EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MASTER THESIS

COGNITIVE SCIENCE

A Likelihood Approach for First Order Depth Estimation from Sparse Stereo-Representation

Author:
Harald-Mircea Papp
Advisor:
Gerrit Ecke

First Examiner:
Prof. Hanspeter Mallot
Second Examiner:
Prof. Felix Wichmann

July 27, 2018

Declaration

I assure to have written this master thesis single-handedly solely with the specified references.

Tübingen, July 27, 2018

____

Name

Acknowledgments

Many people deserve a thanks from the bottom of my heart. I'll just start: Thanks Theresa, my beautiful and lovely soulmate. Thanks to Mama and Tata, for always having my back. Thanks to Gerrit for always answering my many questions. Thanks to Niclas for co-living the stressful Master's life. Thanks to Roland, probably the only person who calmly listened to my confusing, constant blabbering about sparsity and gradient descent. Thanks to Johannes, a good soul of a flatmate, who understood so many forgotten cleaning duties and finally thanks to all my friends for enriching my everyday life.

Abstract

To know how anything is processed in the brain, it is crucial to know, which format information is held in. Incoming sensory data need to be transformed to make processing more efficient: Some redundancies are needed, some only cost capacity. Sparse coding proved itself valuable, as it minimizes energy cost while retaining many useful features of the environment. It has been broadly investigated in many areas, however depth perception remains nearly untouched.

This work shows that a suitable representation of natural images, allows inference of depth at different orders of complexity. For that reason a binocular observer is simulated to generate a dataset of half-image pairs, based on textured surfaces, manipulated in space. A Sparse Convolutional Artificial Neural Network transforms these stimuli to a sparse stereo-representation.

Even through rudimentary methods, employing no prior knowledge of network architecture or quality of stimuli, manipulation parameters of depth could be inferred up to a certain point. This corroborates, that the sparse coding approach is a viable model for sensory information representation.

Contents

List of Figures

Chapter 1

Introduction

Seeing something seems simple, at first glance. We just *know* that an object is rounded or green or even an object. It is obvious, that something lies before something else or that the facade of a house lays slanted before us. There are powers at work, that lie beyond our consciousness, covert to introspection. But somehow we *can* infer a wide spectrum of features about our environment - simply from patterns of bouncing light, hitting our eyes.

This work focuses on one of these features, namely depth, which is all so needed in our daily lives. But the seemingly trivial statement *The distance to the hill is about x meters*, requires some not so trivial computation.

Visual Depth Processing in the Primate Brain

If you close one eye, you are still able to put the above-mentioned *hill* into perspective. In fact there are plenty of *monocular* depth cues, such as Motion parallax (Ferris, 1972) or shading (Lipton, 1982). Yet, other tasks become harder, for example touching the left tip of the index finger with the right tip of the index finger, right before your eyes. Some information is lacking for a vivid impression of depth - information, that your second eye provides.

Depth from Stereo

The fact, that the eyes are horizontally separated, leads to two different perspectives on the environment around you. Each eye has its own 2D-projection of the world on its retina. The differences between the so-called *half-images* are termed disparities and used as strong *binocular* cues (Parker, 2007).

To understand this concept of Stereopsis, one has to imagine the two eyes having exactly the same retinae. Every location on one retina has a respective counter-location on the other. When a primate focuses an arbitrary point in space, the eyes move in an opposing, inward, but symmetric fashion.

The focus point is brought to the middle of both the foveas (called vergence shift (Pierrot-Deseilligny et al., 2004)). By definition, the two retinal images have no differences in this focus point, ergo no disparity. Around the focus point, every single point from the environment projects on different locations on the retinae (respective to the fovea). By integrating the two half-images, a pattern of disparities arises. One can think of a depth-map. But still one problem remains: how does the brain *establish*, which points on the retinae correspond to the same point in the environment. This correspondence problem will be adressed later.

Indeed physiological evidence from macaque monkeys shows that, as soon as the half-image information is integrated, disparity-sensitive, binocular neurons can be found (in V1 (Kandel et al., 2000) for Absolute Disparity (Kruger et al., 2013), in V2 for Relative Disparity, as well as V4 and V5 (Parker, 2007)). As for this work, the macaque monkey brain holds as an animal model, as it was shown to achieve similar thresholds for 3D-object discrimination and detection in psychophysical paradigms, as humans do (Janssen et al., 2003; Verhoef et al., 2010; Orban, 2011).

Absolute and Relative Disparity

The mathematical definition of disparity should be readily describable. However it has some easy-to interchange terms, that might lead to confusion.

Parker (2007) for example, distinguishes between absolute disparity:

Focus on a point P in space. The projection of P falls along the optical axes onto the fovea. Consider a second point Q. It projects onto different locations on the retinae. The absolute disparity of Q is defined as the angular difference of Qs projections with respect to the fovea.

and relative disparity: the difference of the absolute disparities of two points.

Figure 1.1 subsumes the confusion. While the absolute disparity of Q is $\alpha - \beta$, the relative disparity of Q in respect to P is also $\alpha - \beta$. However, if one would focus on another point than P, absolute and relative disparity would differ. As Mallot (2000) (p. 129) points out, the relative disparity can also be calculated as the difference between the angles at P and Q, irrespective of focus point.

Hence, absolute disparity (with sensitive neurons in V1) establishes a reference system in respect to the fovea - it moves, as the gaze moves. Human stereopsis on the contrary seems to rely on relative disparity (with sensitive neurons in V2) (Parker, 2007)). This appears plausible, since only with the help of relative disparities one can start to put different objects into depth-context and infer the 3D-shape (Kruger et al., 2013).

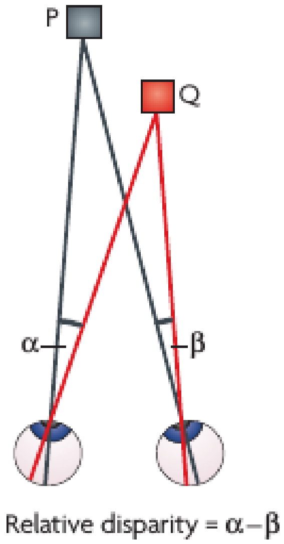


Figure 1.1: Eyes focusing at point P. P projects onto the middle of the fovea in both eyes. Qs projection on the contrary, is on different locations on the retinae. The different locations of the red lines are incorporated by α and β. (Figure from: Parker (2007), p. 381)

Building Images from Scratch

Still, it is not enough to know, what brain region is sensitive to which information. It is also crucial to know, which format it might be held in. The retinal receptors receive a constant bombardment of photons, resulting in a mass of data, which can neither be fully transported via the optic nerve, nor fully processed (Zhaoping, 2006). Thus, the information needs to be held in a format, which appropriately minimizes redundancies, while maximizing brain resources (Simoncelli, 2003). There surely are data compression methods, like the JPEG-algorithm, that can compress images up to 20 fold, without noticable information loss (Zhaoping, 2006), but this must not mean, the brain applies similar mechanisms.

Therefore, Barlow (1961) proposed the hypothesis of efficient coding, sparsely broken down to: the code (or format) in which the brain holds information, should be adapted to stimuli from the individual's environment. The code should minimize action potential (*efficient*), but retain all useful information for stimulus processing. On the one hand side, this corroborates an evolutionary approach, on the other hand (in terms of vision), it breaks down the possible image-space to a clearly defined subset of images, which indeed show specific statistical properties, to be exploited: so-called natural images.

Olshausen & Field (1996) proposed a promising algorithm which gives rise to such an efficient code. Not only do the code's atoms (Fig. 1.2) reflect some of the characteristics of mammalian simple cells of the visual cortex: they are spatially localized, oriented and bandpass (Hubel & Wiesel, 1968; Olshausen & Field, 1996), they also *emerge* from a set of simple rules.

Basis Functions, Gabor Filters and Kernels

To grasp the algorithm's concept, one has to make the basic assumption, that an image I can be represented by a linear combination of basis functions Φ_i:

$$I = \sum_i a_i \Phi_i \tag{1.1}$$

The set of these functions (termed dictionary) should enable a reconstruction of *every* image in the set of natural images and thusly forms a complete code (Olshausen & Field, 1996). There are many methods of achieving such codes, for example Principle Component Analysis (PCA). Yet, the resulting basis functions of most methods do not resemble any physiologically confirmed properties . This stems from the fact, that natural images mainly show higher-order statistical dependencies, that can't be accounted for by (in the case of PCA) linear decorrelation (Olshausen & Field, 1996). Natural images show highly non-gaussian behaviors (Ruderman & Bialek, 1994).
The resulting basis functions from Figure 1.2 on the contrary, capture the quality of simple cell receptive fields. They seem to resemble 2D Gabor filters, which are known to be a model for neurons from V1 (Jones & Palmer, 1987). Gabor filters are sine/cosine functions enveloped by a Gauss funtion, where positive sine/cosine parts resemble the excitatory ON-region of a receptive field, while negative parts resemble the inhibitory OFF-region.
The basis functions can also be interpreted as (convolutional) kernels, known from image processing, where they act as edge or feature detectors Mallot (2000), p.78-93). This latter fact will be revised in section 1.4.

Sparseness

What distinguishes the method of obtaining Gabor-shaped basis functions (Olshausen & Field, 1996), from simple PCA-like methods, is the introduction of *sparseness.* Not only should an image be reconstructed well, it should be reconstructed well, by as few basis functions, as possible: Most activation weights a_i from equation 1.1 should equal to zero (for one image). This creates a trade-off between goodness of reconstruction

$$[\text{preserve information}] = -\sum_{|Images|} [I - \sum_i a_i \Phi_i]^2 \tag{1.2}$$

and sparseness of the code

$$[\text{sparseness of } a_i] = -\sum_i S(\frac{a_i}{\sigma}) \tag{1.3}$$

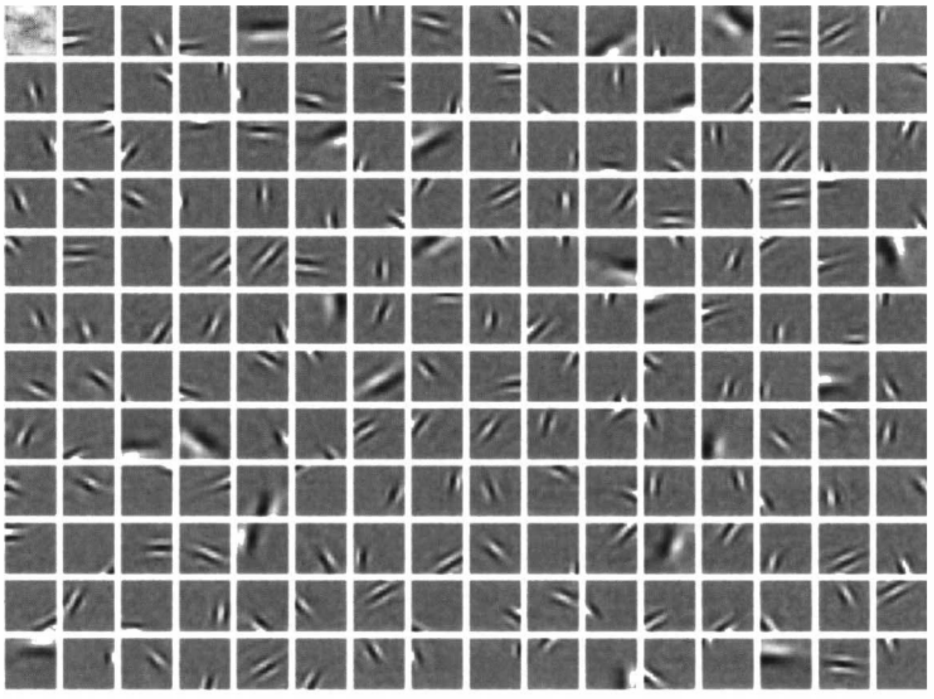


Figure 1.2: 192 basis functions. The shape of the functions resemble Gabor filter-like structure. If an edge-like feature from an image patch falls into the white edge-like area of one filter, the filter enhances the edge-structure, while inhibiting the adjacent parallel parts (that fall on the black filter-parts). This dictionary was obtained by training of 16×16-pixel patches from ten 512×512-pixel natural images (Figure from: Olshausen & Field (1996), p. 609).

which is to be minimized:

$$E = -[\text{preserve information}] - \lambda[\text{sparseness of } a_i] \tag{1.4}$$

Here $S(x)$ is a function, that weights the sparseness. The worse a reconstruction from 1.2 or the more a_i are active in 1.3, the bigger resulting values from equation 1.4 become. λ is a trade-off factor, which modulates how much sparsity is taken into account for minimization. Elements from a sparse activition weights vector A then follow a zero-peaked LaPlace distribution, with most a_i inactive for specific image reconstructions.

Finding a solution for the minimum of E, results in the optimal sparse code.

Sparse Codes for Stereo Stimuli

One crucial property of early vision remains uncaptured by the above mentioned sparse codes: sensitivity to disparity (1.1). Ensuing from binocular vision, Lundquist et al. (2016) presented an approach, how the minimization problem in equation 1.4 could be extended, giving rise to binocular basis functions, seen in Figure 1.3. A binocular disparity neuron should then have different receptive fields, one for each eye. The two receptive fields are slightly shifted (in position or phase), making the binocular neuron sensitive to the feature encoded by its Gabor *at* a given disparity (defined by the shift and/or phase).

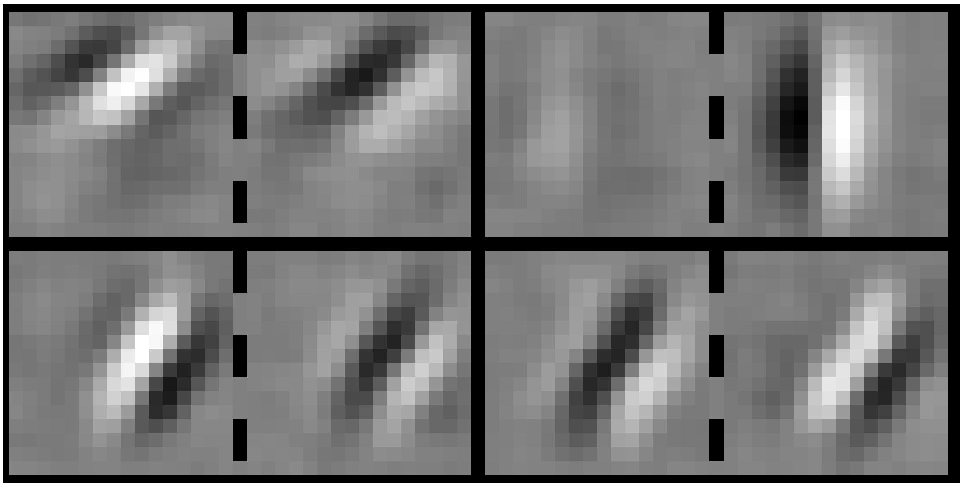


Figure 1.3: Four basis function pairs. One pair mirrors the two receptive fields of one binocular neuron. The pair on the upper left shows shift in posistion, the pair on the lower right shows shift in phase.

Therefore, now two images, namely the half-images from each eyes's retina, should be reconstructed as good as possible, while sharing the same activation weights vector.

$$E = \frac{1}{2}\left(\|G(I_L, \Phi_l, A)\|_2^2 + \|G(I_R, \Phi_r, A)\|_2^2 \right) + \lambda \|A\|_p \tag{1.5}$$

with

$$G(I, \Phi, A) = I - \sum_i a_i \phi_i \tag{1.6}$$

Here the function $G()$ captures the residual from equation 1.2. In addition, the elements from Φ should not only compete in encoding of single independent image patches (as in conventional sparse coding (Lundquist et al., 2016)), they should locally compete for more patches.

Convolutional Networks and Overcompleteness

For this matter, the basis functions - now in light of image convolutions, interpreted as convolutional kernels - are replicated over the whole image. One can imagine to overlap a kernel with size of $r \times r$-pixel over a $r \times r$-pixel patch of the image (see bottom-side of Figure 1.4). The value of the mid point of the image patch is then summed with the weightened (weightening according to kernel values) values of its neighbours. Information is accordingly condensed from a $r \times r$-pixel patch to a single pixel value. The kernel is then translated by a stride of p_x pixel in x direction or p_y pixel in y direction (*replication*), so that the whole image is covered. Edge effects should be considered, as the kernel is overlapping corners or edge points of the image.

r specifies how much of the neighboring area should be taken into account for one resulting pixel value. If the stride $p < r$, the to-be convoluted image patches overlap after translating a kernel, meaning that two adjacent result pixel values share information from the original image.

An $M \times N$-pixel image, convoluted with a kernel f at stride $p_x = p_y = s$, then results in a layer of $\frac{M}{s} \times \frac{N}{s}$ pixel and consecutively, in $\frac{M}{s} \times \frac{N}{s}$ replications of f. One such layer is then called a *feature map* z. Repeating this procedure with all kernels (*basis functions*) contained in Φ results in $|\Phi|$ feature maps. Note that the resulting size of z is independent of the size of f.

In this way the convolution can be inverted. The inversion reframes the mathematical problem of image reconstructions through sparse codes: *De*-convolutional networks (Zeiler et al., 2010) approximate images with the aid of feature maps z_j:

$$I_{L,R} \approx \sum_{j}^{J} f_j^{L,R} * z_j \tag{1.7}$$

$f_j^{L,R}$ are binocular basis functions (1.3) from a dictionary of size J. The z_j incorporate the sparse activation weights, retaining spacial organization and independent of L or R. Finding the minimum of the extended (from 1.5) energy function:

$$E = \frac{1}{2}\Big(\|D(I_L, \Phi_L, Z)\|_2^2 + \|D(I_R, \Phi_R, Z)\|_2^2\Big) + C(z) \tag{1.8}$$

with

$$
\begin{aligned}
D(I, \Phi, Z) &= I - \sum_{j}^{J} \Phi_j * z_j && \text{residual} \\
C(z) &= \sum_{j=1}^{J} C_\lambda(Z_n) && \text{sparsity term} \\
C_\lambda(a) &= \begin{cases} \lambda, & \text{if } |a| \geq \lambda \\ 0, & \text{else} \end{cases} && \text{cost function}
\end{aligned}
$$

yields a layer in form of a $(\frac{M}{p_y} \times \frac{N}{p_x}) \times J$ matrix (see Figure 1.4), resembling a model for V1 ((Schultz et al., 2014)). Every element can be interpreted as a neuron with a Gabor-like receptive field (depending on which feature map, the neuron is located in) for image feature encoding and disparity sensitivity through binocularity (Lundquist et al., 2016).
The dictionary size $K = \frac{M}{p_y} \times \frac{N}{p_x} \times J$ was disregarded so far, although it has an impact on the shape of the basis functions.
As there is only *one* possible linear combination of dictionary elements, approximating (arbitrarily well, depending on the chosen norm) any image, the dictionary is called *complete.* If an element is removed from the dictionary and it is still possible to approximate any image, such a dictionary is called *overcomplete* (Heil, 2010).
In case of image reconstruction it is possible to compute the order of overcompleteness in form of the overcompleteness factor (for stride $p_x = p_y = s \neq 1$):

$$\text{overcompleteness factor} = \frac{J}{s^2 \times 2} \tag{1.9}$$

According to Schultz et al. (2014) basis functions start to look less like Gabor filters, as the overcompleteness factor becomes bigger. Because more dictionary elements are availible, more unconventional features can be encoded by basis functions, giving rise to, amongst others, end stopping filters, similarly found in V1 (Pack et al., 2003).

Different Orders of Depth

With the aid of the V1 model, one could infer a manifold of information about the image. The population of neurons sensitive to disparity, can again show many forms of statistical dependencies, reflecting the depth structure of the image. Disparity patterns, however, are not repetitious or constant across the whole field of view. Fronto-parallel planes, slanted or tilted objects and curved structures all give rise to specific, discernible disparity-behavior along their surface. Because of this, depth-structures can be categorized into orders of depth.

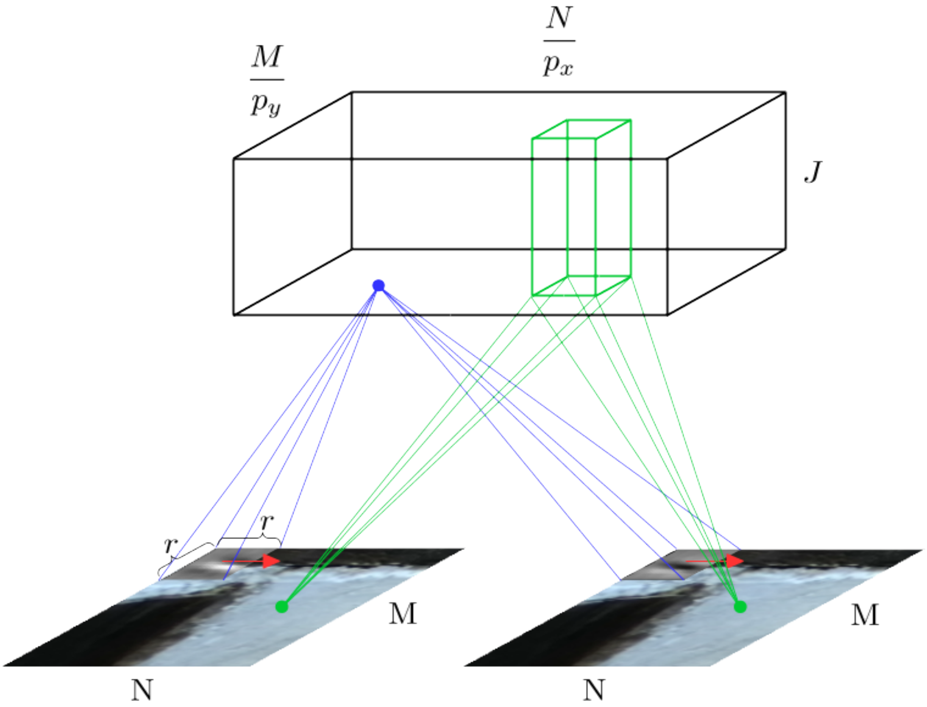


Figure 1.4: The blue point depicts a binocular neuron with two shifted receptive fields. Each receptive field congregates information from a $r \times r$ pixel patch, through convolution. The blue point then integrates the two responses into the value of one voxel. The red arrows depict the stride p_x in x-direction. Their value determines the width of the upper layer $\frac{N}{p_x}$. Similarly a stride in y-direction determines the height $\frac{M}{p_y}$. The whole green column from the upper V1 layer holds information from the green points in the two half-images. The Figure was adapted from (Schultz et al., 2014; Lundquist et al., 2016)

The zeroth depth order is simply an observer-perceived distance, for example to the focus point. The observer can tell, if something is in front of something else or the distance between two depth planes (Anzai & DeAngelis, 2010). The distance to a focus point could simply be estimated through the vergence angle (see 1.1).

First-order depth occurs at tilted and/or slanted planes. As one focuses at such a surface, the disparity is at no two points the same, but, the change-rate of disparity across the surface remains constant. Thereby first-order depth could be interpreted as the derivative of depth, along an axis in the fronto-parallel plane (Orban, 2011). With such information, object orientation in space can be deduced.

Second-order depth appears along curvatures, such as convex or concave shapes. Here, neither disparity, nor disparity change is constant, but the amount at which the disparity change itself changes, is constant, tantamount to the second derivative along an axis in the fronto-parallel plane (Orban, 2011).

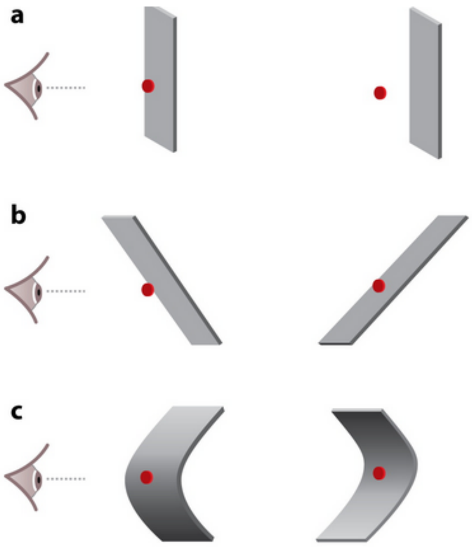


Figure 1.5: *a*: zeroth-order depth, simple distance. An observer can tell if an object is behind or in front of something. *b* first-order depth, depth-gradient along the surface. An observer can tell the slant and tilt of the surface. *c* second-order depth, change of depth-gradient along a curved structure. An observer can tell if an object is concave or convex, as well as the grade of curvature. (Figure from: (Orban, 2011))

Taira et al. (2000) and Tsutsui et al. (2001) showed the caudal intraparietal area (CIP) to contain neurons, selective to first-order depth, whereas Srivastava et al. (2009) showed neurons in the anterior intraparietal area (AIP) to also be sensitive to disparity-gradient. Thus, the pathway for extraction of first-order depth structure through disparity is proposed by Orban (2011) to follow V1 → V3A → CIP → AIP.

Summary and Thesis Statement

Light reflects off the environment and falls through the pupils of an observer's both eyes, creating two different half-images, exciting the photoreceptors in the retinea (1.1). Light information is encoded into an efficient neural code(1.2). The different information from each eye is integrated in the visual cortex: Binocular neurons with slightly shifted receptive fields are excited by disparity (1.3). A population of such simple cell neurons then shows statistical patterns, depending on the depth structure of the environment in the field of view (1.4. Converging information from simple to complex cells must be held in such a form, that following processing areas, such as CIP

and AIP, can infer specific, complex disparity patterns, such as disparity gradients along fronto-parallel axis, evoked by first-order depth(1.5).
This work presents a new likelihood approach, loosely resembling complex cells in CIP and AIP. This does not mean, that either CIP or AIP work in such a manner. The main emphasis is put on showing, that sparse coding transforms visual information in such a manner, that basic methods, such as a naive Bayes Classifier, could infer different orders of depth - a sparse code is a useful code in the Barlowian sense.
Stimuli are half-image pairs of textured planes, horizontally and vertically translated, for zeroth order depth and slanted/tilted at different levels for first order depth. The stimuli are perceived by a binocular observer in a simulated experiment. Half-image information is encoded via a Sparse Convolutional Artificial Neural Network (SCANN) at two different overcompleteness factors. The neural code is then read out. Artificial Neurons from the model show zeroth and first-order depth sensitivity in their tuning maps. A probabilistic interpretation of the tuning maps allows inference of the translation or tilt/slant parameters, and thus depth information of the input stimulus. Additionally the number of neurons, used for inference is varied: either the centered column of 5×5 ($m = 5$) or the centered column of 7×7 ($m = 7$) from the V1 model, is taken into account.
I hypothesize, that:

1. Neuronal sets at a higher overcompleteness factor allow better inference. As more kernels are availible, disparity can be encoded more accurately and unambiguously.

