the baseline, the gaze direction $\mathbf{g}$, ($\|\mathbf{g}\| = 1$) is described by the two angles of azimuth γ and elevation λ, and the left and right visual directions can be parameterized by the gaze direction and the distance of the fixation point (see Equation 33), or by the vergence angle α. Since a pure vergence movement is defined to keep the gaze direction fixed ($\gamma = \text{const}$ and $\lambda = \text{const}$), the left $p_{LE/h}$ and right $p_{RE/h}$ pan rotation angles, for a pure vergence movement, are:

$$\begin{aligned} p_{LE/h} &= \arctan\left(\frac{\sin\alpha + k\sin\gamma}{k\cos\gamma}\right), \\ p_{RE/h} &= p_{LE/h} - \alpha, \text{ where} \\ k &= \cos\alpha\cos\gamma + \sqrt{1 - (\cos\alpha\sin\gamma)^2} \end{aligned} \tag{34}$$

The vergence velocity control can be derived from (34) by a differentiation with respect to time:

$$\begin{aligned} \omega_{LE/h} &= \frac{d}{dt}p_{LE/h}, \\ \omega_{RE/h} &= \frac{d}{dt}p_{RE/h}. \end{aligned} \tag{35}$$

We can observe that the vergence control given in Equation 34 and Equation 35 become symmetric ($\omega_{LE/h} = -\omega_{RE/h} = \alpha/2$) if the gaze direction is orthogonal to the baseline and, thus, $\gamma = 0$ (*e.g.*, in the simplified scenario described in Section 5.5).

On the other hand, if we define the binocular azimuth angle as the average gaze azimuth $\gamma = \frac{1}{2}(p_{LE/h} + p_{RE/h})$ [45], we can accordingly define the locus of points at the same vergence angle ($\alpha = p_{LE/h} - p_{RE/h}$) as the Vieth-Müller circle passing through the optical centers L and R and the fixation point A as in Figure 1. In this case, the cyclopean

coordinate center O lies at the back of the Vieth-Müller circle. This parameterization have the advantage of allowing a *linear* mapping of the horizontal retinal disparity into the vergence control, which can be applied symmetrically, for any azimuth angle γ.

Considering the cyclopean coordinate center at the back of the Vieth-Müller circle, means that the origin of the gaze direction vector changes with α, so that the fixation point A is not exactly the same point in space, but the difference between the two control strategies for the different parameterizations decreases rapidly with the vergence angle.

We decided to adopt the parameterization in terms of version and vergence angles, because these variables are convenient to define conjunctive eye movements from pure gaze shifts at a fixed vergence, and disjunctive eye movements for pure vergence at a fixed version, without requiring information about the actual azimuth angle, γ.

6.6.5 Disparity-vergence analysis

The characteristics of the proposed vergence control subsystem can be presented in the form of a so-called *main sequence*. We use the most common main sequence *peak vergence velocity* versus the *initial vergence error* for the disparity-vergence analysis of the vergence subsystem in VVCA. The initial vergence error (or *target disparity* in some sources) is defined as the difference between the desired vergence angle and the actual one, or $(\delta - \alpha)$ using the notation of Figure 1, thus, the positive vergence error corresponds to the convergence and the negative one corresponds to divergence. The framework proposed in Section 5.1 was ported to the SIMULINK environment, and adapted to the VVCA model, for the observation and/or control of the necessary parameters during the experiments. The simplified scenario (the gaze is directed straight ahead, the 3D scene consists of a fronto-parallel plane) was chosen for the considered analysis. In this case there are only two parameters to control: the distance to the stimulus and the distance to the fixation point. Randomly selecting these two parameters from the range between 300 mm and 1000 mm we obtain the initial vergence error and the corresponding peak velocity. If the initial vergence error is low enough for the fast vergence stage to be skipped (EMS starts directly from the slow stage), the initial vergence velocity is treated as the peak vergence velocity. The results of the disparity-vergence analysis are presented in Figure 32.

7 Conclusions

Binocular energy units are now consolidated models of complex cells in area V1 as demonstrated by numerous recent works that propose architectural variants to enrich their functionality, or that adopt them to describe complex perceptual behaviors. Similarly, in the biologically inspired computer vision literature there exist several examples of neuromorphic (*i.e.*, distributed) approaches that are successfully used for challenging conventional solutions to computer vision problems, by introducing sophisticated interpretations of biologically plausible operations.