2. Neuronal sets at $m = 7$ allow better inference. The wider column contains more excentric neurons, which are exposed to higher disparity for tilted/slanted surfaces. As will be later explained, the farer a pixel is from the focus point (center of tilt and slant), the more discernibly its disparity is, for its level.

This thesis is a parallel work for a yet to-be written publication (Ecke (n.d.),unpublished), which mainly focuses on zeroth-order depth estimation with the same approach. While showing in my work, that zeroth-order depth information *can* be inferred from a binocular sparse representation, my analysis will be bound to first-order depth estimation.

Chapter 2

Methods

This work solely employs computational methods. A simulated observer's retinal preprocessing and early vision, up to higher visual processing areas are modelled, using the concepts mentioned in the introduction. In general the path of information processing can be divided into three parts: Stimulus presentation in form of a simulated experiment, the neural model in form of a artificial neural network and inference in form of a naive Bayes classifier-like method.

Experiment and Stimuli

In context of my work, the experiment models the conditions before information reaches the retina - *what is the observer looking at?* Light hitting an image-textured surface, reflecting off it and being perceived by two horizontally separated eyes. The experimental paradigm therefore encompasses the light source, the surface's texture, reflexional properties and its position and orientation, as well as the eye's position in respect to the surface. The resulting stimuli are two half-images: one projection of the surface to the right eye, one projection to the left eye.

Geometry of the Experiment

For purposes of clarity, geometry will first be summarized in the one dimensional case of the experiment, depicted in Figure 2.1. The free parameter α provides only one possibility (thus, the one-dimensional case) to manipulate the surface in two-dimensional space. Manipulation will later be extended to the two-dimensional case: slant and tilt in three-dimensional space. In addition the geometry and following computations assume linear retinae.

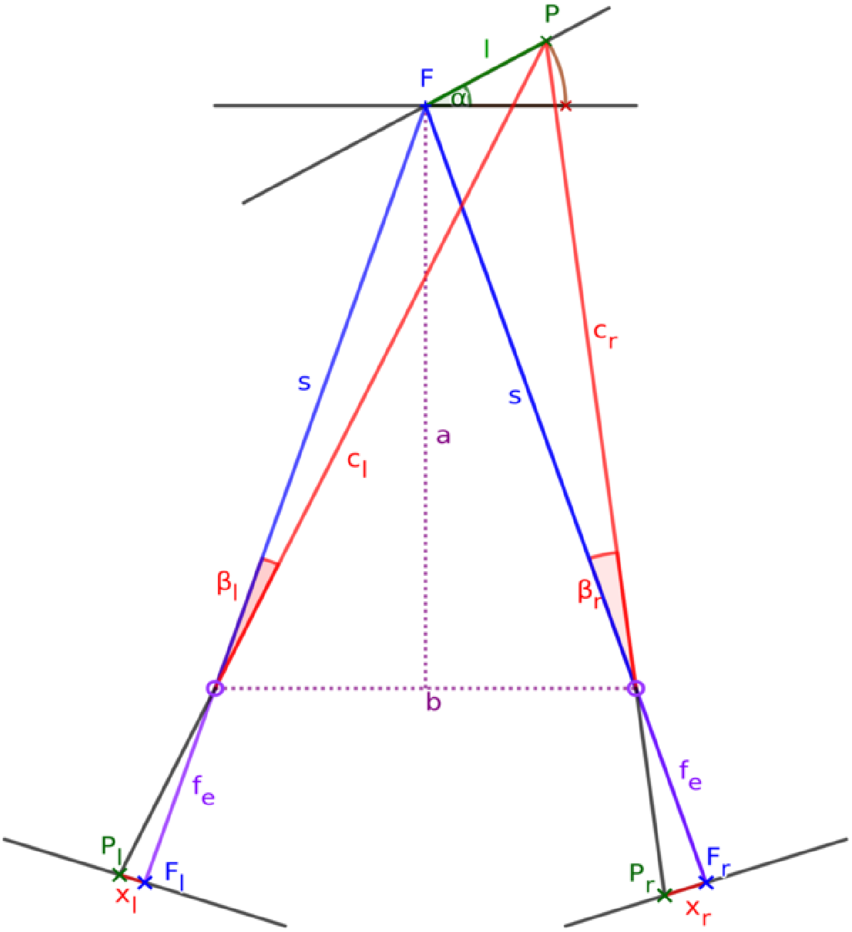


Figure 2.1: The one-dimensional case of the experimental paradigm. Linear retinae are assumed. Green color indicates the values under scrutiny: α: slant of the surface, l: distance of the point, whose disparity is calculated. Purple color indicates parameters, which summerize the observer: a: distance of the observer to the focus point, b: distance between the (pupils of the) eyes, f_e: depth of the eyeball. Blue color indicates the focus point F and its projections (F_l, F_r) with length s. Red color indicates the values, which need to be computed to be able to compute α and l.

In Figure 2.1 purple lines and characters are fixed parameters: a is the distance of the observer's head to the surface, b is the distance between the eyes and f_e is the depth of the eyeball (assumed to be the same for both eyes). From a and b, the distance s (eye's lenses to the focus point F), as well as the vergence angle ($\sphericalangle F_r F F_l$) can be calculated. Any point P with distance l from the focus point F, projects to different points on the eyes retinae: P_l in the left eye and P_r in the right eye. The distance of these projection points P_l, P_r on the retinae, to the middle of the foveae (and hence to the projections of the focus point F_l and F_r) is described by x_l for

the left eye and x_r for the right eye. The difference $x_r - x_l$ is the disparity d, evoked by P. The experiment depicts the case mentioned in section 1.1.2, where absolute disparity is equal to relative disparity.
As either α or l changes, both x_l and x_r change and thus the disparity.

$$d(\alpha, l) = x_r(\alpha, l) - x_l(\alpha, l) \tag{2.1}$$

Consequently, equation 2.1 shows the dependencies of the disparity terms and the free parameters.
For a realistic set of fixed parameters, the depiction of the experiment in Figure 2.1 would be out of scale. Further calculations and derivations of the formulas needed for the geometry can be found in the appendix A.1.

Non-Linear Behaviour of Disparity in Rotational Paradigms

To be able to evoke standardized disparity change on the retina, it is crucial to know how a change in α affects disparity across the surface. [1]. Equation 2.1 can be written out as:

$$x_r(\alpha, l) = f_e \cdot l \cdot \frac{\cos(\alpha) + \frac{b}{2a} \cdot \sin(\alpha)}{\sqrt{U - V - \left(l \cdot (\cos(\alpha) + (\frac{b}{2a}) \sin(\alpha))\right)^2}} \tag{2.2}$$

$$x_l(\alpha, l) = f_e \cdot l \cdot \frac{\cos(\alpha) - \frac{b}{2a} \cdot \sin(\alpha)}{\sqrt{U + V - \left(l \cdot (\cos(\alpha) - (\frac{b}{2a}) \sin(\alpha))\right)^2}} \tag{2.3}$$

with

$$U = s^2 + l^2 + s^2 \cdot (\frac{b}{2a})^2 + l^2 \cdot (\frac{b}{2a})^2$$
$$V = 2 \cdot s \cdot l \cdot \frac{b}{2a} \cdot \cos(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2} + 2 \cdot s \cdot l \cdot \sin(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2}$$

The connection of α, l and the disparity turns out to be non-linear, as can also be seen in Figure 2.2 (for detailed derivation of equation 2.2 and 2.3, see Appendix A.1).
The focus point (F with $l = 0$) in contrast, is independent of α: The disparity is always 0. Because of the experiment's construction, the focus point is the foremost point of the observer's horopter. A rising α results in a counter-clockwise rotation of the surface. Because any point with positive distance l will be farther away (evoking uncrossed disparity) and on the right side of the focus point, x_r is always bigger or equal to x_l, as $\alpha \in [0, \frac{\pi}{2})$. If $\alpha \geq \frac{\pi}{2}$,

[1] This is also dependent on l, but to be neglected, as explained in section 2.1.1

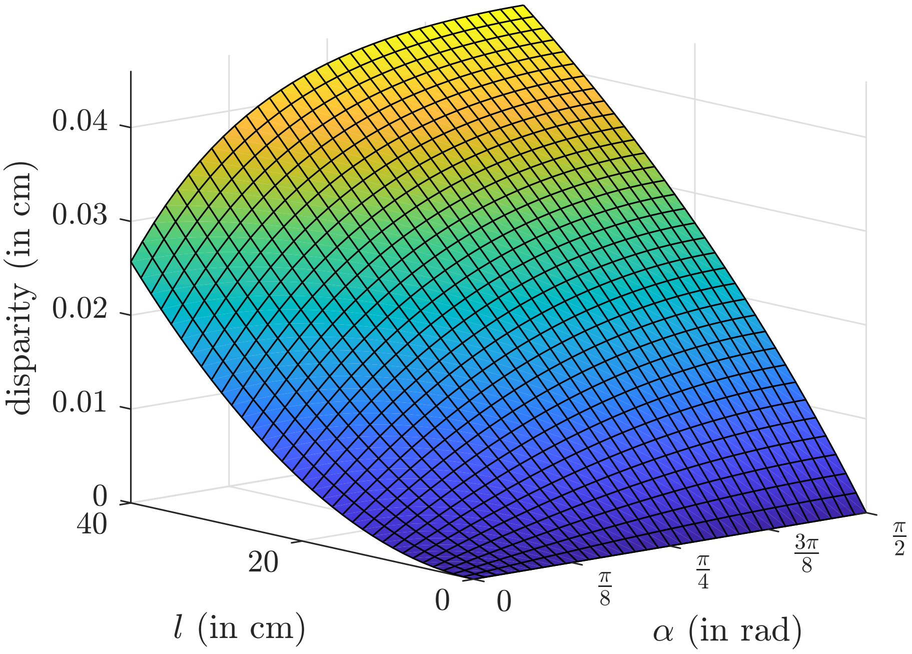


Figure 2.2: The function 2.1 in the relevant interval: The surface is 1m long, therefore one side can maximally be 50cm long. The surface can be slanted no more than 90° ($= \frac{\pi}{2}$ in rad).

x_l becomes bigger than x_r and thus, a negative disparity arises. Similarly, a negative l (P is before the focus point, evoking crossed disparity) leads to a sign switch.

Standardized Stimuli

To allow for later classification (see section 2.4), resulting stimuli must be relatable to clearly discernible subsets, thus the continuous α-rotation must be discretized. Certain α should be chosen, so that standardized classes arise. As inference will work by means of disparity, those α are needed, which result in equally-spaced disparity classes.

If the unit-disparity is d_u, then a disparity class (later referred to as stimulus class) C_i should encompass all stimuli, which give rise to disparity

$$d_i = i \cdot d_u. \tag{2.4}$$

For a proper set of classes, $i = 11$ and $d_u = 0.004$ cm are fixed (see 2.3). To then find the class-associated α_i, a reference needs to be created. This is accomplished by fixing the displacement x_l of P_l. This allows to now calculate which α is needed, to create the desired disparity d_i in reference to P_l.

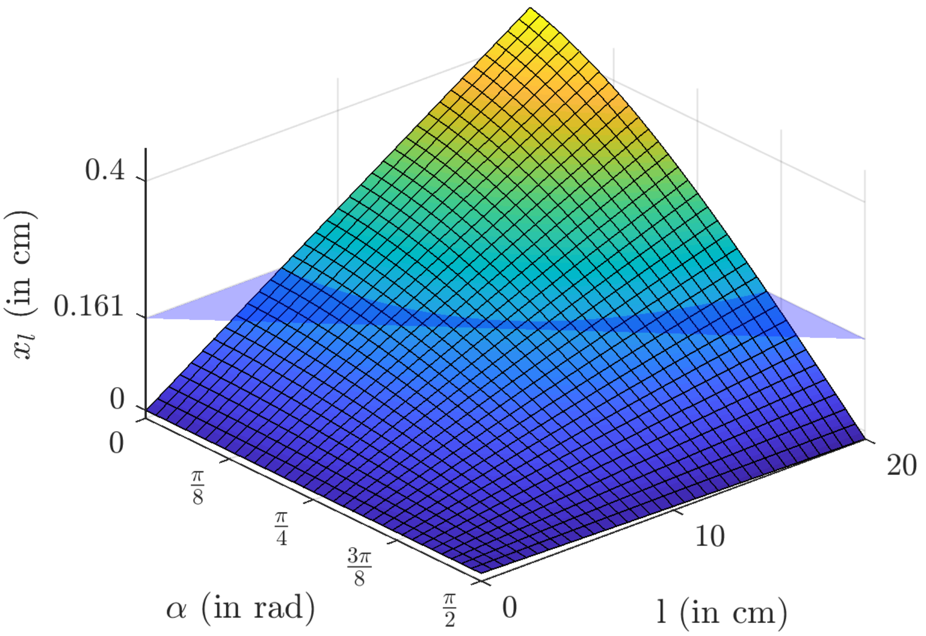


Figure 2.3: Function of $x_l(\alpha, l)$. $0.161 = \frac{100}{621}$ cm corresponds to ten pixels. The intersection of the blue plane and the function, describes all possible α, l combinations, which give rise to a disparity of 1 pixel for x_l

Figure 2.3 shows, that all combinations of α and l on the curve arising from intersecting the blue plane and the x_l-surface (for fixed P_l-displacement) result in a displacement of $x_l = \frac{100}{621}$. Ergo, the intersection could be described by a function $l = f_{P_l}(\alpha)$ shedding light on the dependency of α and l. By always taking the same l at all d_i and rewriting equation 2.1 as equation 2.5, it is now possible to solve for α at d_i.

$$d_i = x_r(\alpha, f_{P_l}(\alpha)) - \frac{100}{621} \tag{2.5}$$

The results for α can be seen in Table 2.4.

Because the $x_l(\alpha, l)$ is symmetric along the l, x_l-axis, negative values of α_i for $i = 1, 2, 3, 4, 5$ also give rise to equally sized C_i, making a total of 11 classes. The exact calculations for the corresponding α_i were performed with the Symbolic Math Toolbox from Matlab (Matlab R2017).

In three dimensions rotation can mathematically be expressed in many ways, spanning from Euler Rotation to Quaternions. I chose the description in Axis-Angle format: an axis v is specified in form of a three dimensional vector in space, while the angle α specifies the rotation around v.

The one-dimensional experiment described above, can be regarded as the two-dimensional case, with $v = [cos(\frac{3\pi}{2}), sin(\frac{3\pi}{2}), 0]'$. Let the tilt ϕ be the

i	displacement of x_r in respect to x_l in cm	α_i in degrees
0	0	0
1	0.004	6.0444
2	0.008	24.2876
3	0.012	38.2489
4	0.016	48.1717
5	0.02	55.1963

Figure 2.4: Different levels, at which the stimulus needs to be slanted, to evoke certain disparities of one point (at the same distance l) between the half-images. A clear non-linearity can be seen.

angle between v and the x-Axis. Then the tilt axis for the one-dimensional case, is at $\phi = 270°$, describing the negative normal of the xy-plane through the focus point F in Figure 2.1. By manipulating ϕ, α now evokes disparity in a circular fashion on the retina (Figure 2.5). Let α be the slant of the surface.
A more in-depth approach on the visualization of the observer-stimulus relationship, can be found in section 2.3.2.

Implementation

The experiment's geometry was reconstructed in virtual space via Blender (Version v2.76 run on Kubuntu 16.04). The surface was chosen to be a simple plane of 1m^2, centered around the origin of the virtual coordinate system. Likewise, rotation occured around the origin. The plane was textured with one of 1005, 1242 × 1242-pixel images (see 2.1.1). Consequently, one pixel covers the area of 0.0805^{2}cm^2, if the surface is not manipulated.
Specular intensity was turned off, while diffuse intensity was set to the maximum. No light-induced depth cues were therefore present. Additionally, atmospheric interaction was also turned off.
Each eye was simulated by a camera with a field of view of 11.77°. Thus, the field of view exactly encompasses the later render-resolution. The cameras were 7 cm (b in Figure 2.1) apart - an estimation of the distance between the eyes. The distance a from the surface to the mid point between the cameras was set to 1 m. To simulate the eye's fixation on the plane, the cameras were inwardly rotated by 2.005° ($\frac{\text{vergence angle}}{2}$). The default Blender Render was used at a resolution of 256×256-pixel (field of view), taking for every camera one snapshot. This covers a small portion of about 20.611^{2}cm^2 from the original image[2]. After plane rotation, the camera positions were randomly initialized, so that distance from the mid point between the cameras and the

[2] through rotation, a smaller part of the nearer part of the plane is captured, while a bigger part of the farer part of the plane is captured

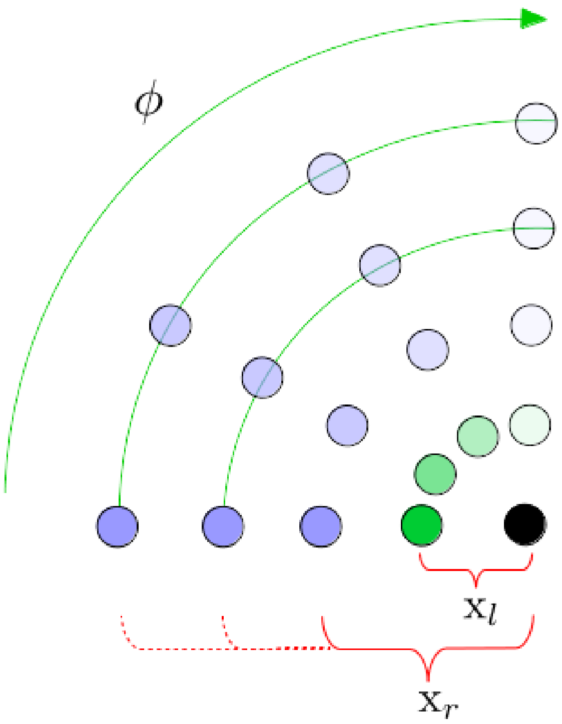


Figure 2.5: x_l (green point) is fixed. According to the computed α, the displacement x_r is bigger or smaller (purple points). By tilting at an angle of ϕ (green arrow), the x_l and x_r are rotated in respect to the focus point (black dot), succesively covering the whole retina.

midpoint of the field of view constantly remained at 1m. Random camera initiation was corrected for the surface's edges, so that even at high slants, only images within surface boundaries were captured.

Automation of the stimulus creation occured via the built-in python library bpy in Blender.

Image Dataset

Images for texturing were taken from the image dataset of Kevin Reich's Bachelor Thesis (Reich, 2017). They were shot, using a ZED Stereo Camera by Stereolabs. Original images were glued-together half-images at a resolution of 4426 × 1242-pixel. For purposes of texturing original images were cut in half, resulting in a 2213 × 1242-pixel monocular images. Because the plane in Blender was quadratic, the left-most part of 1242 × 1242-pixels was cropped. An examplary sample of the database can be found in the appendix A.3.

Tilted/Slanted Stimuli

The full set of stimuli encompassed 10 random snapshots of all 1005 images, at every parameter combination of 11 α and 18 ϕ, resulting in a total of 1989900 half-images per eye or, in other words, 10050 images for 198 stimulus classes. The rotational axis $v = [sin\phi, cos\phi, 0]'$ was set from $\phi = 0°$, resembling slants around the x-Axis, to $\phi = 170°$ in steps of 10°. Consequently, sampling only half a circle with the tilt, still leads to stimuli over the whole retina, as long as negative slants are allowed. Note that this method is equivalent to sampling a whole circle with the tilt and allowing only for positive slants - a stimulus at $\phi = 180°$ and $\alpha = 50°$ looks similar to the stimulus at $\phi = 0°$ and $\alpha = -50°$

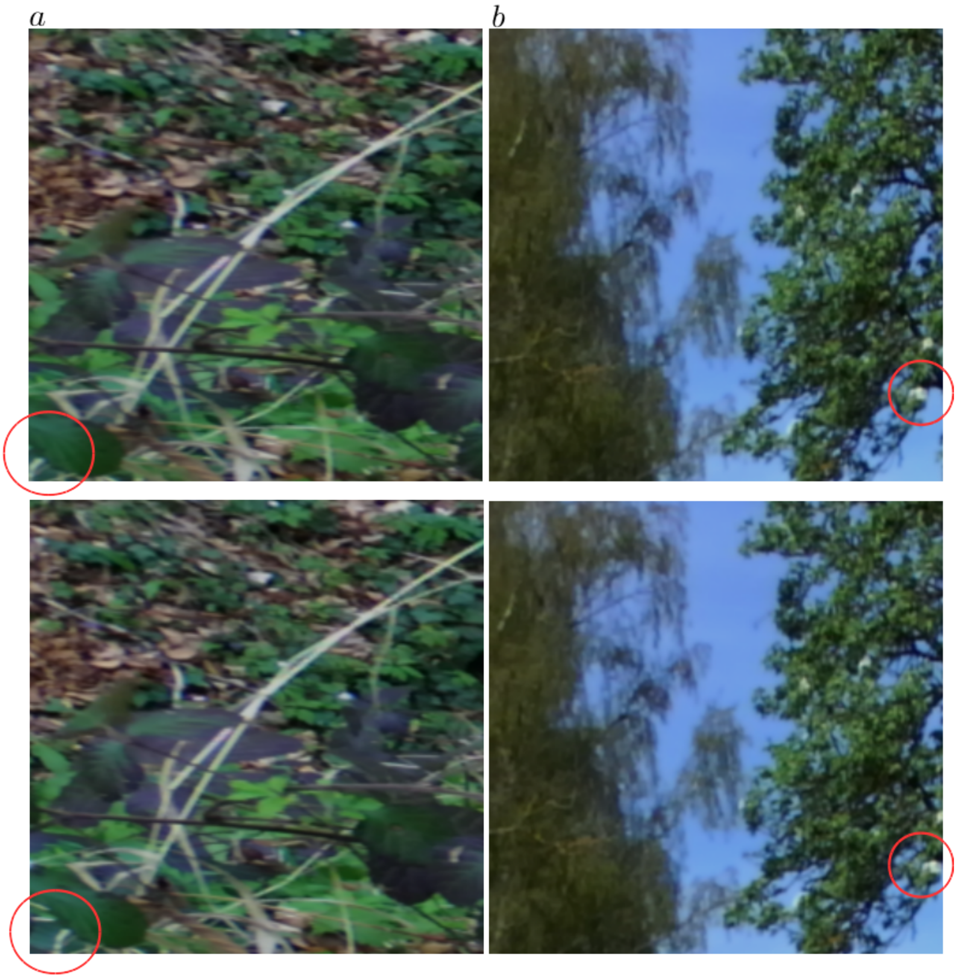


Figure 2.6: *a*: half-images at slant $\alpha = 55.2°$ and tilt $\phi = 170°$, *b* half-images at slant $\alpha = 38.25°$ and tilt $\phi = 80°$.
The red circles point out, easy-to-spot differences between the half-images

Shifted Stimuli

In addition, a set of shifted half-images was created to investigate inference of zeroth-order depth. Free parameters in this case are x-shift from −6px to 6px and y-shift from −6px to 6px, in half-pixel steps each, resulting in 625 possible parameter combinations. 50 images were shifted through all parameter combinations, making a total of 31250 half-images per eye.

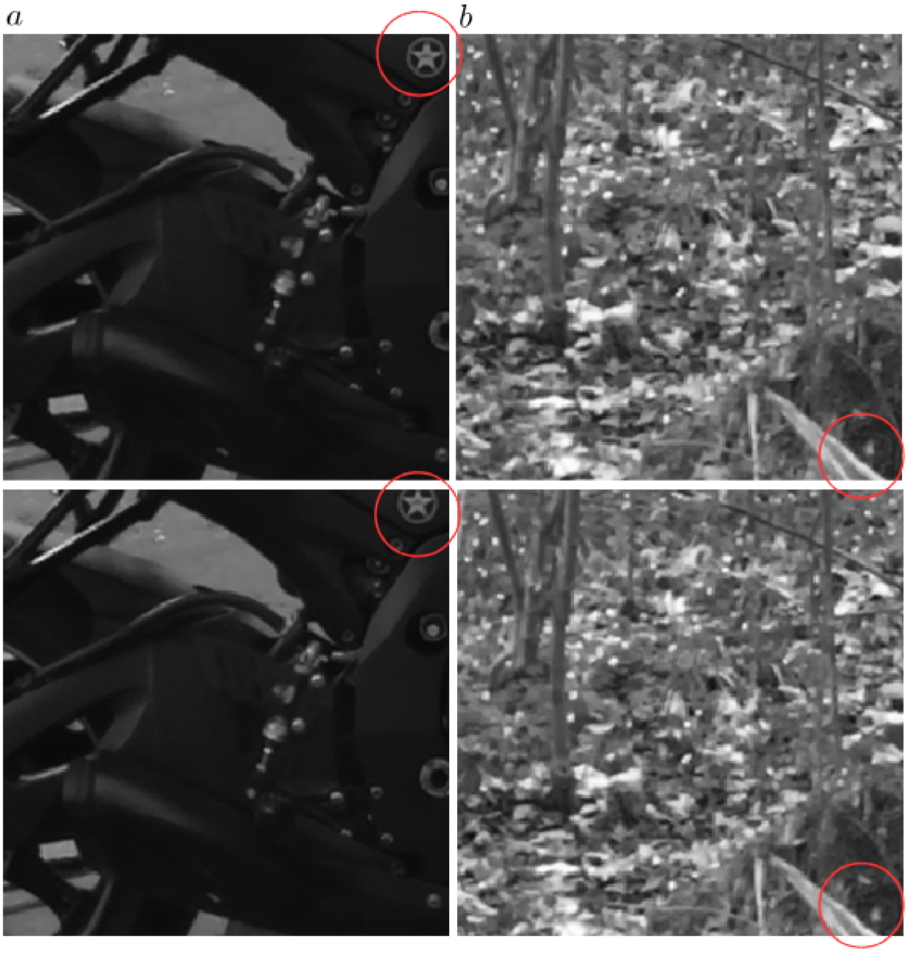


Figure 2.7: *a*: half-images at shifts $x = -3$ px and $y = 6$ px, *b* half-images at shifts $x = -5$ px and $y = 2.5$ px.
The red circles point out, easy-to-spot differences between the half-images

Note that the tilted/slanted stimuli are also converted to greyscale through the neural network processing. The processing of the shifted stimuli will not be specially mentioned: it occurs similar to the tilted/slanted stimuli

Neural Models

The neural model depicts the information processing from the retina to V1 - *how do neurons from the visual cortex fire, when presented a stimulus?*. This was achieved by employing a Sparse Convolutional Artificial Neural Network (SCANN) by (Lundquist et al., 2016) for binocular images. In addition, hard thresholding transfer functions for the network were implemented according to Rozell et al. (2008), modelling leaky integration and lateral inhibition (Schultz et al., 2014). The SCANN was implemented in the open-source network simulator PetaVision (*Petavision*, n.d.), which is optimized for parallel computing.