Most of the conventional vergence control models [52–60] are based on the minimization of the horizontal disparity. Although the performances of these models were

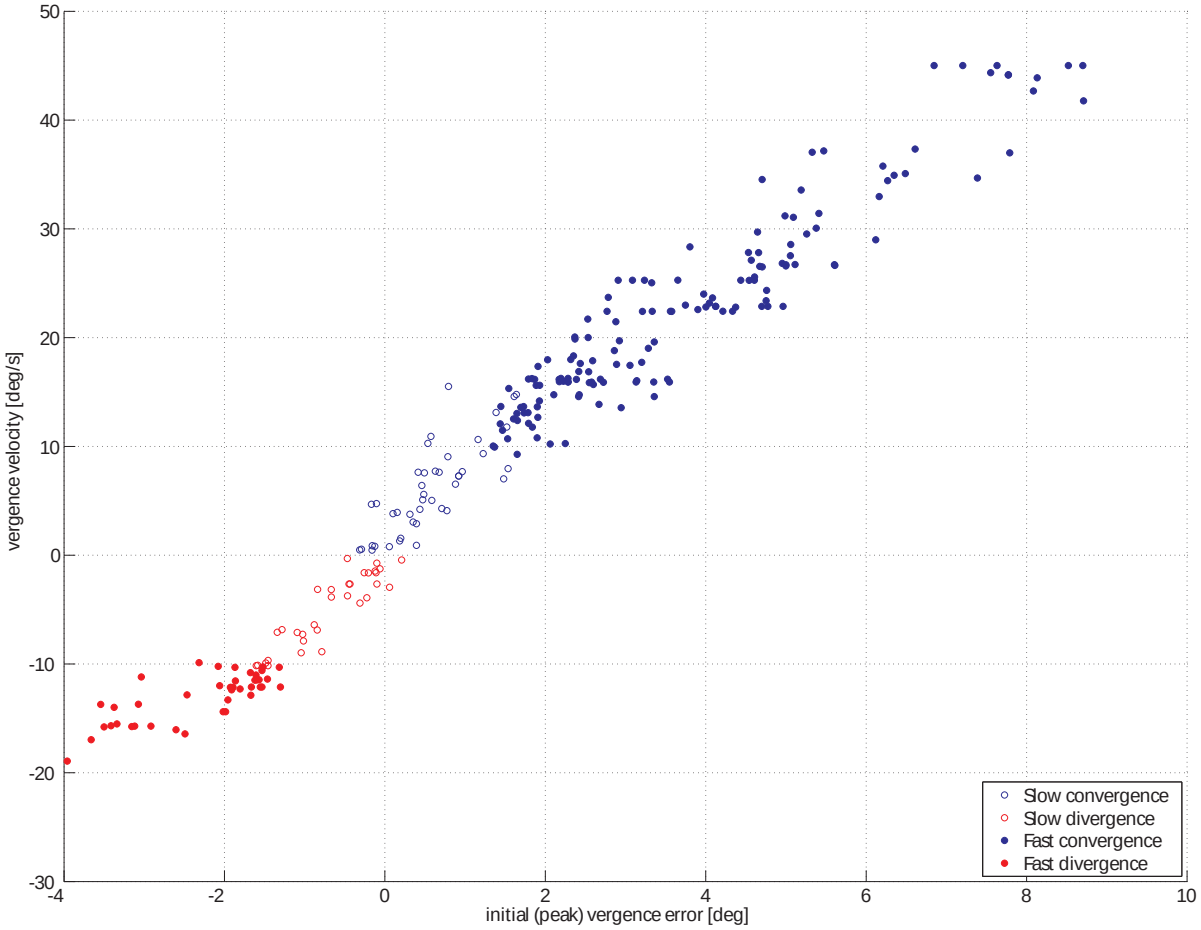


Figure 32: VVCA vergence subsystem main sequence: (peak) velocity of eye vergence *versus* initial vergence error in response to step vergence stimuli. Closed circles are from fast vergence phase and open circles are from slow vergence phase. Similarly to Figure 31, blue color corresponds to convergence and red color to divergence.

promising, they have never been largely employed in real-world applications. With the specific design approach followed to implement the distributed architecture, we demonstrated that we can take full advantage of the flexibility and adaptability of distributed computing to specialize disparity detectors for vergence control and depth vision.

Following an approach similar to [61], we propose to avoid the explicit computation of the disparity map, and to extract the desired vergence angle directly from the population response, over the "foveal" region, of a cortical-like network organized as hierarchy of arrays of binocular complex cells [62]. A neural network paradigm has been chosen for this type of conversion/extraction procedure. Although the paradigm only resorts to a population of neurons in a single scale, we demonstrate that, using a neural network paradigm, accurate and fast vergence control can be achieved in a closed loop, for different orientations of the gaze.

Comparison of the performances of the linear and the convolutional VC networks leads us to a conclusion that, in the simplified case, both networks demonstrate very similar performances. Yet, the convolutional VC-net performs better than the linear one in a more general scenario where any assumption on the scene structure and restrictions on the gaze direction are dropped. The improved performances of the convolutional network comes at the price of a higher number of iterations, which, unfortunately, make convolutional networks much more computationally expensive than the linear-based one.

As an extension of the discussed in Section 4 and Section 5 vergence models, we propose the Vergence-Version Control with Attention effects (VVCA) model. Even though it is still being developed, it already contains some improvements with respect to the discussed earlier vergence control paradigms:

- VVCA provides kinematic eye movement control (*i.e.*, control in terms of rotation velocities);
- VVCA incorporates version control based on an object-related attention signal;
- VVCA is able to reproduce realistic eye movement trajectories.

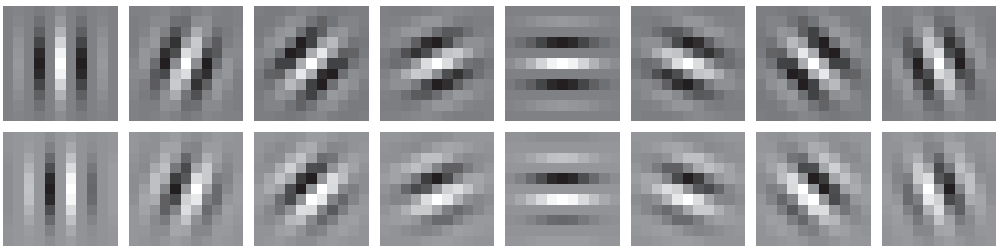


Figure 33: The resulting 11×11 quadrature pair of Gabor filters for $\omega_0 = \pi/2$ and 8 orientations.

A Appendix - Filter design specification

Gabor filters - A Gabor oriented filter along an angle θ with respect to the horizontal axis is defined by:

$$f^{\theta}_{\text{Gabor}}(x, y) = \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}} \mathrm{e}^{j\omega_0(x\cos\theta + y\sin\theta)}$$

where ω_0 is the peak frequency of the filter and σ determines its spatial extension. The spatial window has been chosen as four times σ. At the highest scale (*i.e.*, 11×11 pixels) $\omega_0 = \pi/2$ and $\sigma = 2.67$. Following [21], we implemented the oriented filters as sums of separable filters. By exploiting symmetry considerations, all eight even and odd filters (see Figure 33) can be constructed on the basis of twenty-four 1D convolutions. The 1D filters are modified by enforcing zero DC sensitivity on the resulting 2D filters in which they take part, and by minimizing the difference with the theoretical 2D Gabor filters. Specific care have been paid to adjust the coefficients of each filter function so that the even and odd symmetry is respected. To this purpose, a constrained non-linear multivariable minimization is adopted.

All the filters have been normalized prior to their use in order to have constant energy. The corresponding rosette-like frequency representation of the filters used is shown in Figure 34, for three different scales (octaves).

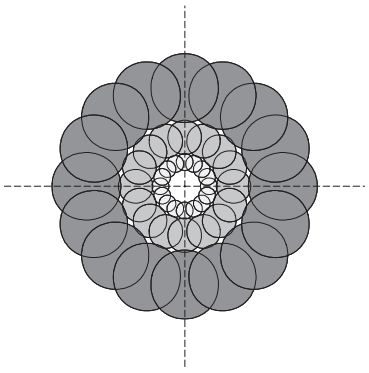


Figure 34: The rosette-like diagram of the multichannel frequency representation obtained by the Gabor filters for three different scales. Contours correspond to half-width cut-off frequencies, and each corona is separated by an octave scale.

References

[1] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367–375, 1987.

[2] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Amer.*, A/2:1160–1169, 1985.

[3] R.A. Young. The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles. Technical Report GMR-4920, General Motors Research, 1985.

[4] A.B. Watson. The cortex transform: rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, 39:311–327, 1987.