Structure

For this matter, the model seen in Figure 1.4 was extended, interposing the preprocessing by retinal cell-layers between image layer and V1 layer (see Figure 2.8). This mainly models the center-surround properties of receptive fields, as well as the overall whitening of the image representations (Abbasi-Asl et al., 2016; Atick & Redlich, 1992).
After preprocessing through the bipolar and ganglion layer, an edge-enhanced, decorrelated image representation is used for weight (basis functions) and feature map learning.
Feature map learning occurs according to Schultz et al. (2014).:

$$r_j = I_j - (\Phi z)_j \tag{2.6}$$

$$\frac{du_k}{dt} = -u_k + z_k + (\Phi^T r)_k \tag{2.7}$$

$$z_k = C(u_k) \tag{2.8}$$

with

$$C(u_k) = \begin{cases} u \text{ if } u \geq \lambda \\ 0 \text{ if } u < \lambda \end{cases} \tag{2.9}$$

Here r_j describes the residual layer, which holds the difference between the to-be-reconstructed and the momentary reconstruction of the image (see 1.8). u_k is the internal state of a neuron z_k. Equation 2.7 then describes the change of neuronal activity over time. $C(u_k)$ is the L_1-Norm (see equation 1.8).
Basis function learning occurs through gradient descent (Yue,2016).

Learning

In a first step, the basis functions need to be learned. This abstractly incorporates the already settled neural pathways for visual processing of an observer.

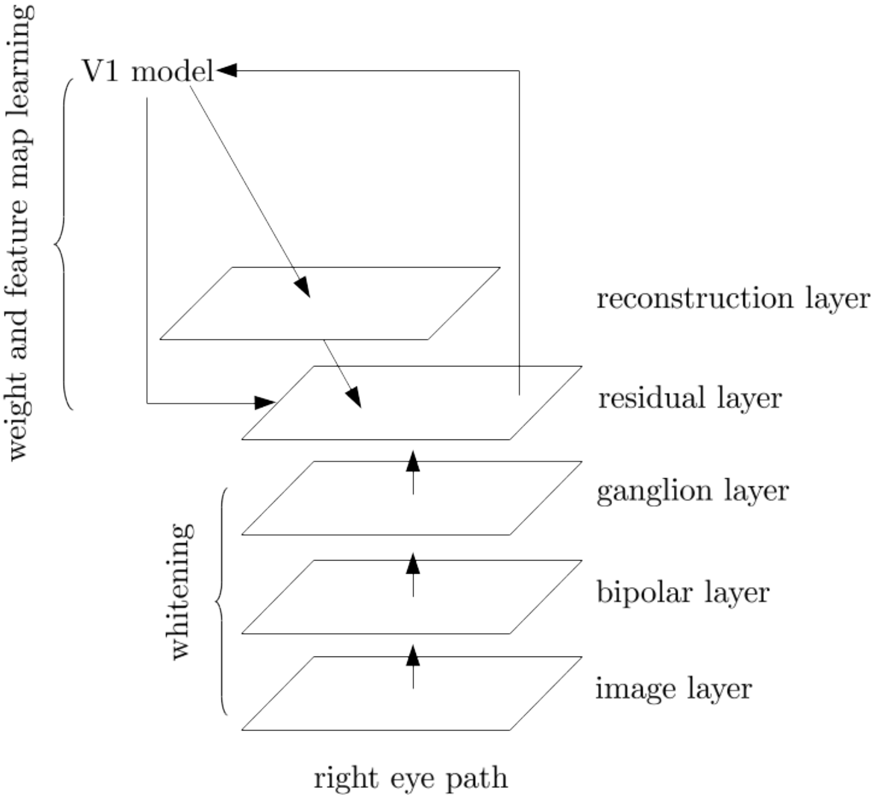


Figure 2.8: Information flow and processing along the neural network. While the first three layers preprocess the the data, in a similar fashion to the retinal layers, the upper loop-part of the network learns the feature maps, by incorporating equation 2.6.

Basis functions were learned on the virtual vergence database, created according to Reich (2017). The database spanned 73991 gray-scaled, 256×256-pixel images for each eye, at different vergence angles, dependent on image content. Width of the basis functions was set to 16 pixels, so that 16×16-pixel kernels emerged (in form of weights). The stride was set to $p_x = p_y = 8$ pixel. Images were presented in random order to evade bias. Every image was consecutively presented 150 times, leaving enough time for gradient descend to settle.

Sparsity factor λ was set to 0.1. Initial learning with a learning rate of 0.05, allowed the weights settle to the awaited Gabor-like shape. After qualitative validation, the learning rate was succesively reduced, to remove noise from the kernels.

The neural model did not need to run through all images, weights settled after approximately 10000 images.

Two neural models were learned at different overcompleteness factors of 1 and 8, resulting in sets of 128 and 1024 basis functions. The complete sets of basis functions are depicted in the appendix A.2.

Stimulus Presentation

In the second step, the created stimuli were presented in random order [3] with the set of learned basis functions at both overcompleteness factors. To reduce the duration of computation, stimulus presentations were parallelized to 80 batches, while every image was presented only 50 times. To account for the reduced number of presentations, the momentum parameter τ was reduced to 0.3. The reconstructions were then qualitatively validated, to assure good performance.
Activation of the V1 layer model was recorded into a PetaVision-specific pvp-container. The pvp-container holds the activations of the V1-layer at every 50 image presentions, ergo the last presentation of every image, in form of a sparse matrix.
Testing resulted in two such pvp-containers: The first, with activations from the V1 layer model of dimensions $(32 \times 32) \times 128$, the second with activations from the V1 layer model of dimensions $(32 \times 32) \times 1024$.
The procedure was similarly executed for the shifted stimuli, but only at an overcompleteness factor of 1.

Read-Out

To allow for later inference, every image's activation from the pvp-container needed to be reassigned to one of the 198 stimulus classes. In addition, only the assigned area under observation $m = 5$ or $m = 7$ is cut out from the V1 layer.

Matching Activations to Stimulus Classes

The read-out and matching algorithm was implemented in Matlab (Matlab R2017b), according to the following rules:

1. load the sparse activation matrix of image i.
2. reconstruct the sparse matrix to a full matrix of dimensions $(32 \times 32) \times J$, depending on the overcompleteness factor.
3. truncate the mid column of dimensions $(m \times m) \times J$ centered on the layer's center. Only a window of given size of every feature map z_j is thusly taken into accout.

[3] this allowed later sub-sampling of the presentations, with approximately the same stimulus class sizes.

4. reshape the matrix to a vector x_i of $m \cdot m \cdot J$ elements. Every element resembles an activation weight and thusly represents a neuron from the feature map of the former V1 layer.

5. read-out from the i-th line in the random presentation file-list, to which α, ϕ-combination x_i is related.

6. Let $S_{\alpha,\phi}$ be a 11×18-Matrix (α-levels $\times$ ϕ-levels), where each element in itself is a $(m \cdot m \cdot J \times 10050)$-Matrix $R^{\alpha,\phi}$ (length of x $\times$ number of stimuli in a stimulus class). x_i is then a column of $R^{\alpha,\phi}$.

7. re-iterate for all i .

The resulting matrix $S_{\alpha,\phi}$ then holds all V1-layer activations of all images' centered $m * p_x + 2 \cdot p_x \times m * p_y + 2 \cdot p_y$ pixel, sorted into the respective stimulus classes.
For example, at an overcompleteness factor of 1 and $m = 5$, $S_{-55.2°,0°}$ holds a 3200×10050 matrix $R^{-55.2°,0°}$. Hence, each of the 10050 columns represents neural activation for the presentation of one surface, slanted at $\alpha = -55.1963°$ around the tilt axis at $\phi = 0$, equivalent to $v = [\cos(0), \sin(0), 0]' = [1, 0, 0]'$, ergo the x-Axis. In a column, every $m \cdot m$ elements correspond to one feature map and thus to the same kernel, only at different positions. The first element of the first column at $S_{-55.2°,0°}$, therefore corresponds to an activation from z_1, more precise: it is the result after convoluting the first image from stimulus class $(-55.1963°, 0°)$ with with the first kernel at position $x = 112$ px, $y = 112$ px.

Tuning Maps

Definition

With sorted activations, it is now possible to build tuning maps. Tuning maps are an agglomeration of the data from $S_{\alpha,\phi}$ to visualize, to which form of stimuli a certain neuron is sensitive. Being sensitive means, that the neuron is active, while stimuli of a certain α, ϕ-combination are presented and remains inactive for stimuli of other α, ϕ-combinations. In light of the sparse coding approach, being active is modelled as having a non-zero activation weight in z_j. Because every voxel $v_{m,n,j}$ in the V1-layer stands for the activation weight of one neuron i, it is sufficient to count all non-zero entries (denoted as $r_{i,j}^{\alpha,\phi}$) across the ith row of all $R^{\alpha,\phi}$. In other words, this counts for how many of the 10500 stimuli per stimulus class, neuron i was active. A tuning map T^i (with elements $t^i_{\alpha,\phi}$) for neuron i is defined as:

$$t^i_{\alpha,\phi} = \sum_{j}^{|C_k|} A(r_{i,j}^{\alpha,\phi}) \tag{2.10}$$

with

$$A(x) = \begin{cases} 1 \text{ if } x > \lambda \\ 0 \text{ if } x = 0 \end{cases} \tag{2.11}$$

$A(x)$ is a decision function, which omits the strength of activation of neuron i and only considers if a neuron is active or not. Hereby λ is set to 0.12, resembling the hard threshold from equation 2.6 (Rozell et al., 2008).

To obtain the tuning map for neuron number three for example, one has to count the non-active elements of the third rows of all $R^{\alpha,\phi}$.

Presupposing, that a neuron is equally-likely to be active for all image-presentations of a stimulus class C_k, the likelihood of that neuron being active for one image-presentation of C_k (an elementary event) can be calculated according to a simple LaPlace experiment:

$$p(neuron = 1|C_k) = \frac{1}{|C_k|} \tag{2.12}$$

The tuning map condenses many such elementary events to one event for every C_k. This compound event's likelihood is then:

$$p(t^i_{\alpha,\phi}|C_k) = \frac{t^i_{\alpha,\phi}}{|C_k|} \tag{2.13}$$

By normalizing T_i, every element therefore shows the likelihood of neuron i to fire at a stimulus class.

Visualization

In section 2.1.1 I described that, to create stimuli, a surface is slanted and tilted, in respect to an observer. This equivalently can be interpreted, as an observer looking from a certain position on a fixed surface. Also, negative slants in the tilt interval $[0°, 180°)$, are equivalent to the same positive slant in the tilt interval $[180°, 360°)$. According to these equivalencies, the stimulus classes are spread around a half-sphere laying above the surface, depicted in Figure 2.9. This interpretation of stimulus presentation, leads to the tilt-data, being circularly distributed.

Depicted tuning maps are then the envelope of the half-sphere. Figure 2.10 shows an exemplary tuning map. Note that, while the data is doubled for $\alpha = 0°$ and $\phi \in [180°, 350°]$, to allow a continuous display of the tuning map, the *whole row* for slant $\alpha = 0°$ is one point on the sphere, ergo one stimulus class, directly above the stimulus.

This would change the number of stimulus classes from 198 to 181 (as the eighteen $\alpha = 0$ stimulus classes, collapse to one). Because of this asymmetry in stimulus class distribution, stimulus class $\alpha = 0$ will be omitted for later inference.

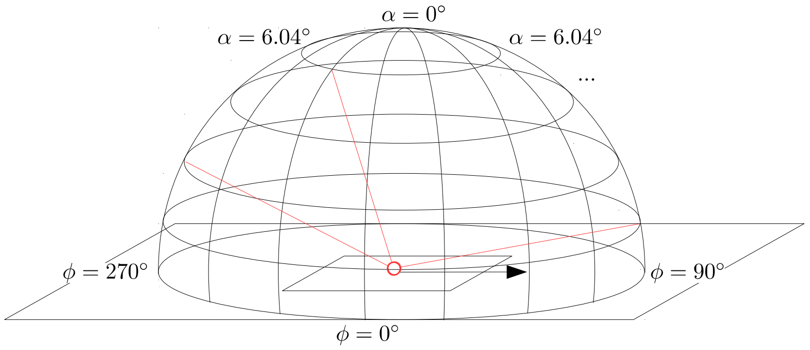


Figure 2.9: Similar to moving the stimulus in respect to the observer, one can interpret a moving of the observer in respect to the stimulus. The small surface inside the half-sphere depicts the stimulus. The half-sphere covers the observer's position in respect to the stimulus for all stimulus classes. The three red lines depict three exemplatory observer positions. The black arrow depict the tilt-axis at $\phi = 90°$.

Further Processing

Figure 2.10 also shows, that the likelihood is not continuous across the tuning map: it seems noisy. Bosking (2008) however, points out that (especially for orientation) neural selectivity is continuous. To encounter this, the raw tuning maps are smoothed. For this matter I employed a Savitzky-Golay Filter (Savitzky & Golay, 1964), implemented in Matlab by (Huang, n.d.). By fitting a third degree polynomial (by method of least squares) on 7×7 subsquares of the tuning map, the signal-to-noise ratio can be increased. Figure 2.11 shows the same tuning map from Figure 2.10, but smoothed.
The smoothing procedure is applied to all tuning maps.
Again, the number of neurons used for inference $m \cdot m \cdot J$, is impacted by both, area under observation ($m = 5$, $m = 7$ and the overcompleteness factor of 1 or 8). Consequently, for every condition, one set of neurons, with their tuning maps is computed, with sizes depicted in Table 2.12.

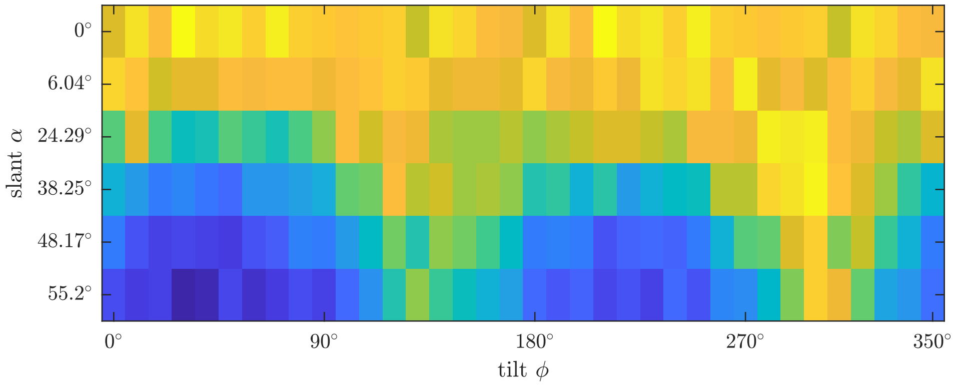


Figure 2.10: A raw tuning map, which incorporates the number of being active of one neuron, for all stimulus classes.

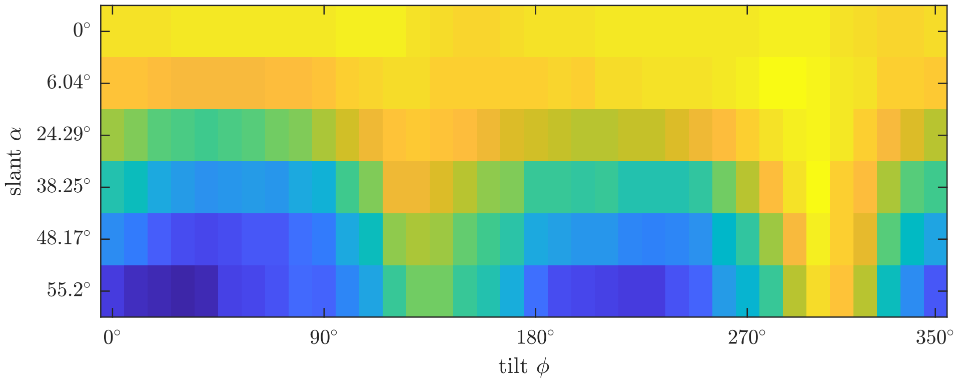


Figure 2.11: The same tuning map as in Figure 2.10, but smoothed. Transitions between different stimulus classes are more continuous.

Depth Inference using a Naive Bayes-Classifier

Through a neuron's tuning map, its activity behaviour in respect to the stimulus classes is known. If a neuron fires at an image presentation, that should mean, that image belongs to one of the stimulus classes with high likelihood in the tuning map. However, single tuning maps are too unspecific, to allow any exact stimulus class inference.

As mentioned in section 1.4, an image is encoded by a linear combination of binocular basis functions. Every basis function pair thereby stands for the receptive fields of one neuron. Taking the tuning maps of those neurons, that encode one image, should enable an inference of the stimulus class of the presented image: every tuning map cuts out stimulus classes with small likelihoods and reinforces stimulus classes with big likelihoods. This interaction can be implemented by multiplying the tuning maps element-wise. As soon as one of the encoding neurons contradicts a stimulus class,

Area Under Observation	Overcompleteness Factor	
	1	8
5×5	3200	6272
7×7	25600	50176

Figure 2.12: Different number of neurons in the neuronal sets for the four conditions of the experimental paradigm. Choosing $m = 7$ roughly doubles the number of neurons taken into account for estimation, in contrast to $m = 5$.

its low likelihood value enters the multiplication term, strongly weakening the result. The image belongs to the stimulus class C_k, where the likelihood-product is the biggest.

This train of thought is perfectly formalized through the naive Bayes-classifier. The general classifier (Murphy et al., 2006) is described by:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\arg\max}\; p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.14}$$

In light of my work, n stands for the number of neurons, C_k stands for the stimulus classes, with $k = 180$ and $p(x_i|C_k)$ is the likelihood of a neuron x_i to fire at C_k. As every stimulus class is equally probable at image presentation, $p(C_k)$ is a uniform distribution and can be omitted. The naive Bayes-classifier is then equivalent to a Maximum-Likelihood classifier.

A tuning map encompasses the likelihoods of a neuron to be active, for all stimulus classes. $T^i_{Neg} = 1 - T^i$ then describes the likelihoods of a neuron to remain non-active, for all stimulus classes.

All tuning maps are logarithmized. This not only simplifies computation, by allowing to add the tuning maps, it also prevents possible integer overflow. If an image is encoded by j neurons, the classifier can then be rewritten as:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\arg\max} \sum_{active} \ln(T^i) \tag{2.15}$$

However, to exactly follow the formalism of the naive Bayes Classifier, *all* neurons must enter the sum:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\arg\max} \sum_{active} \ln(T^i) + \sum_{inactive} \ln(T^i_{Neg}) \tag{2.16}$$

$$= \underset{k \in \{1,\ldots,K\}}{\arg\max} \sum_{i}^{n} \ln(T^i_{Neg}) - \sum_{active} \ln(T^i_{Neg}) + \sum_{active} \ln(T^i) \tag{2.17}$$

By rewriting equation 2.16 into 2.17 a further simplification of computation can be achieved. Let the term $\sum_i^n \ln(T^i_{Neg})$ be the *bias*: the summed log-likelihoods of all neurons to stay inactive, across all stimulus classes. $\hat{y}$ is

the position (α, ϕ) of the maximum log-likelihood in the summed log-tuning maps.
Stimulus class inference will be once tested according to equation 2.15 and once according to equation 2.17.

Measures for Goodness of Inference

To evaluate how well stimulus classes are inferred and to allow for comparison between the different conditions of the experiment, some measures of accuracy and precision of the estimator $\hat{y}$ are introduced.

Tilt

ϕ follows a circular distribution and therefore requires special statistical treatment. If the estimation for one image results in $\phi = 350°$ and another estimation results in $\phi = 10°$, then their mean is not $\frac{350°+10°}{2} = 180°$, but it is $0°$. For this matter I used the circular statistics toolbox for Matlab (Berens et al., 2009).
According to Fisher (1995), estimations of ϕ are construed as vectors with a length R_p and a direction T_p. A whole sample of n estimations then allows to compute the mean direction T_l of the sample, as well as the resultant length R_l: by adding all sample vectors, a long, new vector arises. The mean resultant length $\bar{R}_l = \frac{R_l}{n}$, then also carries information about variance. If direction among the vectors is diverse, R_l is shorter. Consecutively, the circular Variance is defined by $V = 1 - \bar{R}_l$ and the circular standard deviation by $v = \sqrt{-2\ln(\bar{R}_l)}$. Note that while V is in the interval $[0, 1]$, v can (theoretically) get infinitely big as $\bar{R}_l$ converges to 0.
For a measure of precision the circular standard deviation will be used. For a measure of accuracy, mean signed error (the first error momentum) $MSD = T_l - T_{\text{ground truth}}$ will be used.

Slant

At first glance α also seems to follow a circular distribution. However what is taken into account at estimation is only a small interval of $[0°, 55.2°]$, with underlying unit-disparities, following linear sampling. Therefore the normal statistical approach suffices: as for a measure of precision standard deviation will be employed and for accuracy the mean signed error.

Testsets

To be able to assess the goodness of measurement, furthermore test-data is needed. For this matter, at tilt/slant stimulus creation (2.1.1), additional 1000 images for every of the 180 stimulus classes, were created. The test stimuli too, were fed to the SCANN, so that the activations from the neural model could be recorded. From the activations, the stimulus-encoding neurons can be read out, so that the Bayes Classifier could be employed. Similarly an additional set of 50 shifted stimuli for all 625 stimulus classes were created, presented and recorded.

Chapter 3

Results

Stimulus class inference is solely based on which neurons encode a stimulus and on the quality and behaviour of the neurons' tuning maps. Therefore first, differences and similarities across the four sets of neurons (according to 2.12) in respect to *all* stimuli will be evaluated.
Next, the interaction between a neuron's position in the feature map and its receptive field will be assessed.
Finally, the performance of stimulus class inference will be evaluated in respect to tilt and slant. A short section will point out, how the likelihood approach performs for zeroth order depth. For matters of simplicity overcompleteness factor 1 and overcompleteness factor 8 will be referred to as O1 and O8.

Neuronal Activity across all Stimuli

Every stimulus is encoded by a few neurons (see equation 1.1). Through sparsity, the number of encoding neurons is kept low. Figure 3.1 shows the mean number of encoding neurons at all stimulus classes for all four conditions.
The mean number of encoding neurons is doubled for the $m = 5$ conditions, to be able to display all conditions on the same scale. This does not distort the results, since the $m = 5$ conditions have roughly half the neurons of the $m = 7$ conditions (respectively for O1 and O8).
For low slants ($\alpha = 0°, \alpha = 6.04°, \alpha = 24.29°$), stimuli are consistently encoded by around the same number of neurons for every condition, respectively. Stimuli around $\phi = 60°$ and $\phi = 240°$, especially at the highest slant of $\alpha = 55.2°$ are encoded by the most neurons and seem to systematically deviate from the mean.
The mean number of encoding neurons of low slants ($\alpha = 0°, \alpha = 6.04°, \alpha = 24.29°$), is significantly different from the mean number of encoding neurons of high slants ($\alpha = 38.25°, \alpha = 48.17°, \alpha = 55.2°$), for all four conditions

(one-tailed, two-sampled t-test with unequal variance). The exact statistical procedure can be found in the appendix (A.4.1).

Figure 3.2 shows how many stimuli per stimulus class remained unencoded. This means, that no neuron was active (within the area under observation) at stimulus presentation. Note, that no doubling occured for the conditions $m = 5$.

For the lower four slant levels, around 1000 images could not be encoded (corresponding to about 10% of stimuli of a stimulus class). For slants of $\alpha = 48.17°$ and $\alpha = 55.2°$, especially around tilt $\phi = 60°$ and $\phi = 240°$, up to 2500 stimuli (25 % of the stimuli from a stimulus class) remained unencoded. At the highest slant level and $\phi = 150°$ or $\phi = 320°$ a smaller rise of unencoded images can be observed.

The stimulus classes with the most unencoded images, coincide with the stimulus classes, where stimuli were encoded by the most neurons.

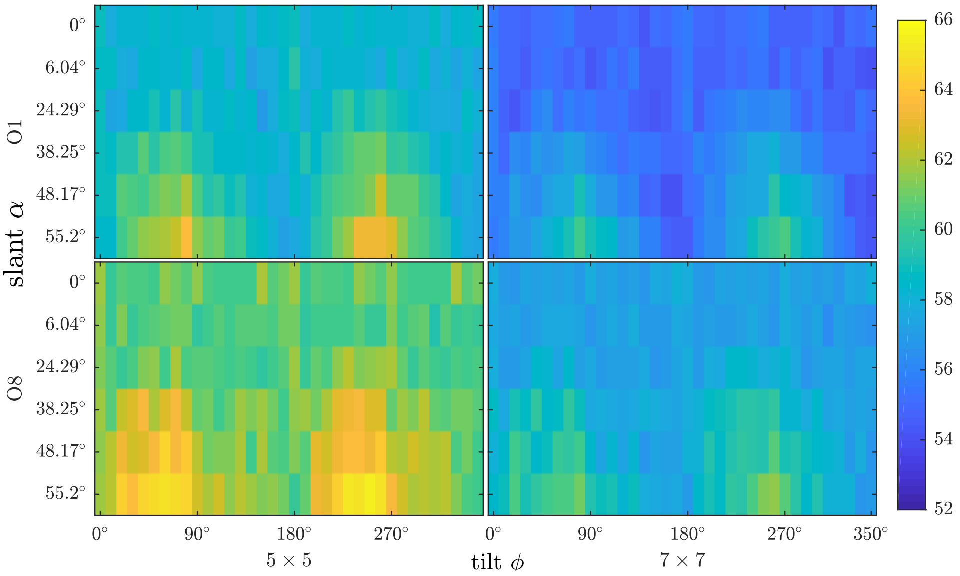


Figure 3.1: Mean number of encoding neurons per image, per stimulus class. Depicted are all four neuronal sets: *left, top*: O1, $m = 5$; *right, top*: O1, $m = 7$; *left, bottom*: O8,$m = 5$; *right, bottom*: O8,$m = 7$.

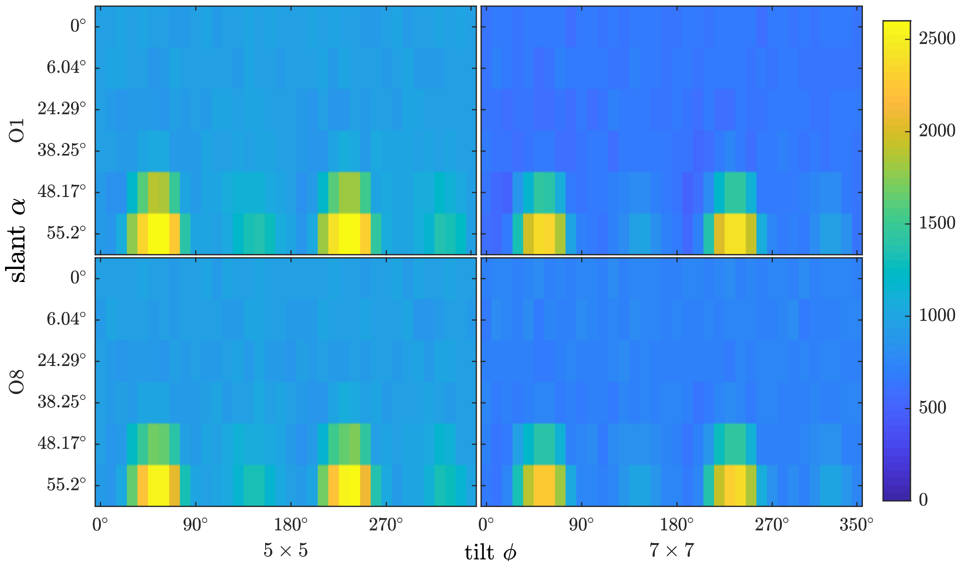


Figure 3.2: Number of images, that evoked no neuronal activation, per stimulus class. Depicted are all four neuronal sets: *left, top*: O1, $m = 5$; *right, top*: O1, $m = 7$; *left, bottom*: O8,$m = 5$; *right, bottom*: O8,$m = 7$.

Tuning Maps

O1, m=5

The underlying V1 model for this condition spanned 128 feature maps $z_1 ... z_{128}$. From this V1 layer only a centered column of 5×5 neurons is cut out. Although every neuron, associated with one z_i, acts individually, all 25 neurons from the same z_i have the same receptive field, by definition. Differences in tuning maps only emerge from the position within their feature map.

In total this set contains 3200 neurons. Figure 3.3 shows two tuning maps.

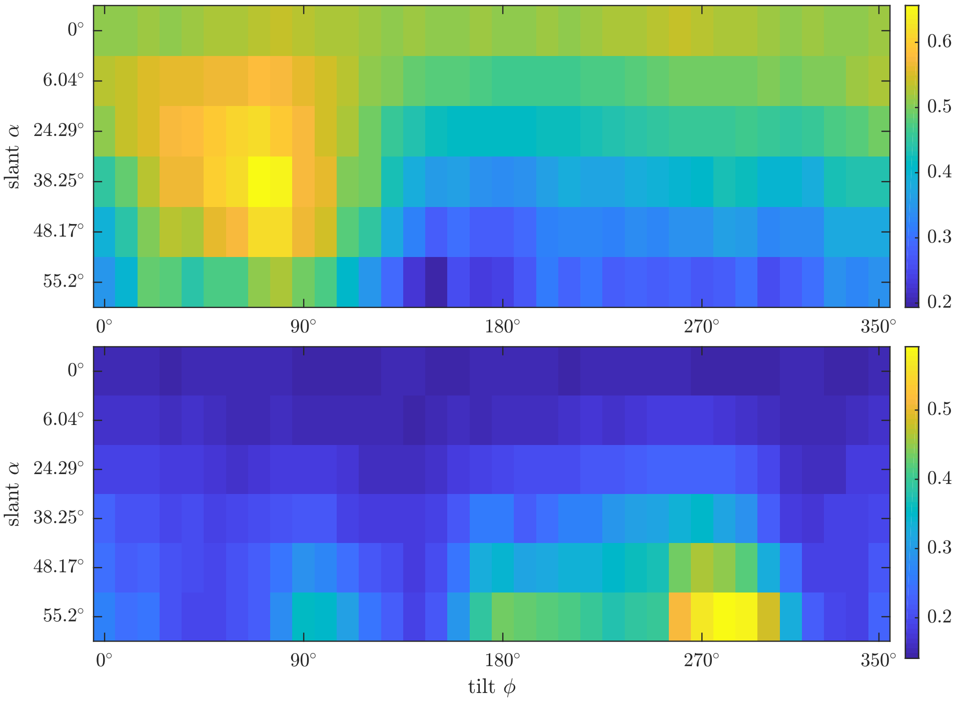


Figure 3.3: Two exemplary tuning maps from the neuronal set O1,$m = 5$. Color shows % likelihood to fire at a stimulus class.

Clear selectivity for certain stimulus classes can be seen. To be able to further assess this set of neurons, every neuron is associated with the stimulus class, where it has the highest probability, to be active to.

Figure 3.4 (upper row, blue) depicts the distributions of the favored stimulus class for all neurons, across slant α and tilt ϕ.

By far the most kernels favor $\alpha = 55.2°$, whereas almost no kernels are selective for the three small slant levels. Even $\alpha = 38.25°$ and $\alpha = 48.17°$ are seldomly preferred.

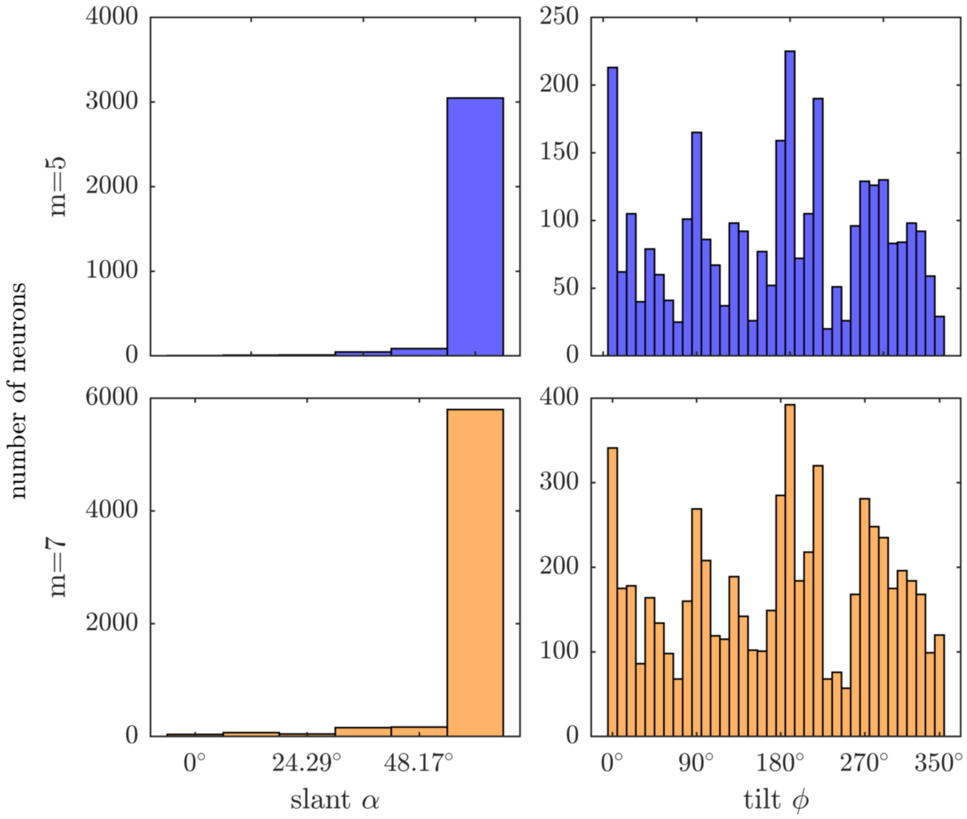


Figure 3.4: Histograms of the distributions of favored stimulus classes in tuning maps. Depicted are two neuronal sets: *top*: O1,$m = 5$; *bottom*: O1,$m = 7$; *left*: favored slant α; *right*: favored tilt ϕ.

Selectivity for tilt is more uniformly distributed. Still, a clear favor for $\phi = 0°$ and $\phi = 180°$ is visible. Orthogonal tilts thereto, at $\phi = 90°$ and $\phi = 270°$ also show peaks. $\phi = 220°$ breaks this scheme. Tilts around $\phi = 70°$, $\phi = 150°$, $\phi = 230°$ and $\phi = 350°$ almost completely lack selective neurons. Especially the contrast between $\phi = 0°$ and $\phi = 350°$ or $\phi = 220°$ and $\phi = 230°$, substantiates the discontinuity of the distribution of maximum likelihood across tilts.

The mean tuning map (Figure 3.5, at the top) mirrors the strong selectivity for the tilts in cardinal directions ($\phi = 0°$, $\phi = 90°$, $\phi = 180°$ and $\phi = 270°$). Additionally, nearly no neurons are availible, that encode tilts around $\phi = 50°$ and $\phi = 230°$ at the highest slant level. The lack of neuronal selectivity for small slants is also mirrored.

O1, m=7

This condition takes into account the centered column of 7 × 7 neurons from the V1 model with 128 feature maps: 3200 neurons are similar to the $m = 5$ condition, while 3072 new, more excentric neurons are added, totalling in 6272 neurons. In this case, every 49 neurons share the same receptive field, being from the same z_i.

The additional neurons don't have an impact on the distributions of favored tilts and slants, as can be seen in Figure 3.4 (bottom row, orange). In total however, more neurons are on-hand for low slant levels. Some discontinuities for tilts are less steep, for example at $\phi = 20°$ or $\phi = 230°$.

Consequently, the mean tuning map remains nearly unchanged (Figure 3.5, at the bottom).

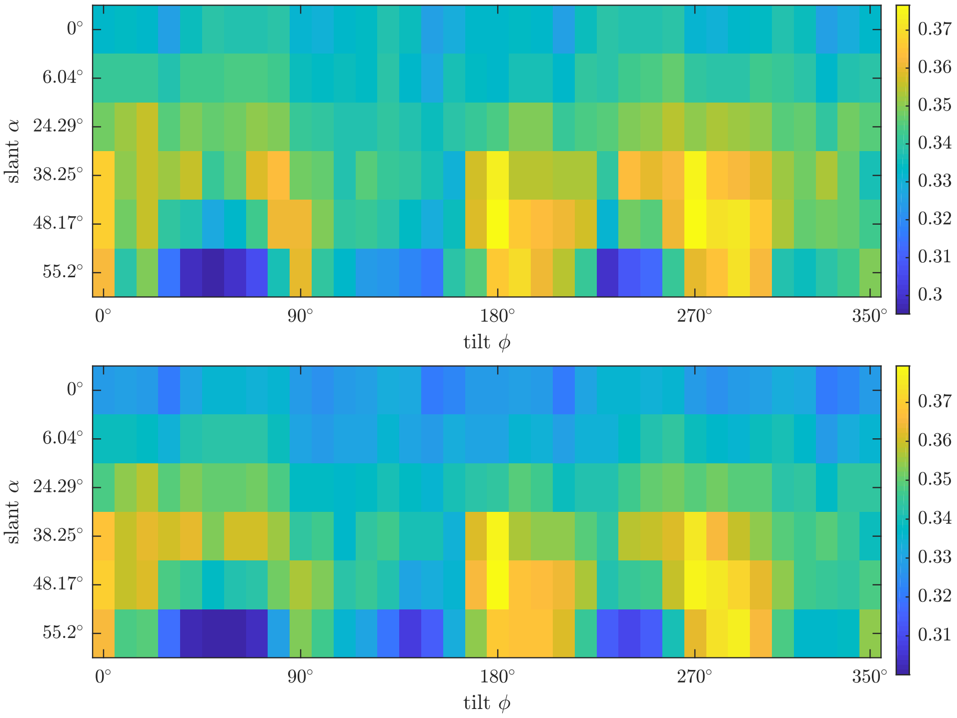


Figure 3.5: Mean tuning maps of two neuronal sets. Color shows % likelihood to fire at a stimulus class. *top*: O1,$m = 5$; *bottom*: O1,$m = 7$

Figure 3.6 shows the distributions of the maximum likelihood value (strength of activation at the favored stimulus class) across the two neuronal sets. The additional neurons from the $m = 7$ set, follow the same bimodal distribution as the neurons from the $m = 5$ set.

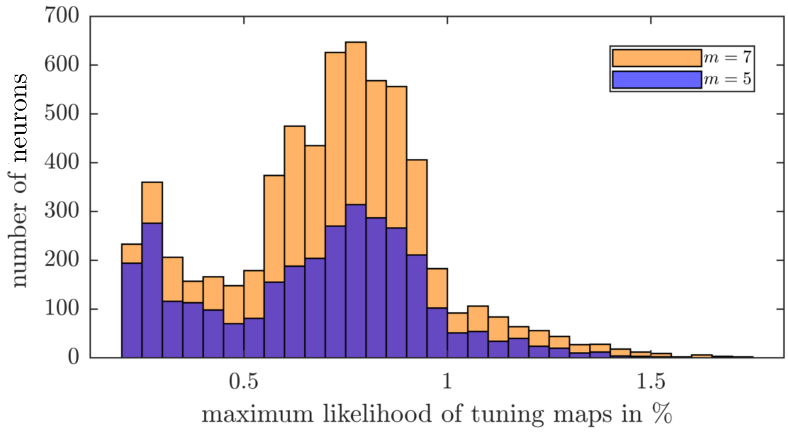


Figure 3.6: Histogram of the distribution of the magnitude of maximum likelihood for two neuronal sets. Bin size is set to 0.05.

O8

The underlying V1 model for these conditions spanned 1024 feature maps. The resulting neuronal sets again, stem from the centered 5 × 5 column (25600 neurons) and the 7 × 7 column (50176 neurons).
Figure 3.7 shows four exemplary tuning maps from both conditions, which are almost indiscernible from one another. In fact many tuning maps from both conditions look nearly identical: a general high selectivity for the two lowest slant levels. The higher the slant level, the more selective the neurons become for the tilt at $\phi = 0°$ and $\phi = 180°$.
This is further corroborated by the mean tuning maps (Figure 3.8, $m = 5$ at the top, $m = 7$ at the bottom). Similar to the O1 conditions, no neurons are selective for tilts around $\phi = 50°$ and $\phi = 230°$ at the highest slant level. Figure 3.9 (on the right side) depicts, that for many tilt levels there are no selective neurons at all. Almost all tuning maps favor $\phi = 0°$, $\phi = 180°$ and $\phi = 350°$. The shape of the distributions of tuning map preference looks alike between the two O8 conditions, although O8, $m = 7$ has more tuning maps favoring $\phi = 180°$ in respect to $\phi = 0°$ and $\phi = 350°$. The high peak for $\alpha = 38.25°$ on the left two histograms is also visible in the mean tuning maps.

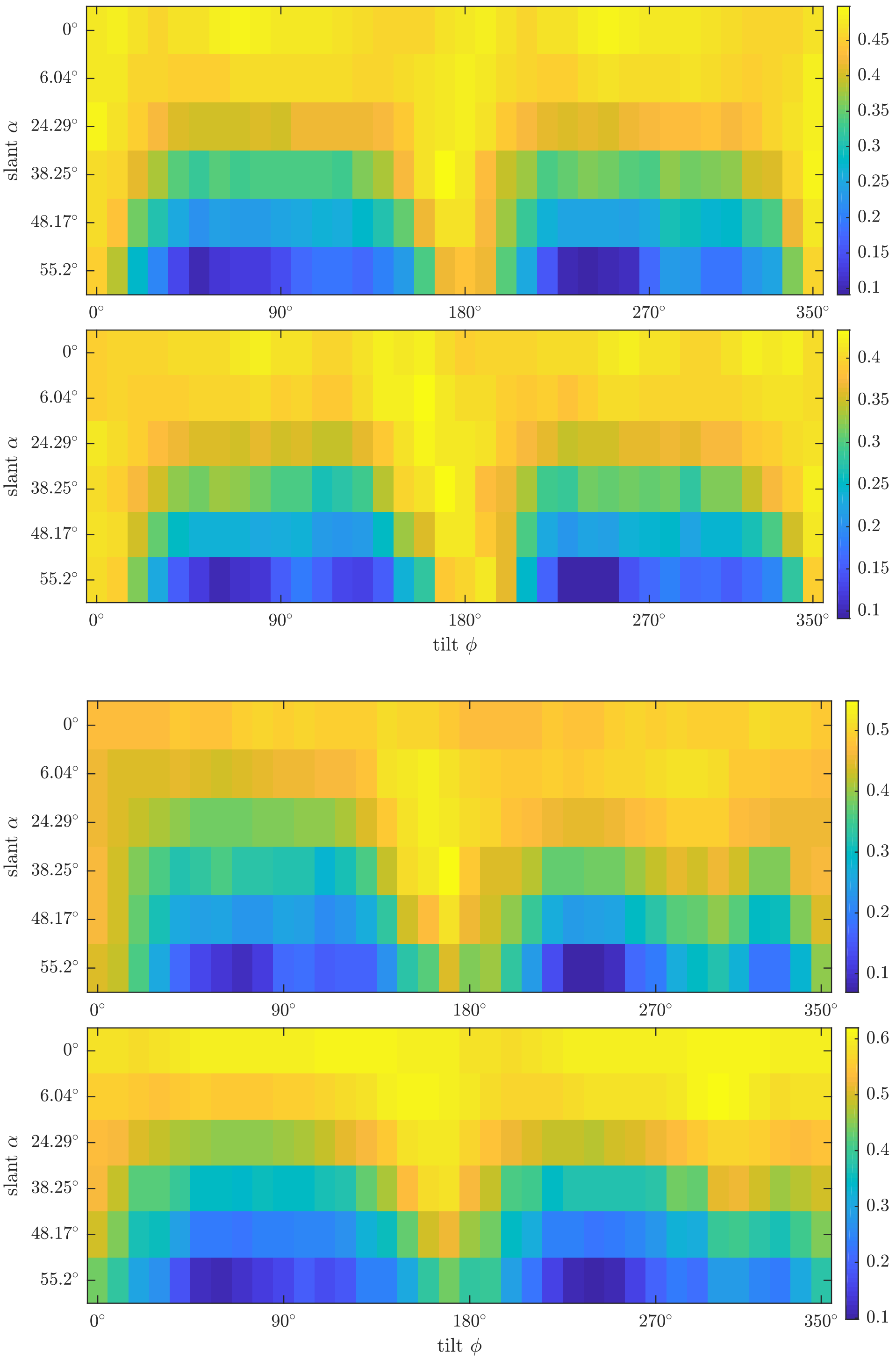


Figure 3.7: Four exemplary tuning maps from two neuronal sets. Color shows % likelihood to fire at a stimulus class. *top two*: O8,$m = 5$; *bottom two*: O8,$m = 7$.

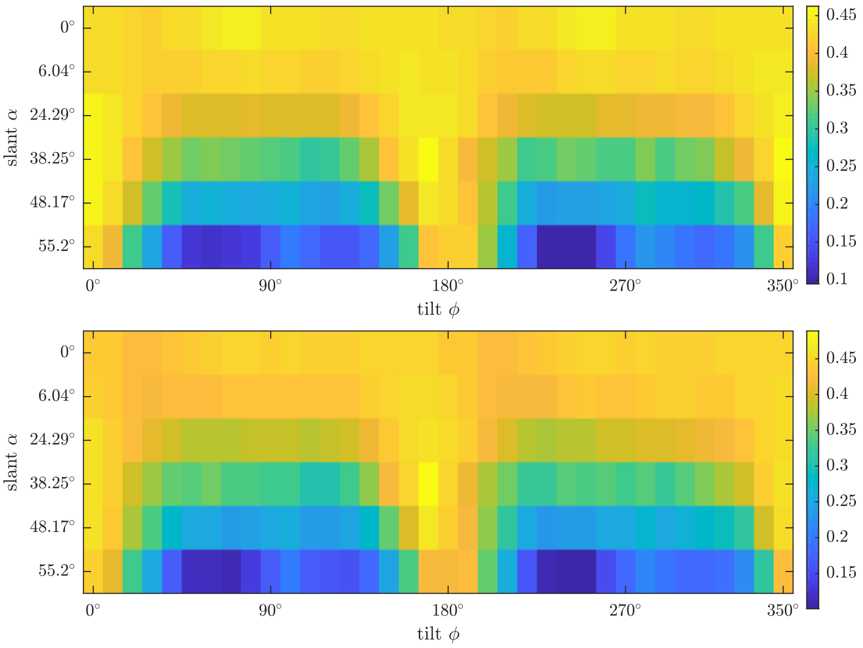


Figure 3.8: Mean tuning maps of two neuronal sets. Color shows % likelihood to fire at a stimulus class. *top*: O8,$m = 5$; *bottom*: O8,$m = 7$

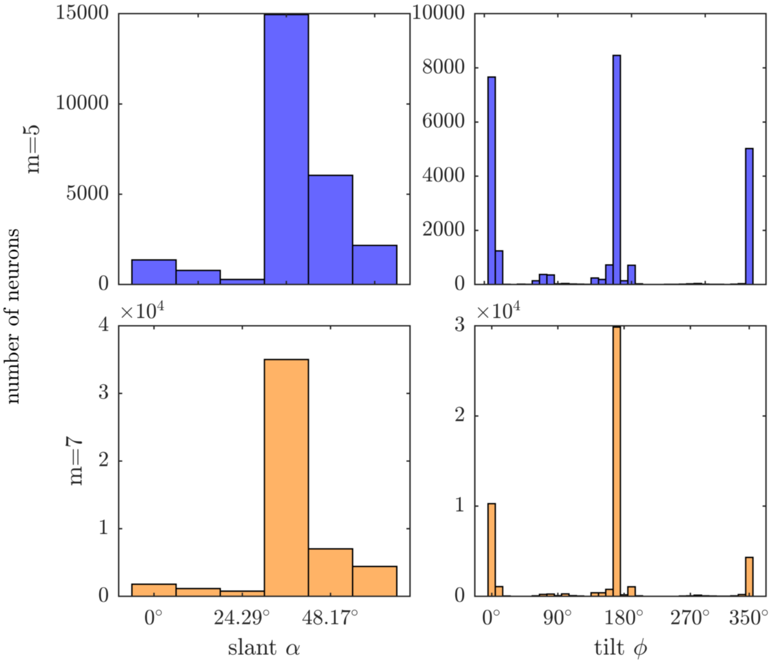


Figure 3.9: Histograms of the distributions of favored stimulus classes in tuning maps. Depicted are two neuronal sets: *top*: O8,$m = 5$; *bottom*: O8,$m = 7$; *left*: favored slant α; *right*: favored tilt ϕ.

Diversity of Tuning Maps

Figure 3.10 shows the Frobenius Norm

$$\|A\|_F := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{3.1}$$

of every tuning map in respect to its mean tuning map: $A = T^i - T_{mean}$. The Frobenius norm is a suitable measure for difference between two $M \times N$-Matrices. In the Figure tuning maps are thereby only divided after their overcompleteness factor.