[5] M.J. Hawken and A.J. Parker. Spatial properties of neurons in the monkey striate cortex. *Proc. Roy. Soc. Lond. B*, 231:251–288, 1987.

[6] D.J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Amer.*, 4:2379–2394, 1987.

[7] J.B. Martens. The Hermite transform - Theory. *IEEE Trans. Acoust., Speech, Signal Processing*, 38:1595–1606, 1990.

[8] D.G. Stork and H.R. Wilson. Do Gabor functions provide appropriate decriptions of visual cortical receptive fields? *J. Opt. Soc. Amer.*, 7(8):1362–1373, 1990.

[9] J. Yang. Do Gabor functions provide appropriate decriptions of visual cortical receptive fields?: comment. *J. Opt. Soc. Amer.*, 9(2):334–336, 1992.

[10] S.A. Klein and B. Beutter. Minimizing and maximizing the joint space-spatial frequency uncertainty of Gabor-like functions: comment. *J. Opt. Soc. Amer.*, 9(2):337–340, 1992.

[11] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.

[12] T. Reed and H. Wechsler. Segmentation of textured images and gestalt organization using spatial/spatialfrequency representations. *IEEE Trans. Pattern Analysis Mach. Intell.*, 12:1–12, 1990.

[13] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906, 1991.

[14] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. *Image Vis. Comput.*, 10:663–672, 1992.

[15] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *IEEE Trans. on Information Theory*, 38(2):587–607, 1992.

[16] M. Felsberg and G. Sommer. The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision*, 21:5–26, 2004.

[17] L.D. Jacobson and H. Wechsler. Joint spatial/spatial-frequency representation. *Signal Processing*, 14:37–68, 1988.

[18] H. Wechsler. *Computational Vision*. Academic Press, 1990.

[19] R. Navarro, A. Tabernero, and G. Cristobal. Image representation with gabor wavelets and its applications. In P. W. Hawkes, editor, *Advances in Imaging and Electron Physics*, pages 1–84. Academic Press, San Diego CA, 1996.

[20] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, and J.M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[21] O. Nestares, R. Navarro, J. Portilla, and A. Tabernero. Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, 1998.

[22] T.D. Sanger. Stereo disparity computation using Gabor filters. *Biol. Cybern.*, 59:405–418, 1988.

[23] D.J. Fleet, A.D. Jepson, and M.R.M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

[24] F. Solari, S.P. Sabatini, and G.M. Bisio. Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Elect. Letters*, 37(23):1382–1383, 2001.

[25] A.D. Jepson and M.R.M. Jenkin. The fast computation of disparity from phase differences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'89)*, pages 398–403, 1989.

[26] Carl-Johan Westelius. Preattentive gaze control for robot vision. Lic. Thesis LiU-Tek-Lic-1992:14, ISY, Linköping University, SE-581 83 Linköping, Sweden, June 1992. Thesis No. 322, ISBN 91-7870-961-X.

[27] M. J. Morgan and E. Castet. The aperture problem in stereopsis. *Vis Res.*, 37:2737–2744, 1997.

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

[29] D Fleet. Disparity from local weighted phase-correlation. In *Proc. of the IEEE Int. Conf. on Systems,Man and Cybernetics*, volume 1, pages 48–54, 1994.

[30] H. Wagner D.J. Fleet and D.J. Heeger. *Modelling binocular neurons in the primary visual cortex.* Jenkin, M. and Harris, L., Cambridge University Press, 1996.

[31] N. Qian. Computing stereo disparity and motion with known binocular cell properties. *Neural Computation*, 6(3):390–404, 1994.

[32] Freeman R. D. I. Ohzawa and G. C. DeAngelis. Stereoscopic depth discrimination in the visual cortex: Neurons ideally suited as disparity detectors. *Science*, 249:1037–1041, 1990.

[33] B.G. Cumming S.J.D. Prince and A. J. Parker. Range and mechanism of encoding of horizontal disparity in macaque v1. *J. Neurophysiol.*, 87:209–221, 2002.

[34] N. Qian and S. Mikaelian. Relationship between phase and energy methods for disparity computation. *Neural Comp.*, 12:279–292, 2000.

[35] P. Dayan A. Pouget and R. S. Howard. Computation and inference with population codes. *Annu. Rev. Neurosci*, 26:381–410, 2003.

[36] M. Chessa, S.P. Sabatini, and F. Solari. A fast joint bioinspired algorithm for optic flow and two-dimensional disparity estimation. In *International Conference on Computer Vision Systems 09*, Liege, Belgium, 12-15 October, 2009.