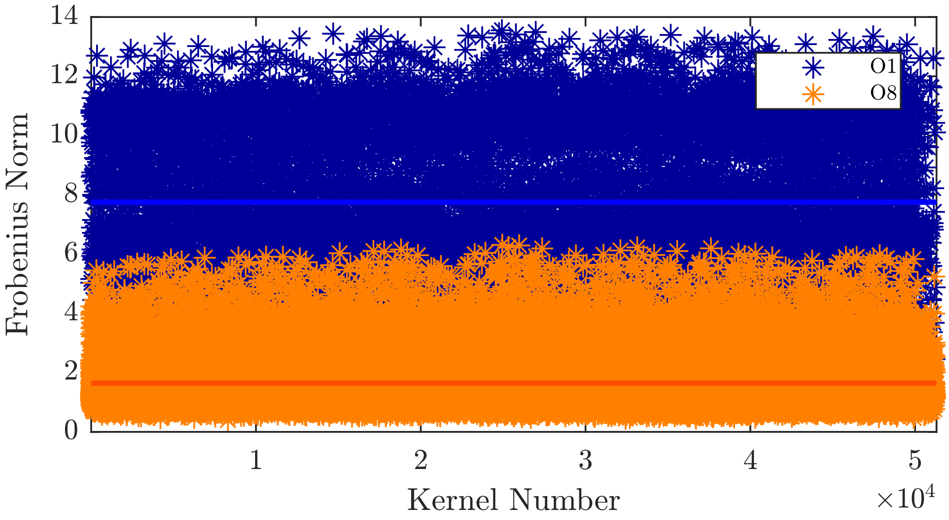


Figure 3.10: Magnitude of Frobenius Norm of all tuning maps in respect to their mean tuning map. Color shows membership in O1 or O8. Horizontal lines depict the mean norm of the tuning maps of the respective overcompleteness factor.

Indeed, the O8 conditions are throughout similar to one another, with a mean distantance of under 2. The O1 conditions, on the contrary, show bigger differences in respect to their mean, having a mean norm around 8.

Intermission

For the O8 conditions it is already possible to mention, that inference is impossible. With such similar tuning maps, the whole train of thought, that every tuning map cuts out stimulus classes, which are unlikely, fails. Irrespecive of which neurons encode an image, the resulting estimation would be the same. For further analysis I will omit the O8 cases, as their results can be summarized in one sentence: they all estimate the same.

Selectivity Dynamic of Neurons in the same Feature Map

Depending on where a neuron is positioned in a feature map, its receptive fields observe different patches of the stimulus. Neurons in the middle of a feature map, for example, cannot tell anything about slant, due to the fact, that their receptive fields lay above the focus point. By definition, the only information, they might encode, stems solely from tilt. Neurons from the excentric parts of the feature maps, on the other hand, receive input of high disparity: they should be able to give information about slant.

Figure 3.11, Figure 3.12 and Figure 3.13 show the dynamic of selectivity with the aid of five exemplary neurons from the same feature map.

The neuron at the left-most, bottom position (Figure 3.11,upper image) is strongly selective around $\phi = 110°$ and $\alpha = 48.17°$. A weaker selectivity (in green) describes an arc-like structure, along $\alpha = 24.29°$, up to $\phi = 330°$. Remind the circular structure of the depiction: the left border of tuning map is connected to the right border. The core of the arc shows almost no selectivity. The structure is reminiscent of ON-OFF structures of ganglion-receptive fields from V1.

Looking at the right-most, bottom neuron (Figure 3.11, image below), the strong selectivity wandered along the arc-structure to $\phi = 270°$ and $\alpha = 55.2°$. In addition the whole arc-structure is shifted anti-clockwise for about 60°.

The left-most, upper neuron (Figure 3.12, upper image) shows strong selectivity around the same stimulus classes, like the neuron on the left-most, bottom position. The arc of weaker selectivity however, has flipped, extending along $\alpha = 38.25°$ to $\phi = 180°$.

The right-most, upper neuron (Figure 3.12, image below) similarly shares its selectivity with the right-most, bottom neuron, while the weaker arc selectivity is shared with its left counterpart.

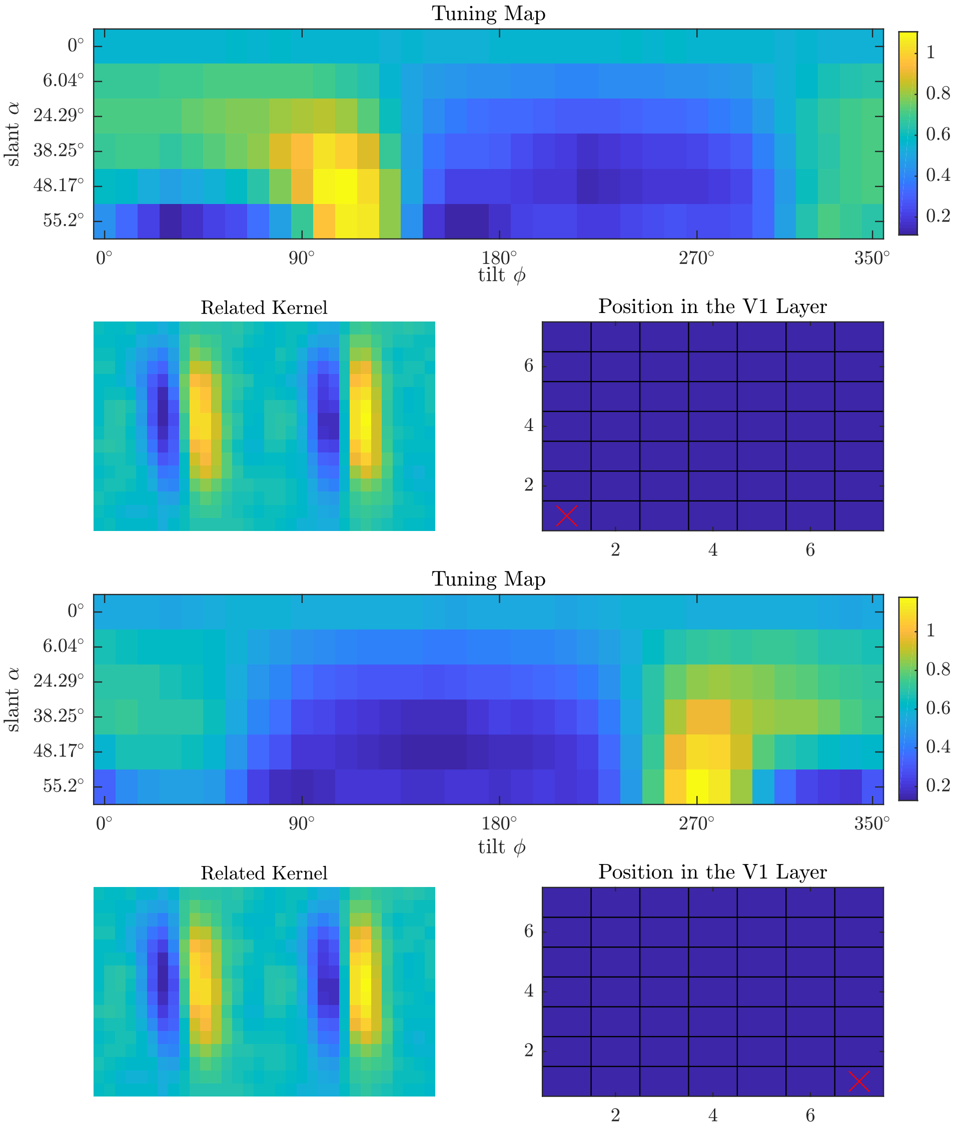


Figure 3.11: Two tuning maps from different corner positions of the same feature map with the depicted underlying basis function.

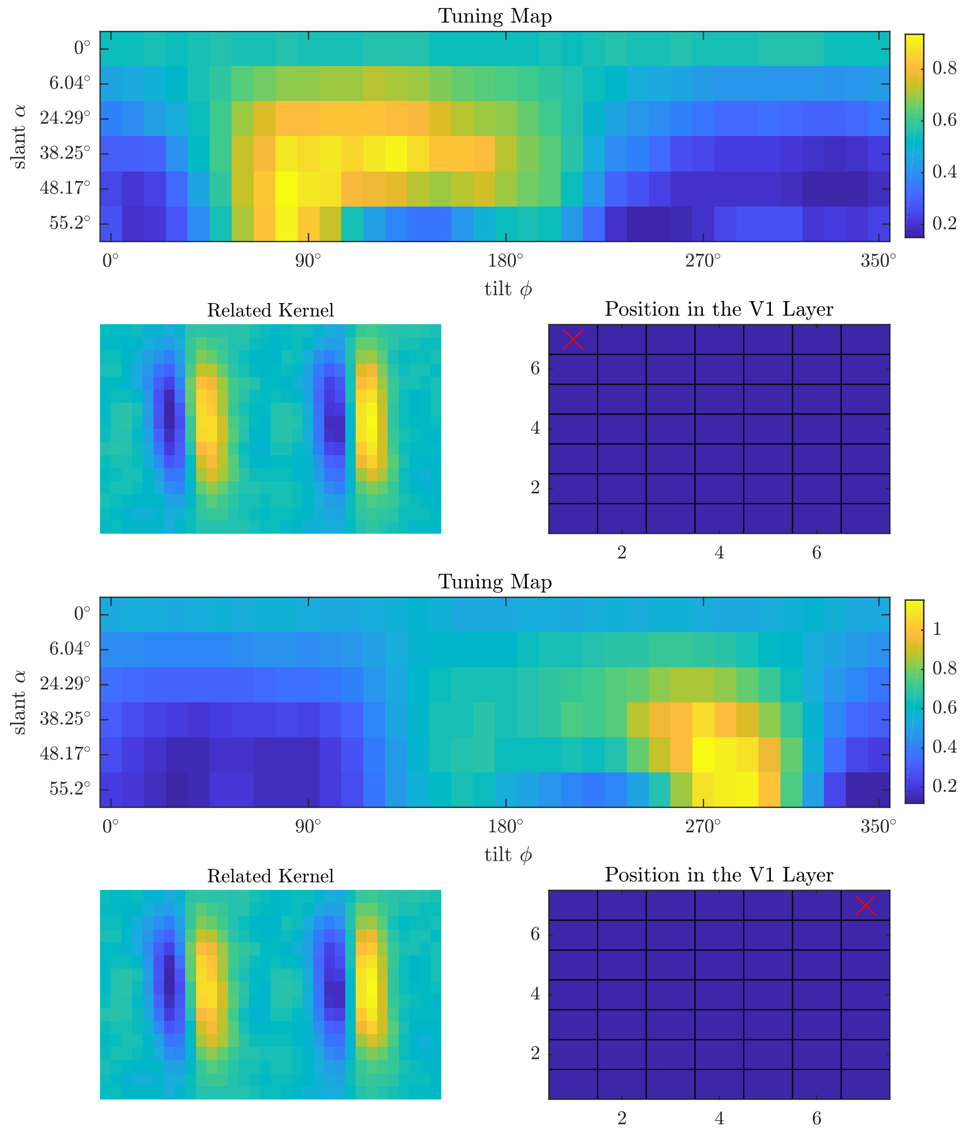


Figure 3.12: Two tuning maps from different corner positions of the same feature map with the depicted underlying basis function.

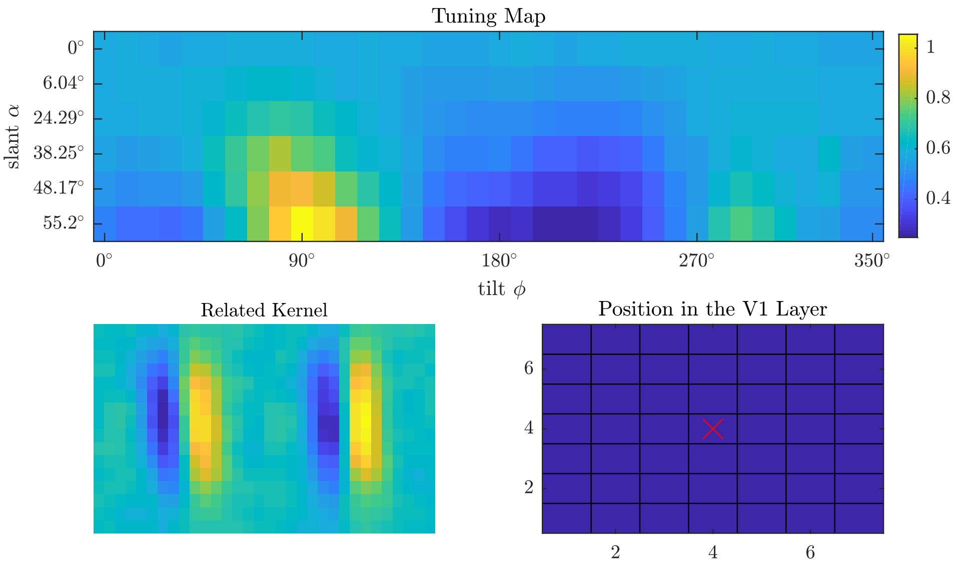


Figure 3.13: A tuning map from the middle position of a feature map with the depicted underlying basis function.

Looking at the middle neuron (Figure 3.13) of the feature map, a strong selectivity around the highest slant level, at $\phi = 90°$ is visible. The arc structure disappeared, only a weak selectivity for $\phi = 290°$ is left.

Strinking is, that the selectivity of the exemplatory neurons across the whole feature map, seems to bimodally agglomerate around $\phi = 90°$ and $\phi = 270°$: the two orthogonal directions, to the ON-OFF border of the underlying basis function.

Figure 3.14 shows the relationship of the middle neuron of all feature maps (which carries only information of tilt) and the rotation of their underlying basis function. Thereby, the selectivity of opposing directions were collapsed to the intervall of $[0°, 180°]$[1], to encounter the bimodality. The middle neurons are strongly, negatively correlated with the rotation of the Gabor fit of their underlying basis functions ($\rho = -0.9761$). Note that the rotation parameter of the Gabor fit is in the intervall $[-90°, 90°]$, but due to the circular structure of the data, this doesn't have an impact: the correlation stays informative.

[1] A method called angle doubling, which is common for circular data.

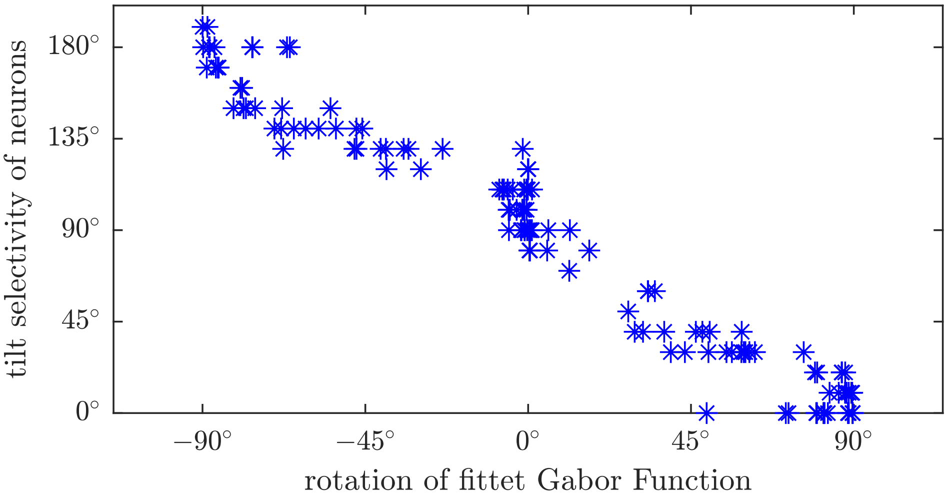


Figure 3.14: Relationship of the favored tilts of the tuning map, associated with the mid neuron of every feature map from the O1 neuronal set and the Gabor rotation parameter ϕ of the Gabor fits of the basis functions.

Estimations

To get an intuition on how exactly an estimation looks like, Figure 3.15 and Figure 3.16 depict four exemplary estimations. One has to imagine the stimulus being in the middle of the half-sphere (according to Figure 2.9).
The dark-blue belt on the lower side of the half-sphere is unsampled space, due to a maximal slant of $\alpha = 55.2°$. The magnitude of the log-likelihood-sum is visualized by the topography of the half-sphere. The bigger a ridge is, the more probable the observer looked at the stimulus from that perspective. In addition, high probabilities are color-coded.
For the tilt dimension, one step in the grid, corresponds to 10°. For the slant dimension the grid is upsampled, resulting in four grid-steps corresponding to one slant level. Ground truth of the stimulus is depicted by a red circle.
The assesment of the goodness of estimation is based on 1000 such estimations - per stimulus class. As for visualization: all estimations are congregated into one chart. The data is divided into five big columns along the x-axis, each standing for one slant level (note, that slant level $\alpha = 0°$ is left out). Every big column is divided into the 35 sub-columns, indicating the tilt level *at* every slant level. The sub-columns start at $\phi = 0°$ up to $\phi = 350°$ and are color-coded for better discrimination. Note, that, due to the circularity of the tilt-classes, the right-most sub-column is connected to the left-most sub-column.
In addition, every big column has a horizontal red line, depicting the mean of the estimations over all tilts. The y-axis shows the respective dimension. Every Figure is devided into two charts. While the upper chart shows estimations, employing the bias: the with-bias approach (see equation 2.17), the lower chart depicts estimations employing only encoding neurons: the without-bias approach (see equation 2.15). Due to the fact that the slant level of $\alpha = 0$ is left out, the lowest slant level, is slant level 2.

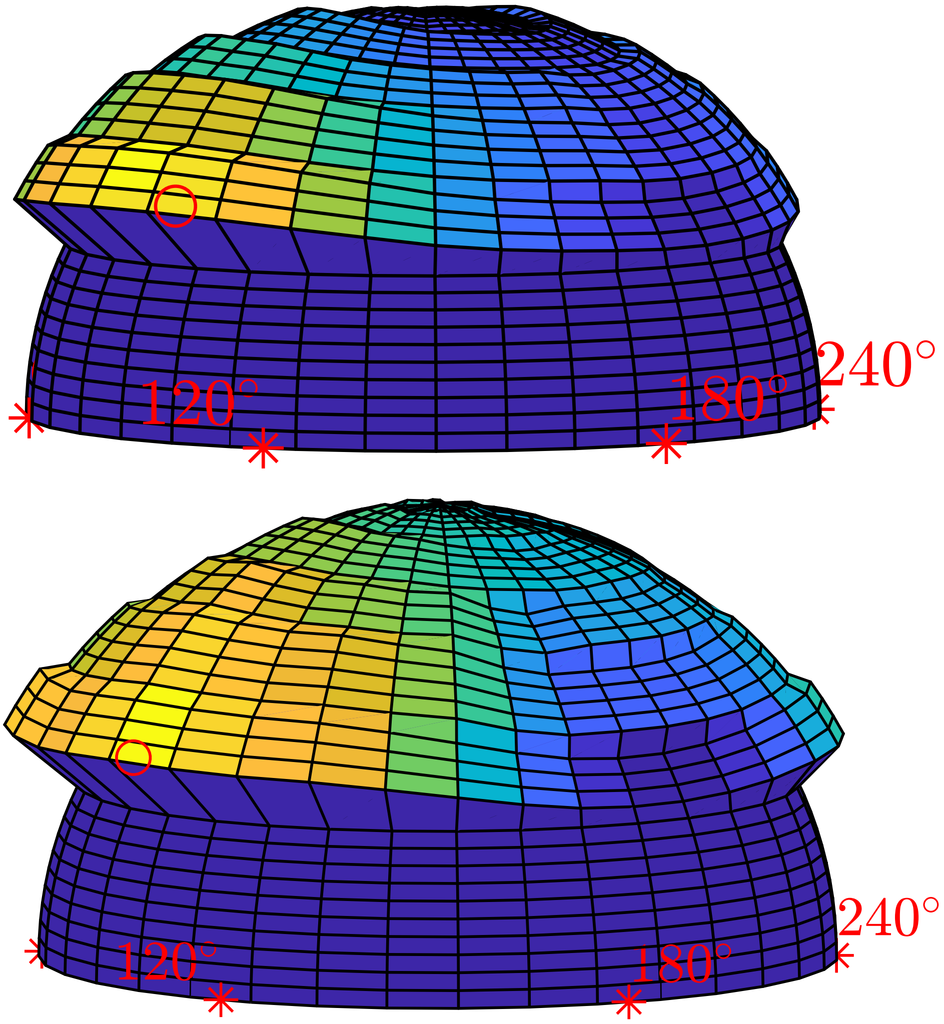


Figure 3.15: Two exemplary estimations, visualized on the half-sphere of perspective. Log-likelihood-sum magnitude is color and topology coded. The red circle depicts the ground truth of the underlying stimulus parameters.

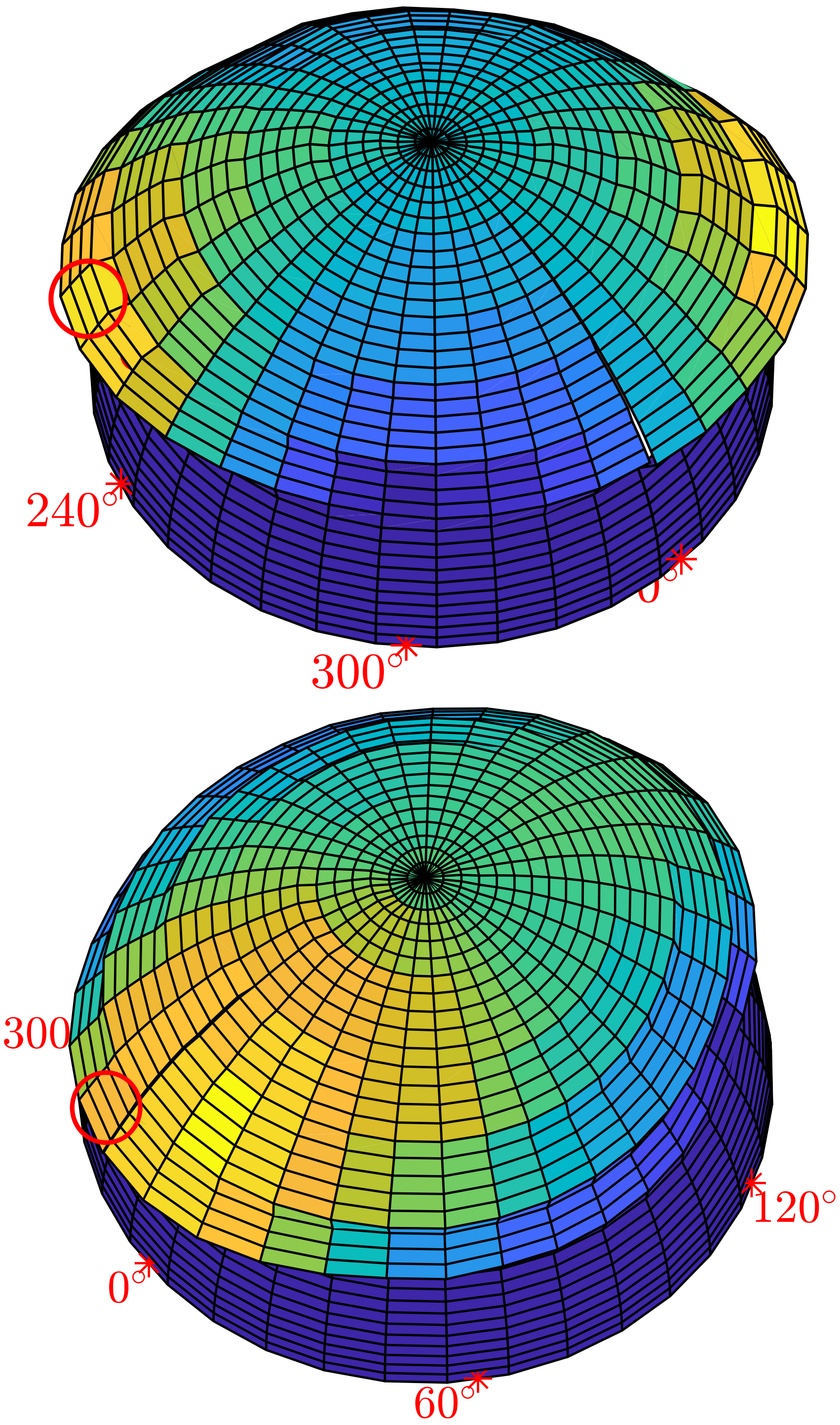


Figure 3.16: Another two exemplary estimations, from another perspective. Log-likelihood-sum magnitude is color and topology coded. The red circle depicts the ground truth of the underlying stimulus parameters.

Slant

Accuracy

Figure 3.17 shows the mean estimations for the O1, $m = 5$ set.
Although a rise of estimation in respect to slant level can be seen, both conditions fail for the three lowest slant levels, by constantly overestimating. The upper two slant levels are more on point: they rudimentarily estimate the correct slant.
Figure 3.18 depicts the with-bias/ without-bias conditions at O1, $m = 7$. The three lowest slant levels still are inaccurate, while the upper two match at least the right order. These conditions however, better estimate than the $m = 5$ condition: the rise along the slant levels is steeper and the upper two slant estimations are more accurate.
There seems to be a systematic interaction between slant estimation and tilt. Tilts at the cardinal directions of $\phi = 0°$,$\phi = 90°$,$\phi = 180°$ and $\phi = 270°$ constantly estimate lower slants, while at tilts in between at 45° rotation from the cardinal directions, estimate higher.
UEmploying the with-bias approach, enhances this regularity, as does a higher slant level.

Precision

Figure 3.19 shows the standard deviation of slant-estimations for the neuronal set O1, $m = 5$.
Both approaches show a stepwise decline of standard deviation: as the slant level gets bigger, the estimations become more precise. The with-bias approach has thereby a steeper secession: while having a mean standard deviation of 2.1557 for the lowest slant level, it falls to 0.7267 for the highest slant level, whereas the without-bias approach starts at a mean standard deviation of 1.77057, to then fall to 0.8581 for the highest slant level.
Figure 3.20 depicts the standard deviation for the $m = 7$ set. Similarly, standard deviation declines steeper for the without-bias approach. In comparision with the $m = 5$ set, standard deviation is smaller - the estimations are more precise.
The standard deviations at the three lowest slant levels support, that in fact no informative estimations are executed, but that inference occurs randomly.

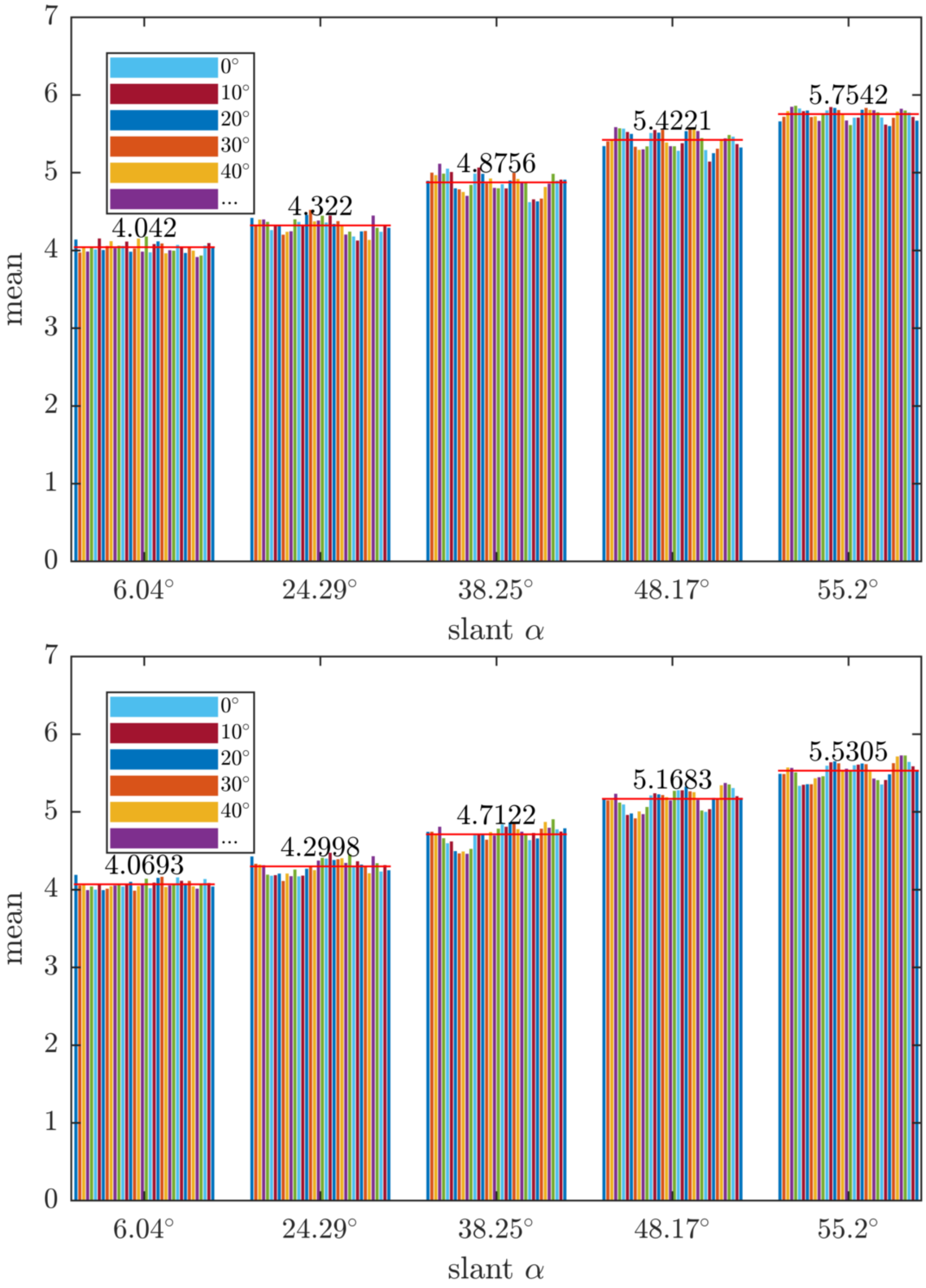


Figure 3.17: Mean estimations of two conditions for the neuronal set O1,$m = 5$. Big bars show slant level, sub-bars show mean estimations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all estimation-means at a slant level. *top*: with-bias condition; *bottom*: without-bias condition. Slant level is encoded for estimation: 6.04°: 2, 24.29°: 3, 38.25°: 4, 48.17°: 5, 55.02°: 6

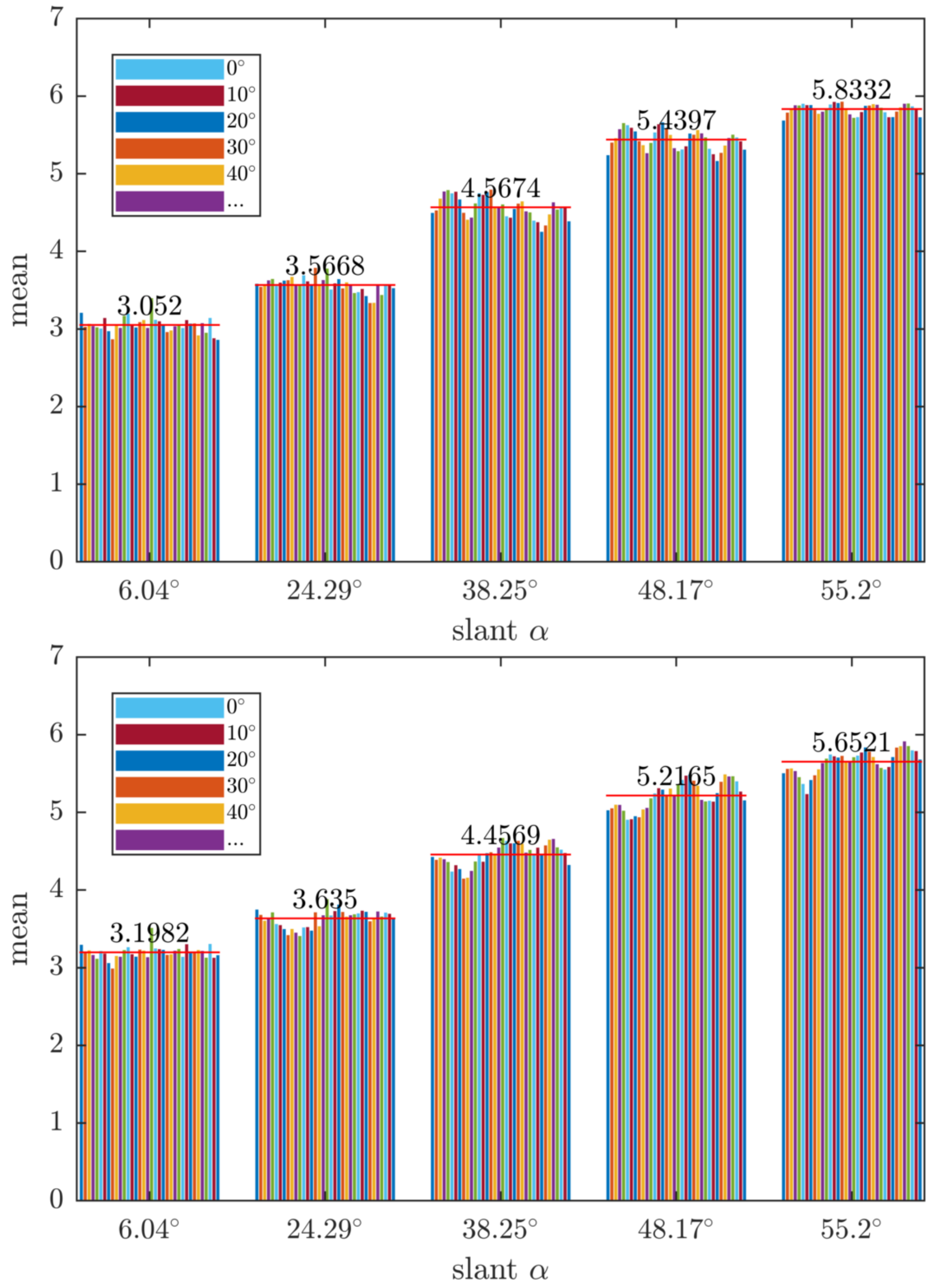


Figure 3.18: Mean estimations of two conditions for the neuronal set O1,$m = 7$. Big bars show slant level, sub-bars show mean estimations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all estimation-means at a slant level. *top*: with-bias condition; *bottom*: without-bias condition. Slant level is encoded for estimation: 6.04°: 2, 24.29°: 3, 38.25°: 4, 48.17°: 5, 55.02°: 6

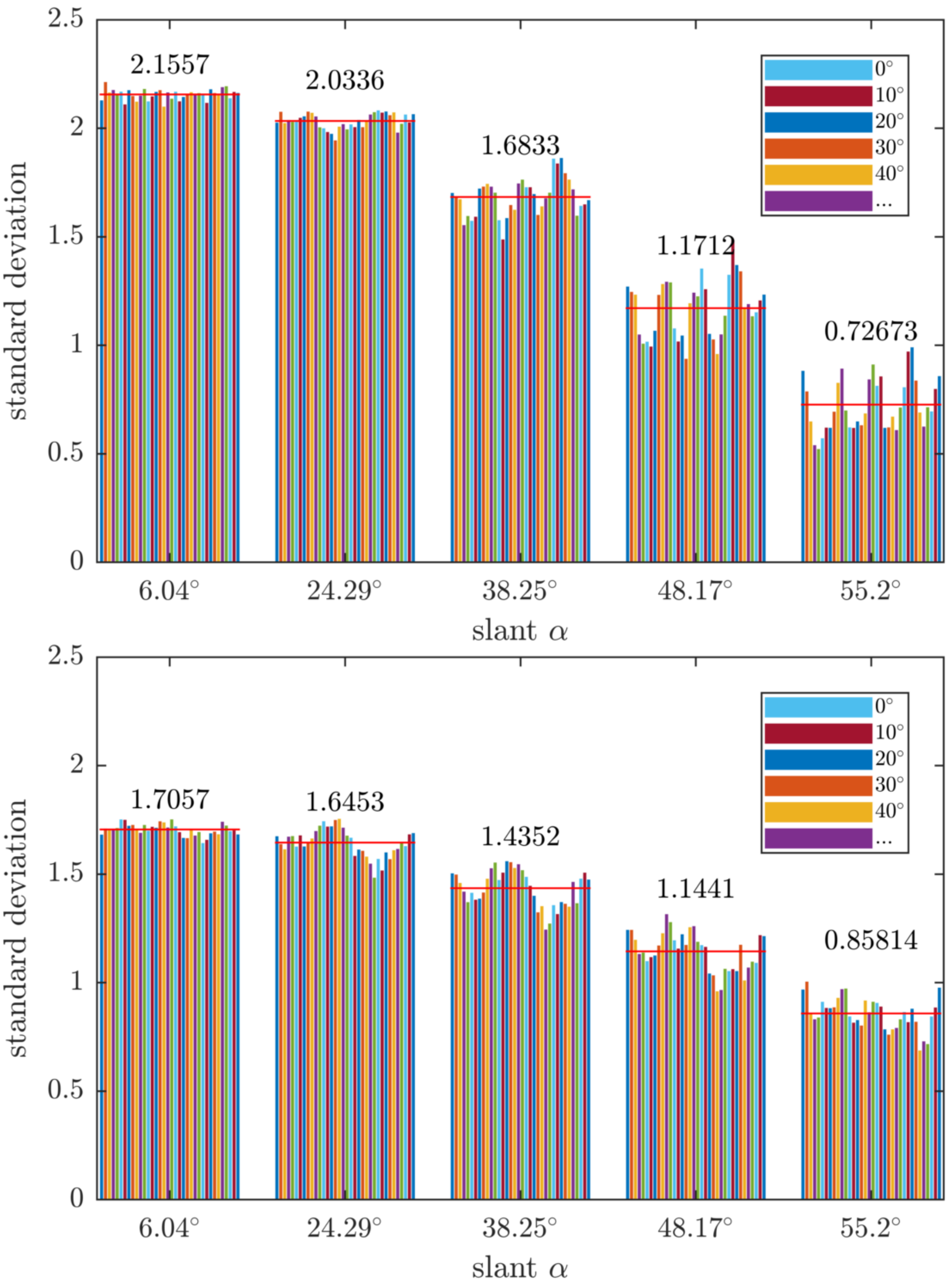


Figure 3.19: Standard deviation of two conditions for the neuronal set O1,$m = 5$. Big bars show slant level, sub-bars show standard deviations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all standard deviations at a slant level. *top*: with-bias condition; *bottom*: without-bias condition. Slant level is encoded for estimation: 6.04°: 2, 24.29°: 3, 38.25°: 4, 48.17°: 5, 55.02°: 6

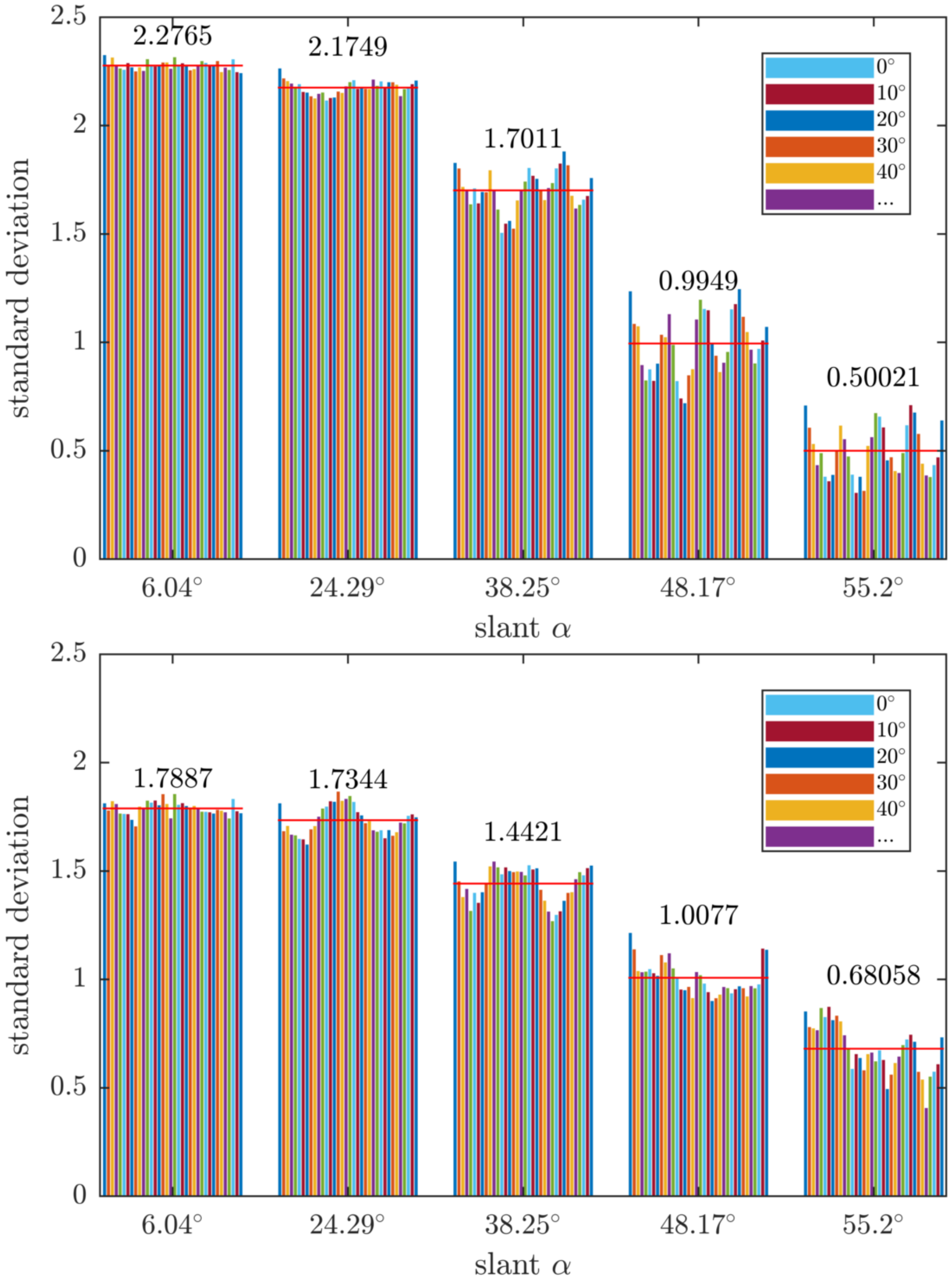


Figure 3.20: Standard deviation of two conditions for the neuronal set O1,$m = 7$. Big bars show slant level, sub-bars show standard deviations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all standard deviations at a slant level. *top*: with-bias condition; *bottom*: without-bias condition. Slant level is encoded for estimation: 6.04°: 2, 24.29°: 3, 38.25°: 4, 48.17°: 5, 55.02°: 6

A Closer Look at Slant Estimations

Mean and standard deviation do not catch the whole behaviour of slant estimation. A more in-depth look is provided by Figure 3.21: All estimations for the second slant level from the two neuronal sets, with-bias and without-bias respectively, are put next to each other. The O1,$m = 5$, with-bias set correctly estimates in some cases, but is overshadowed by many wrong estimations of the sixth slant level. In the without-bias approach, estimations of the correct slant level are nearly inexistent, while estimations of slant level 6 remain frequent. In contrast, the neuronal set of O1, $m = 7$, with-bias correctly estimates more often. In some cases estimations of the correct slant level are more common, than any other estimations. The without-bias condition on the other hand, seldomly estimates correctly, but constantly estimates slant level three. While the with-bias approaches show nearly no other estimations, than slant level two and slant level six, the without-bias approaches estimate more uniformly.

Figure 3.22 depicts all estimations for slant level three. The O1,$m = 5$ set never estimates correctly in the with-bias approach and seldomly in the without-bias approach. However, estimation distribution for the without-bias approach looks similar to the estimation distribution for slant level two. The O1,$m = 7$ set in the with-bias approach, estimates slant level two and slant level six at the same rate. Still, some correct estimations are done, as the distribution seems to be pulled towards slant level two. The without-bias approach correctly estimates for the most time across all tilt levels.

Figure 3.23 and Figure 3.24 reveal that both conditions with both approaches continue this trend - O1,$m = 5$ with both approaches and O1,$m = 7$, with-bias mostly estimate the highest slant level. Slight shifts in the remaining distribution can be however seen. Only the without-bias approach from O1,$m = 7$ has a stronger shift in its estimation-distribution.

At the highest slant level, all conditions with both approaches estimate most oftenly correct, as can be seen in Figure 3.25.

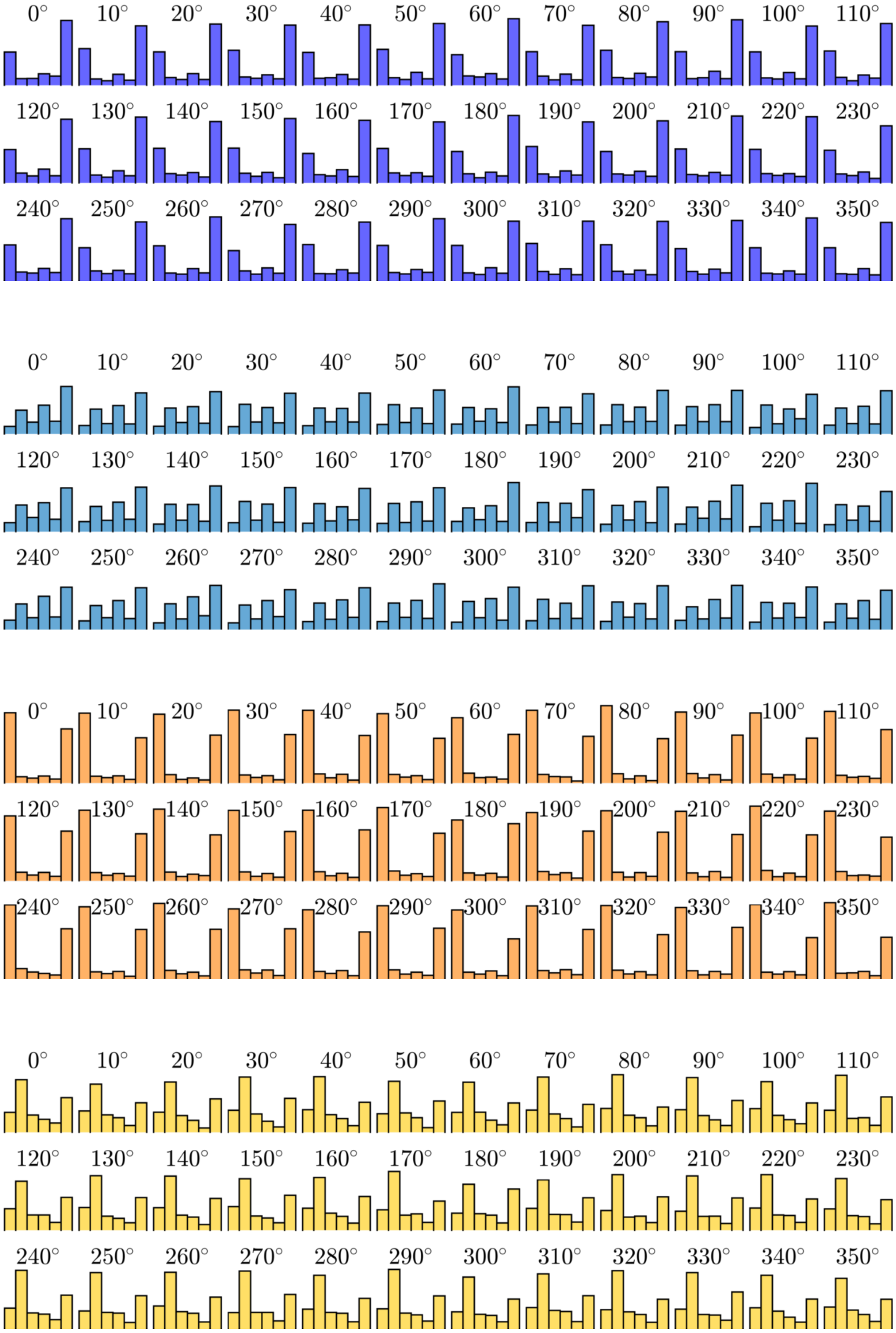


Figure 3.21: Histograms of slant estimations at all tilts for slant level 2 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

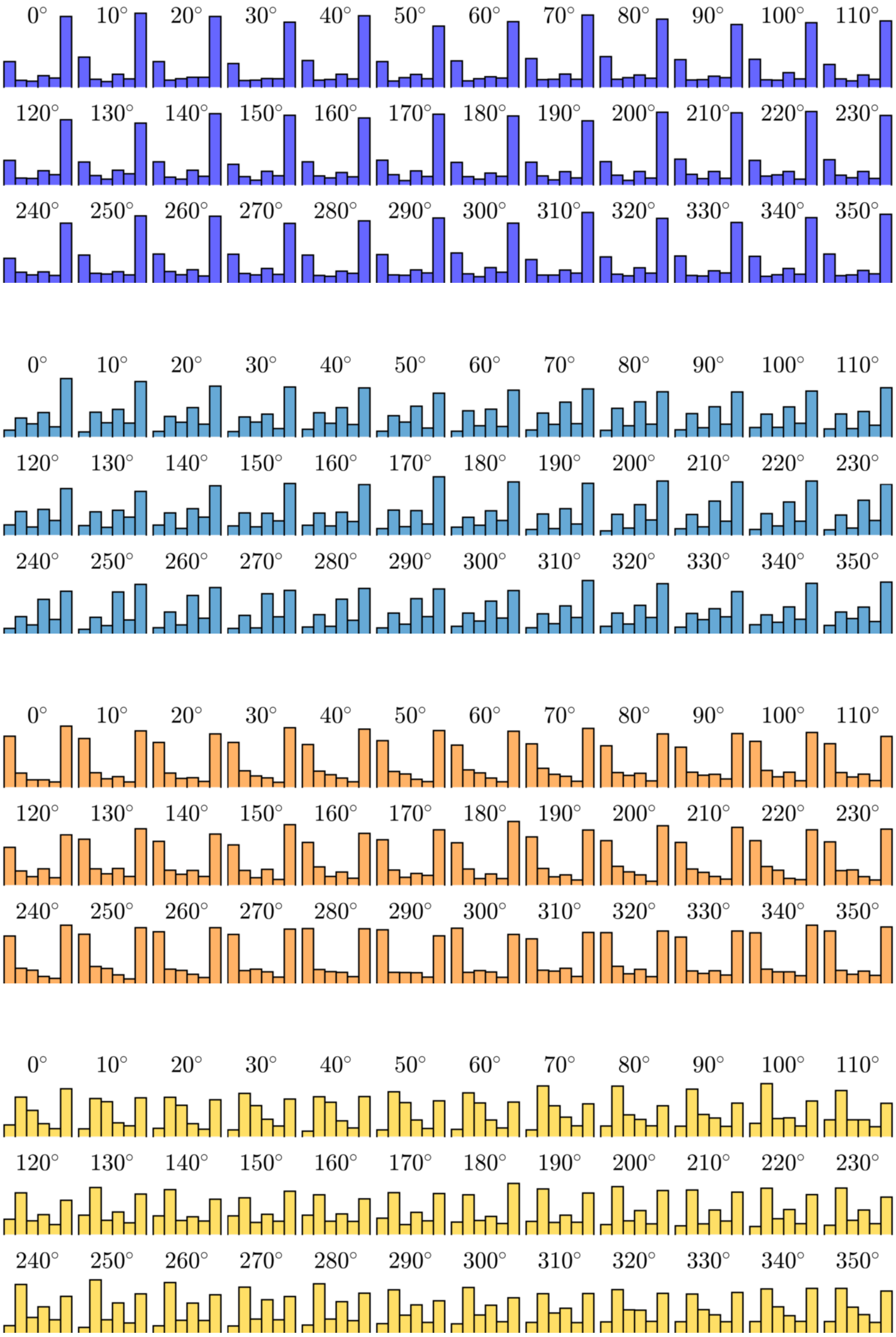


Figure 3.22: Histograms of slant estimations at all tilts for slant level 3 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