[37] Sabatini S.P., G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, M. Van Hulle, N. Pugealult, and N. Krueger. Compact and accurate early vision processing in the harmonic space. In *Proc. VISAPP'07*, 8-11 March, 2007, Barcelona, Spain, 2007.

[38] G.S. Masson, C. Busettini, and F.A. Miles. Vergence eye movements in response to binocular disparity without depth perception. *Nature*, 389:283–286, 1997.

[39] B.G. Cumming and A.J. Parker. Responses of primary visual cortical neurons to binocular disparity without depth perception. *Nature*, 389:280–283, 1997.

[40] D.A. Wismeijer, R. van Ee, and C.J. Erkelens. Depth cues, rather than perceived depth, govern vergence. *Exp. Brain Research*, 184:61–70, 2008.

[41] G. F. Poggio. Mechanism of stereopsis in monkey visual cortex. *Cerebral Cortex*, 5:193–204, 1995.

[42] I. P. Howard R. S. Allison and X. Fang. The stimulus integration area for horizontal vergence. *Exp. Brain Res.*, 156:305–313, 2004.

[43] J. L. Semmlow G. K. Hung and K. J. Ciuffreda. A dual-mode dynamic model of the vergence eye movement system. *Trans. on Biomedical Engineering*, 36(11):1021–1028, 1986.

[44] H. Wagner D.J. Fleet and D.J. Heeger. Neural encoding of binocular disparity: Energy models, position shifts and phase shifts. *Vision Research*, 36(12):1839–1857, 1996.

[45] M. Hansard and R. Horaud. Cyclopean geometry of binocular vision. *J. Opt. Soc. Am.*, 25:2357–2369, 2008.

[46] A. Gibaldi, M. Chessa, A. Canessa, S.P. Sabatini, and F. Solari. A cortical model for binocular vergence control without explicit calculation of disparity. *Neurocomp.*, 73:1065–1073, 2010.

[47] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, volume 2, pages 958–962. IEEE Computer Society, 2003.

[48] Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of CVPR'04*. IEEE Press, 2004.

[49] J. Ruiz-Pinales, R. Jaime-Rivas, E. Lecolinet, and M.J. Castro-Bleda. Cursive word recognition based on interactive activation and early visual processing models. *Int J Neur Syst*, 18(5):419–431, 2008.

[50] Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. In G. Orr and Muller K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324, 1998.

[52] C. Rashbass and G. Westheimer. Disjunctive Eye Movements. *Journal of Phisyology*, (159):339–360, 1961.

[53] G. Westheimer and A.M. Mitchell. Eye movement responses to convergence stimuli. *Archives of Ophthalmology*, 55(6):848, 1956.

[54] V.V. Krishnan and L.A. Stark. A heuristic model for the human vergence eye movement system. *IEEE Trans. Biomed. Eng.*, 24:44–49, 1977.

[55] C.M. Schor. The relationship between fusional vergence eye movements and fixation disparity. *Vision Research*, 19(12):1359–1367, 1979.

[56] G.K. Hung, J.L. Semmlow, and K.J. Ciuffreda. A dual-mode dynamic model of the vergence eye movement system. *IEEE Trans. Biomed. Eng.*, 36(11):1021–1028, 1986.

[57] M. Pobuda and C.J. Erkelens. The relationship between absolute disparity and ocular vergence. *Biological Cybernetics*, 68(3):221–228, 1993.

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

[59] S.S. Patel, H. Ogmen, and B.C. Jiang. Neural network model of short-term horizontal disparity vergence dynamics. *Vision Research*, 37(10):1383–1399, 1996.

[60] J. Horng, J. Semmlow, G.K. Hung, and K. Ciuffreda. Initial component in disparity vergence: A model-based study. *IEEE Trans. Biomed. Eng.*, 45:249–257, 1998.

[61] A. Gibaldi, M. Chessa, A. Canessa, S.P. Sabatini, and F. Solari. A neural model for binocular vergence control without explicit calculation of disparity. In *Proc. European Symposium on Artificial Neural Networks (ESANN'09)*, Bruges, Belgium, April 2009.

[62] M. Chessa, S.P. Sabatini, and F. Solari. A fast joint bioinspired algorithm for optic flow and two-dimensional disparity estimation. In *Proc. International Conference on Computer Vision Systems (ICVS'09)*, Liege, Belgium, October 2009.