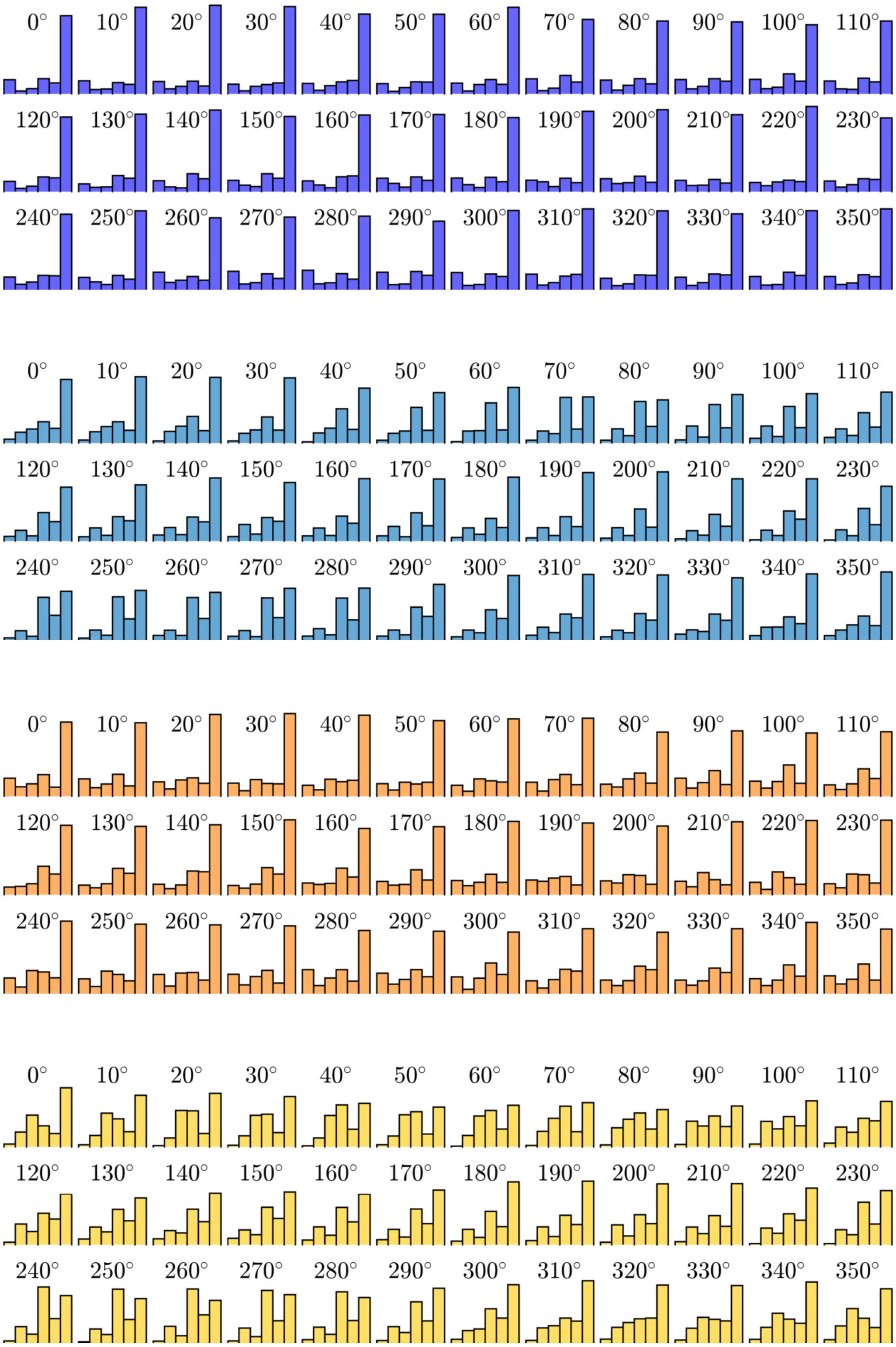


Figure 3.23: Histograms of slant estimations at all tilts for slant level 4 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

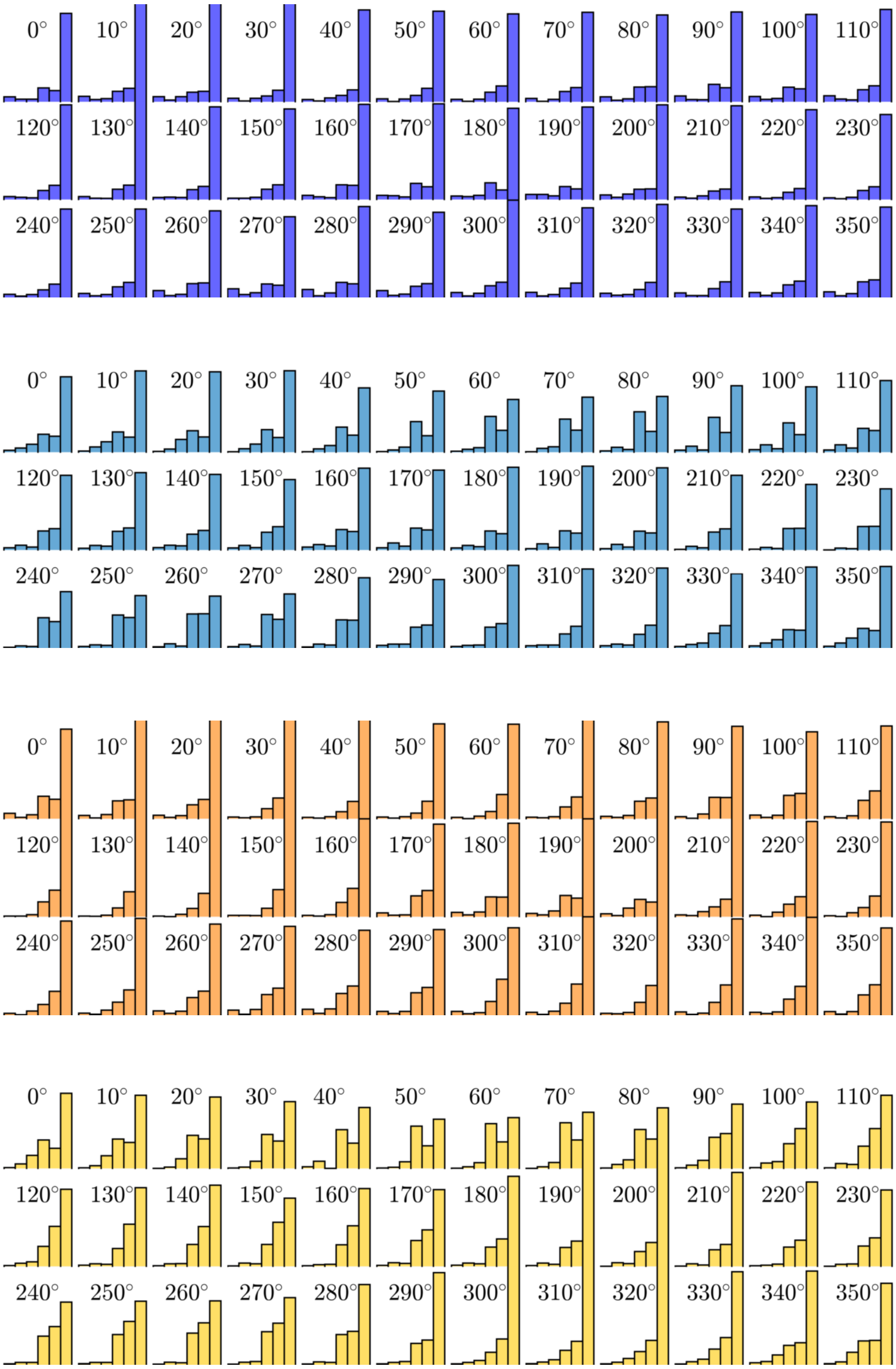


Figure 3.24: Histograms of slant estimations at all tilts for slant level 5 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

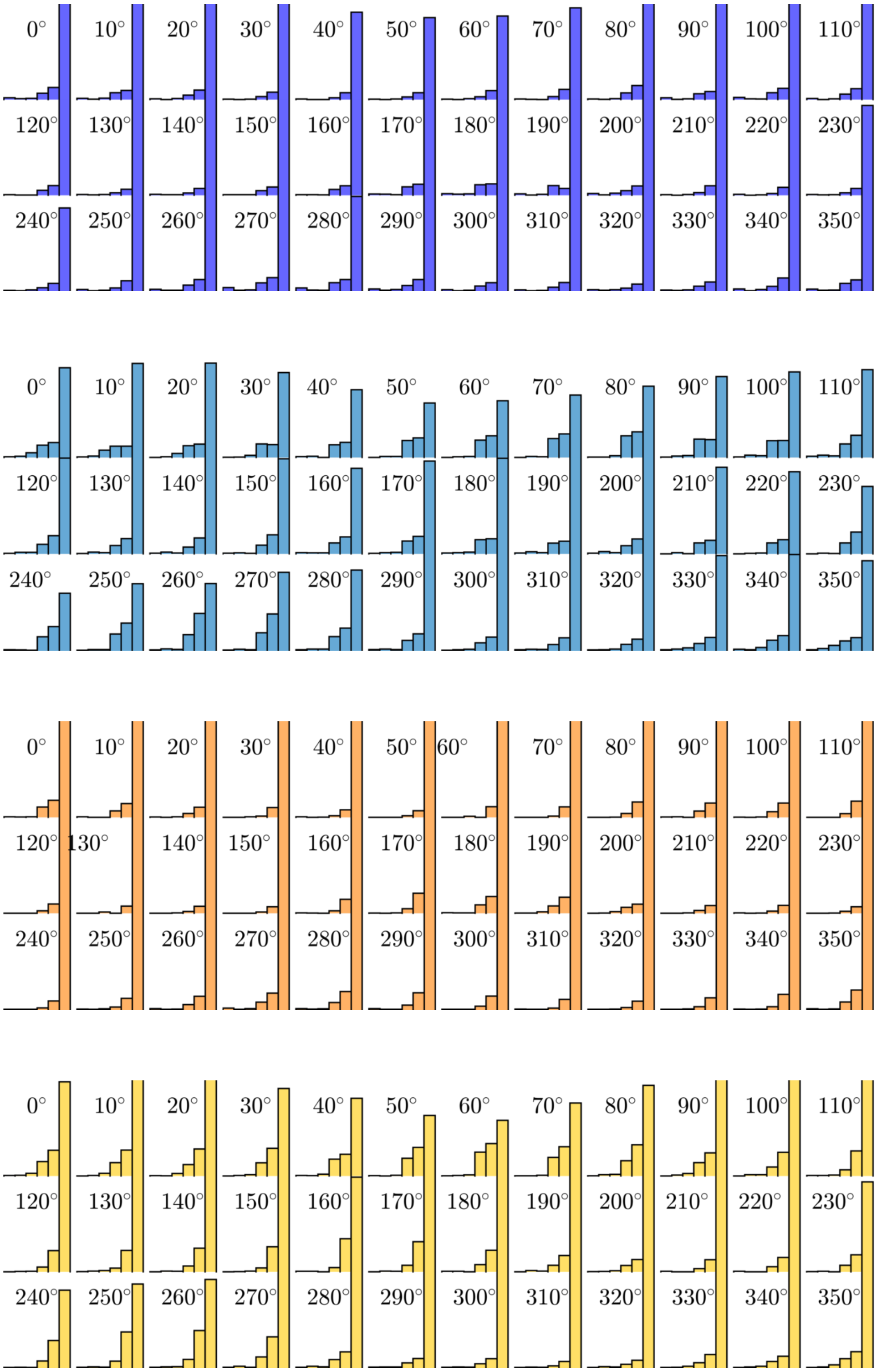


Figure 3.25: Histograms of slant estimations at all tilts for slant level 6 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

Tilt

Accuracy

As the mean signed error in Figure 3.26 shows, tilt estimation fails for the two lowest slant levels for both conditions of O1, $m = 5$. For the bias condition, estimations at the lowest slant level are constantly $\phi = 120°$: the farer a tilt level is from $\phi = 120°$ the bigger its estimation error, up to the diretly opposing tilt level of $\phi = 300°$, where estimation error is ±180°. This regularity also stands for the tilt estimations at slant level $\alpha = 24.29°$. As for the three higher slant levels, tilt estimation is more accurate. There seem to be a kind of pivot points, where estimation matches ground truth. Estimations from neighbouring tilt levels are gravitating to the nearest pivot point. As slant level is higher, the number and thus density of such pivot points seems to rise.

The same scheme holds for the no bias condition. Nevertheless, estimations at some tilt levels, even at the high slant levels, are completely inaccurate.

Figure 3.27 shows tilt estimations for the O1, $m = 7$ set. For the with-bias approach, estimations at the two lowest slant levels are again throughout the same : $\phi = 140°$. Remarkably, the no bias condition developed three pivot points.

Starting with the slant level $\alpha = 38.25°$, tilt estimations start to be fairly accurate: they never exceed ±50° deviation. For the highest slant level, estimations even are under ±20° deviation.

Precision

Figure 3.28 depicts the rotational standard deviation of tilt estimation for O1, $m = 5$. Both conditions show, that with higher slant level, estimation becomes more precise. At different slant levels, the variability of dispersion within the tilt levels coincides with the measures from the mean signed error: estimations at the so-called pivot points are more precise. The farer a tilt level is from such a pivot point, the more unprecise its estimation is. This behaviour is more smoothly observable for the with-bias approach.

In Figure 3.29, circular standard deviation for the O1, $m = 7$ can be seen. Behaviour is similar to the $m = 5$ case, only that $m = 7$ is constantly more precise.

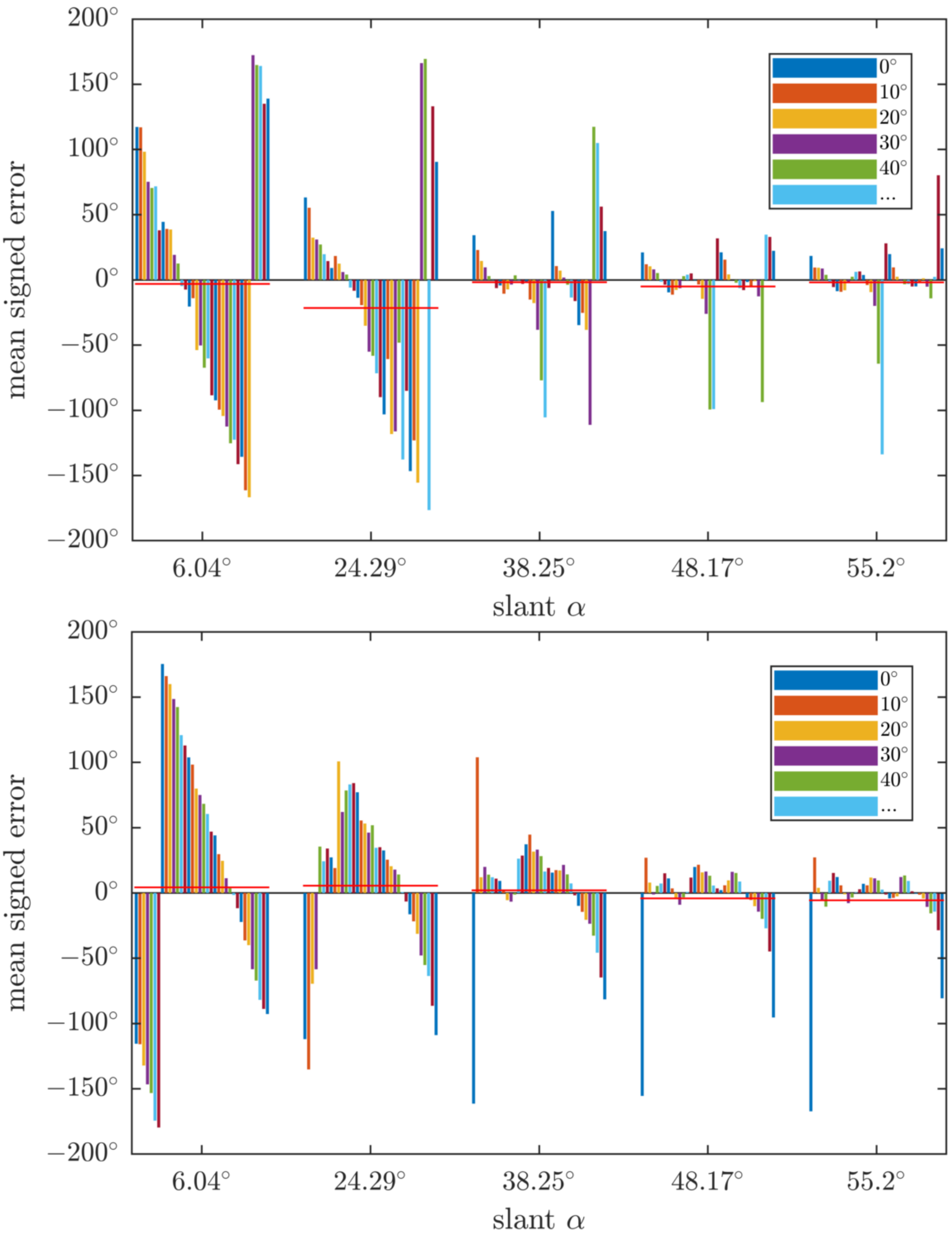


Figure 3.26: Mean signed errors of two conditions for the neuronal set O1,$m = 5$. Big bars show slant level, sub-bars show mean signed errors at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all mean signed errors at a slant level. *top*: with-bias condition; *bottom*: without-bias condition.

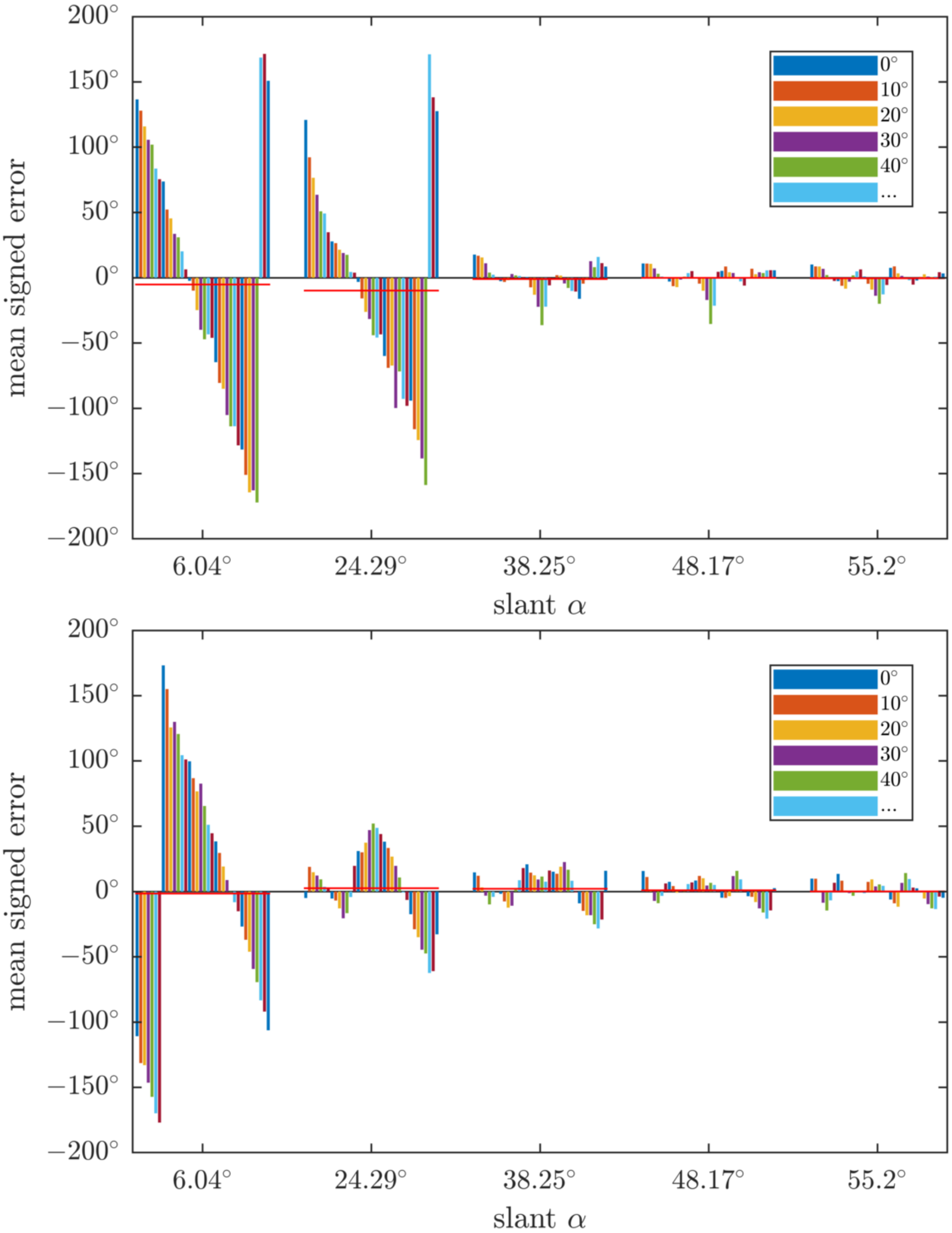


Figure 3.27: Mean signed errors of two conditions for the neuronal set O1,$m = 7$. Big bars show slant level, sub-bars show mean signed errors at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all mean signed errors at a slant level. *top*: with-bias condition; *bottom*: without-bias condition.

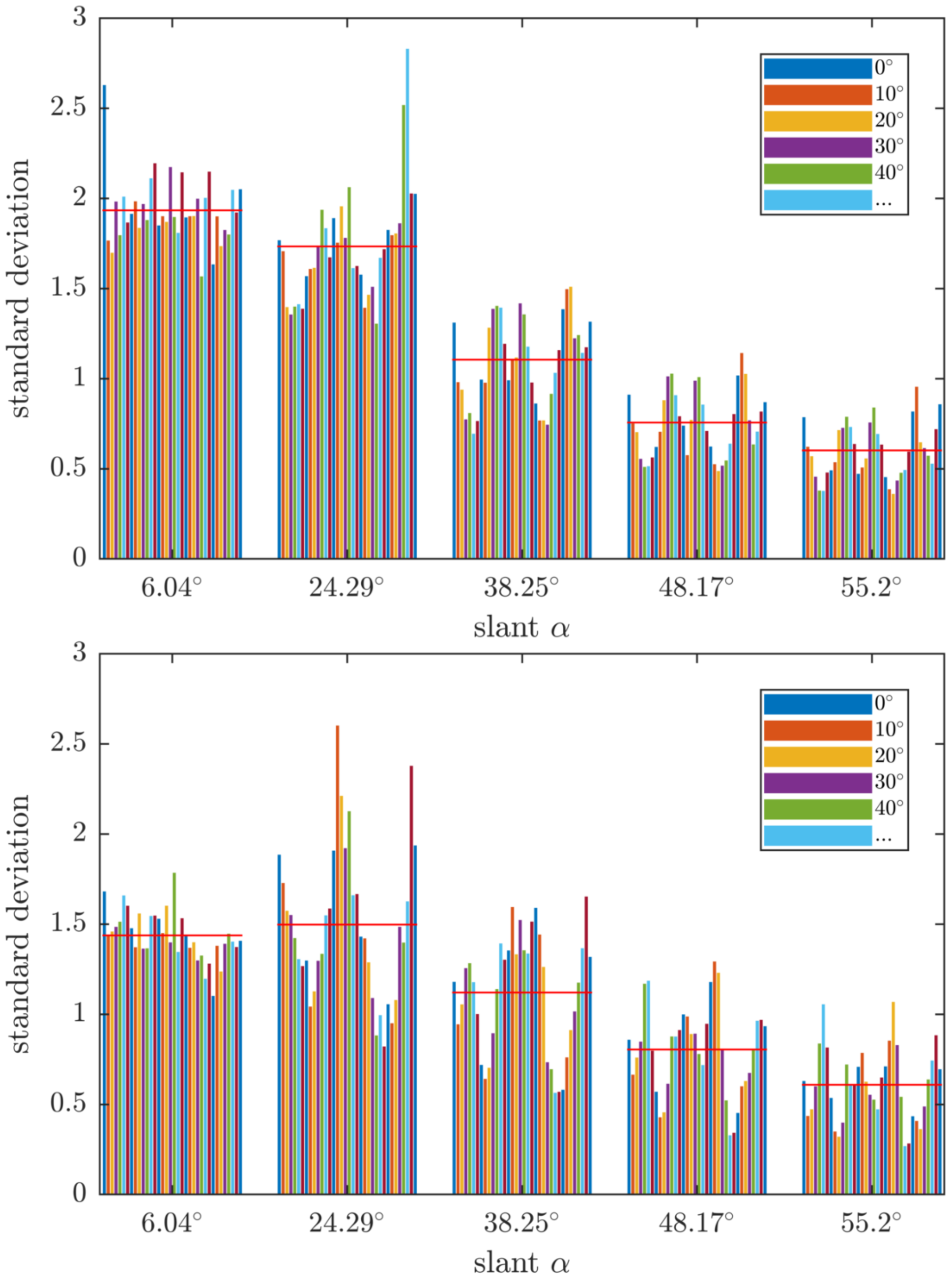


Figure 3.28: Circular standard deviations of two conditions for the neuronal set O1,$m = 5$. Big bars show slant level, sub-bars show circular standard deviations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all circular standard deviations at a slant level. *top*: with-bias condition; *bottom*: without-bias condition.

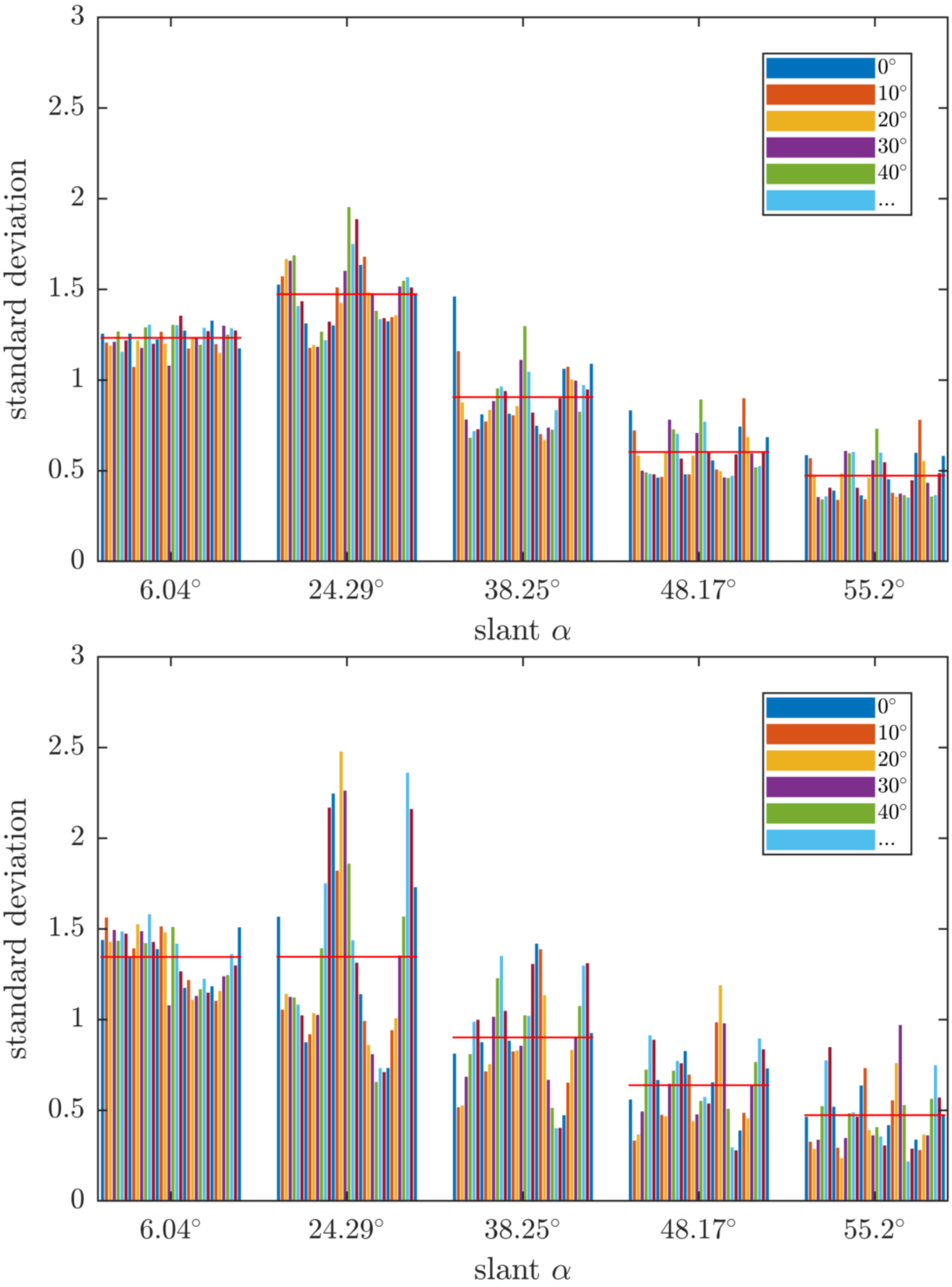


Figure 3.29: Circular standard deviations of two conditions for the neuronal set O1,$m = 7$. Big bars show slant level, sub-bars show circular standard deviations at tilt levels at the respective slant level. Red horizontal lines indicate the mean of all circular standard deviations at a slant level. *top*: with-bias condition; *bottom*: without-bias condition.

A Closer Look at Tilt Estimations

The two lowest slant levels did not allow any tilt-estimation, as could be already seen in the previous section. Figures can be found in the Appendix A.4.2. Slant level four, five and six can be seen in Figures 3.30 to 3.32.
Following the development of the first condition (O1,$m = 5$, with-bias) through the slant levels, one can see that starting at slant level 4 no sharp on-point estimations occur, at level 5 a clearer structure of estimation distribution emerges, which at level 6 finds its peak: rotating with the ground truth.
O1,$m = 7$, with-bias starts already at slant level 4 to roughly estimate the correct tilt. With rising slant level, the estimation-distribution becomes clearer and sharper, so that the estimation performs well at slant level 6.
Both without-bias conditions throughout all slant levels, estimate correctly for some tilt levels (for example $\alpha = 90°$), while other tilt levels are never correctly estimated.

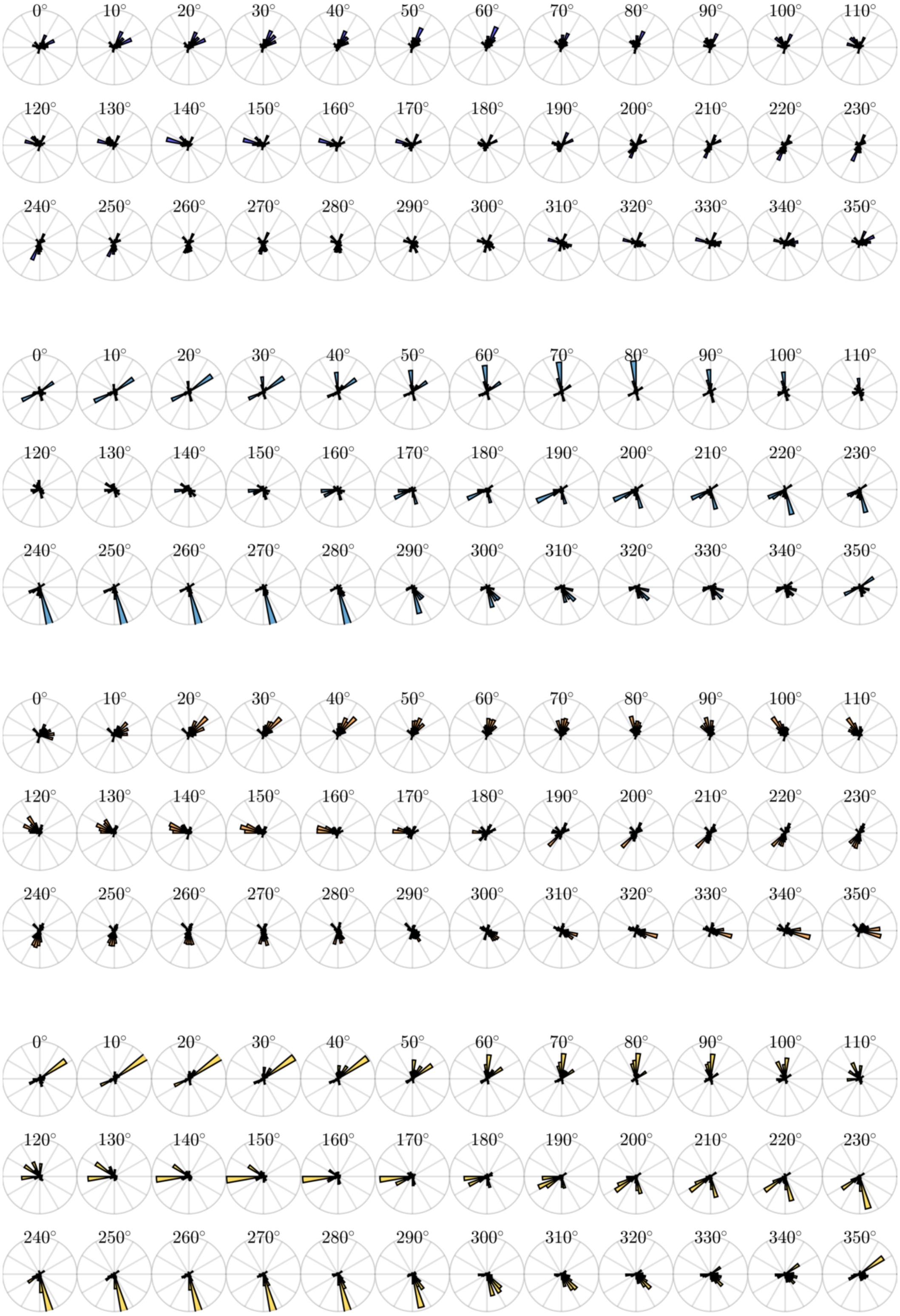


Figure 3.30: Histograms of tilt estimations at all tilt levels at slant level 4 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

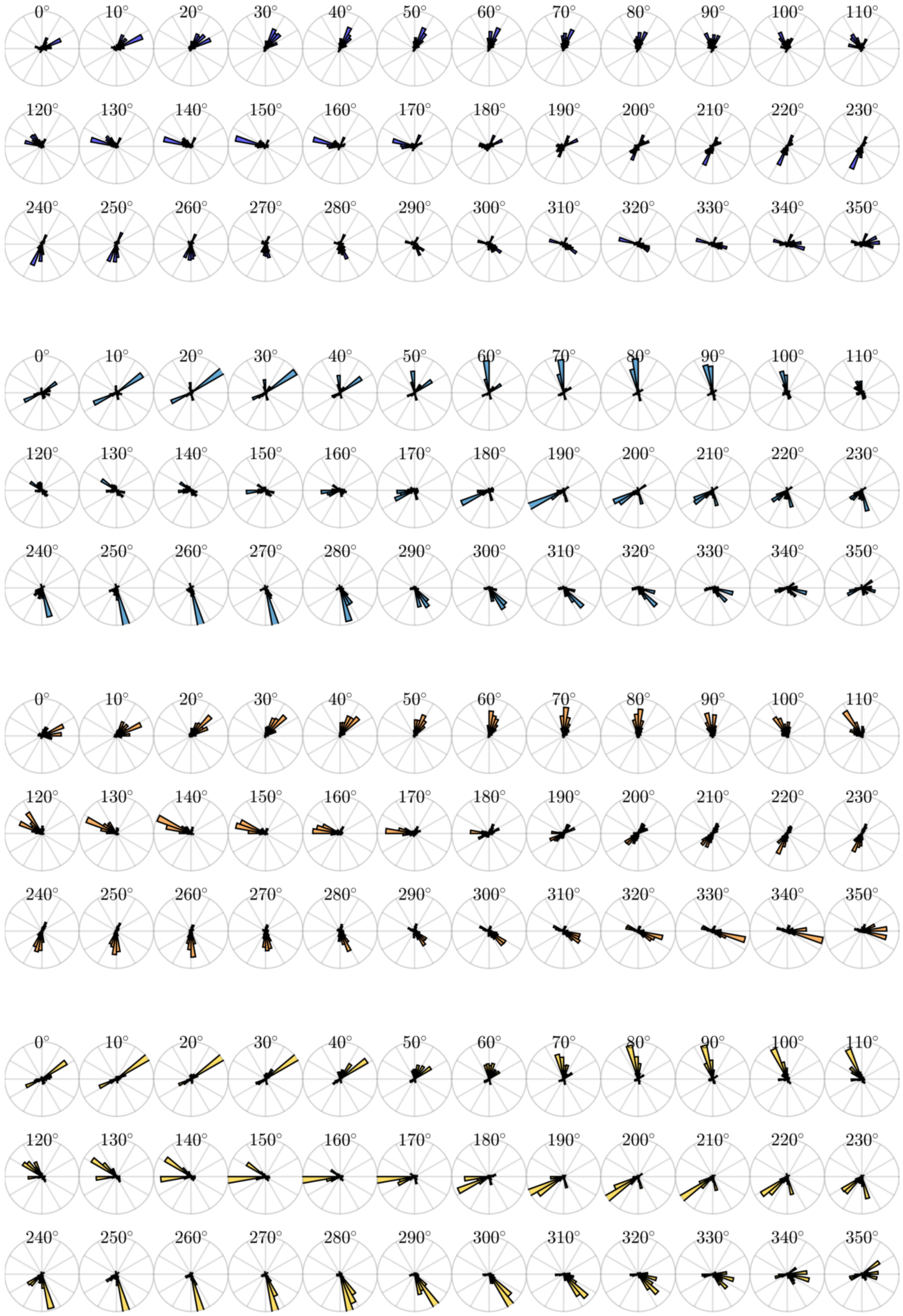


Figure 3.31: Histograms of tilt estimations at all tilt levels at slant level 5 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

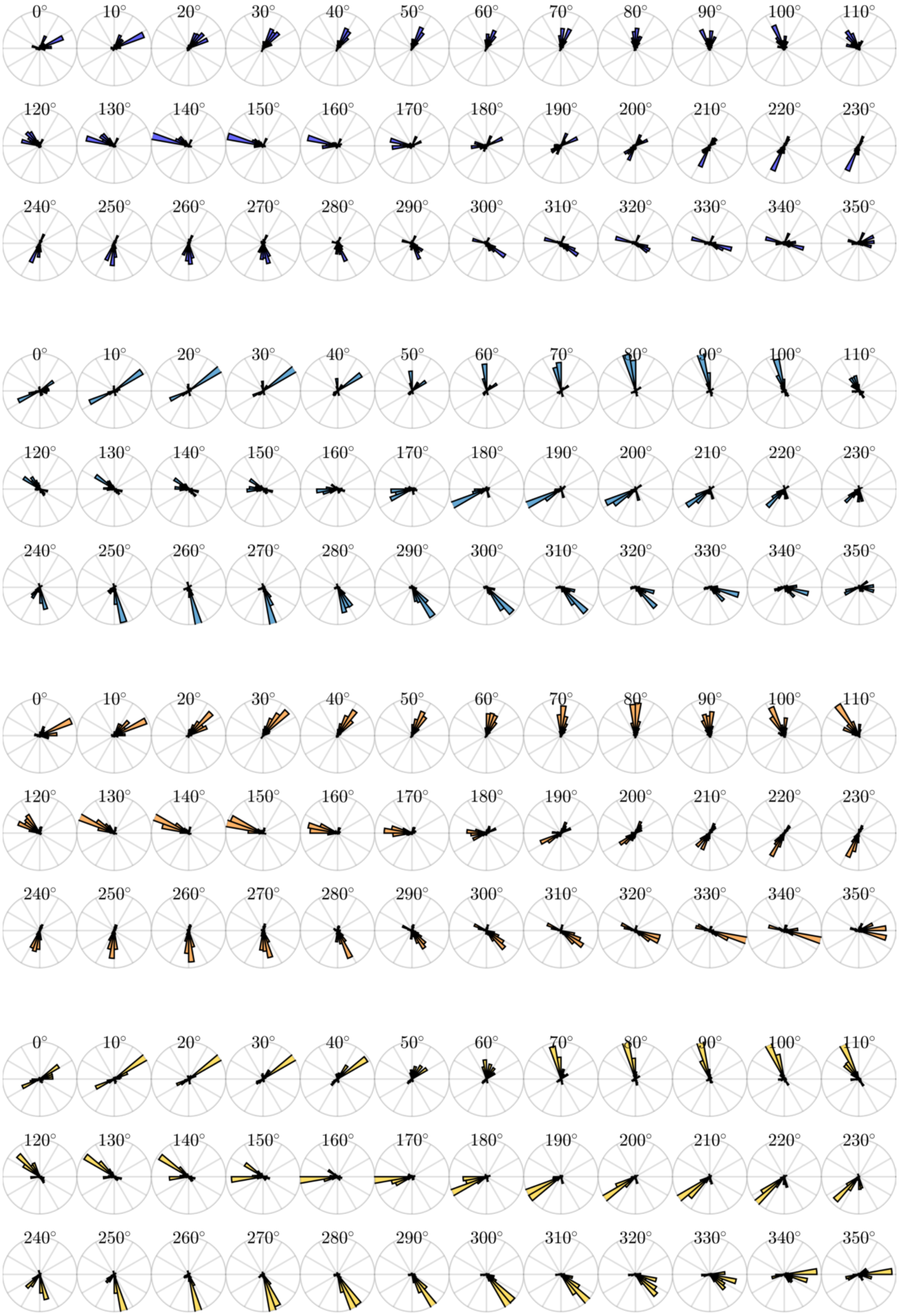


Figure 3.32: Histograms of tilt estimations at all tilt levels at slant level 6 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

Zeroth-Order Depth

Using tuning maps of neurons that emerged from stimuli with only zeroth order depth manipulation, the following analyses were conducted. Tuning maps were generated from only 50 stimuli per stimulus class (whereas tuning maps for the first-order depth estimation relied on 10500 stimuli per stimulus class). Exemplatory inference is based on a O1, $m = 7$, neuronal set with the with-bias approach.

Following figures all show the magnitude of the respective measure (mean signed error, standard deviation and mode) at every stimulus class.

Accuracy

The upper chart from Figure 3.33 shows the mean signed error for estimation of x-shift. Horizontal shift-estimation does not depend on y-shift: the estimation performs constantly across all y-shift stimulus classes.

X-shift however does have an impact on estimation. For negative shifts, the Bayes Classifier tends to slightly overestimate, while for strong positive shifts, x-shift is strongly underestimated. Estimation in the interval of $[-2.5, 2.5]$ pixel x-shift is very accurate.

The same behaviour stands for y-shift estimation, in the lower chart. X-shift, does not impact y-shift estimation, strong negative shifts are slightly overestimated, strong positive shifts are strongly underestimated. The interval of $[-2.5, 2.5]$ pixel y-shift seems well estimated.

Precision

Figure 3.34 shows the standard deviation for x-shift estimation (upper chart) and y-shift estimation (lower chart).

For x-shift estimation, the standard deviation seems to be constantly low for the well estimated interval. For strong negative x-shifts, the dispersion is more variable: some stimulus classes have a standard deviation of 0 (the classes, where estimation was perfect), some stimulus classes have a standard deviation of up to 6.

Stimulus classes of strong positive x-shifts show a high standard deviation.

Again, the same pattern can be seen for estimation of y-shifts, although the interval of strong variability at strong y-shifts is slightly narrower, than for x-shift estimation.

Estimating correctly, for the most Time

Figure 3.35 shows the mode of the 50 estimations per stimulus class. Clearly, at all stimulus classes, for x-shift estimation, as well as for y-shift estimation, the estimations were correct in most cases.

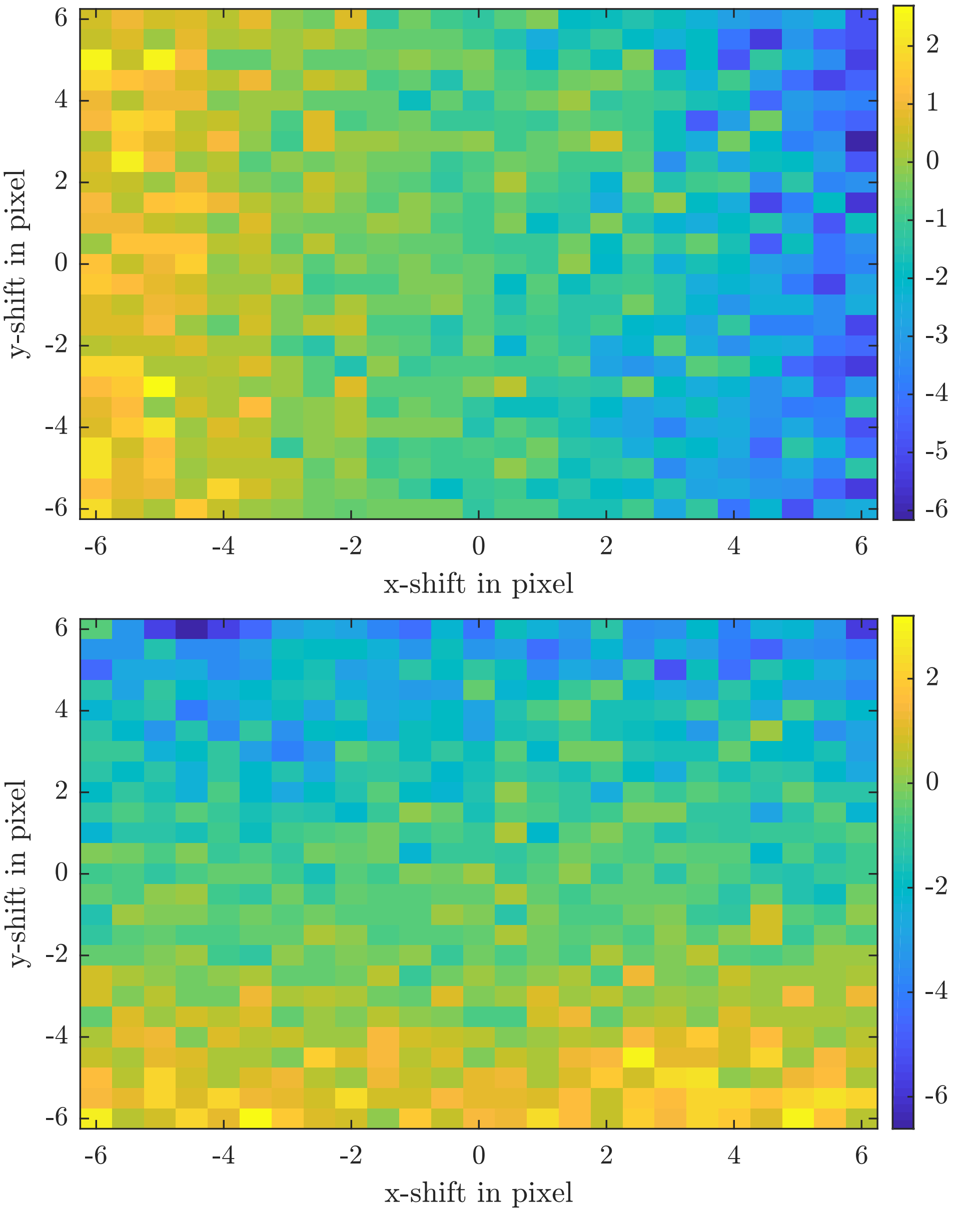
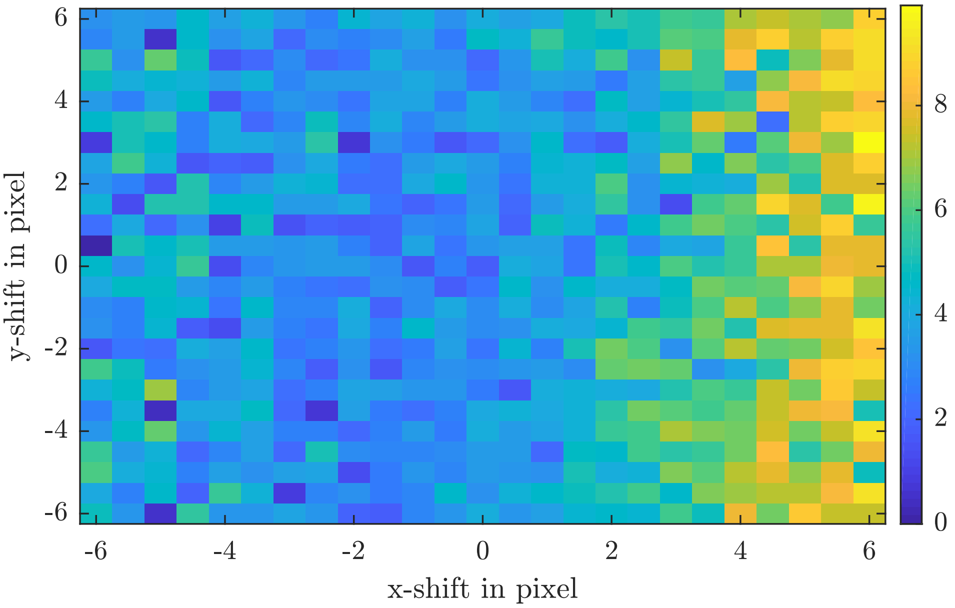


Figure 3.33: *top:* Mean signed errors of x-shift estimation at all stimulus classes. Color encodes the magnitude of the mean signed error; *bottom:* Mean signed errors of y-shift estimation at all stimulus classes. Color encodes the magnitude of the mean signed error.



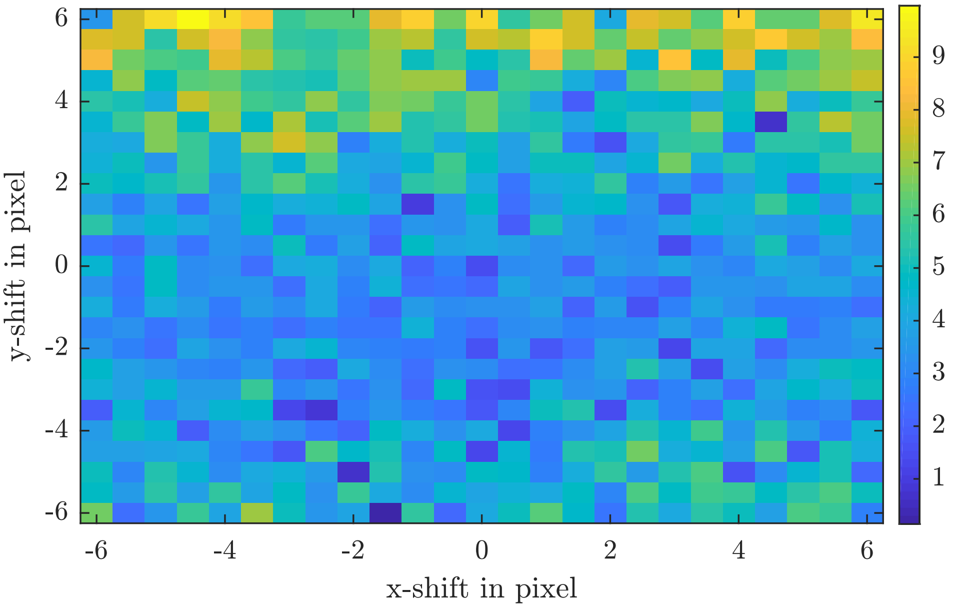


Figure 3.34: *top:* Standard deviations of x-shift estimation at all stimulus classes. Color encodes the magnitude of the standard deviation; *bottom:* Standard deviations of y-shift estimation at all stimulus classes. Color encodes the magnitude of the Standard deviation.

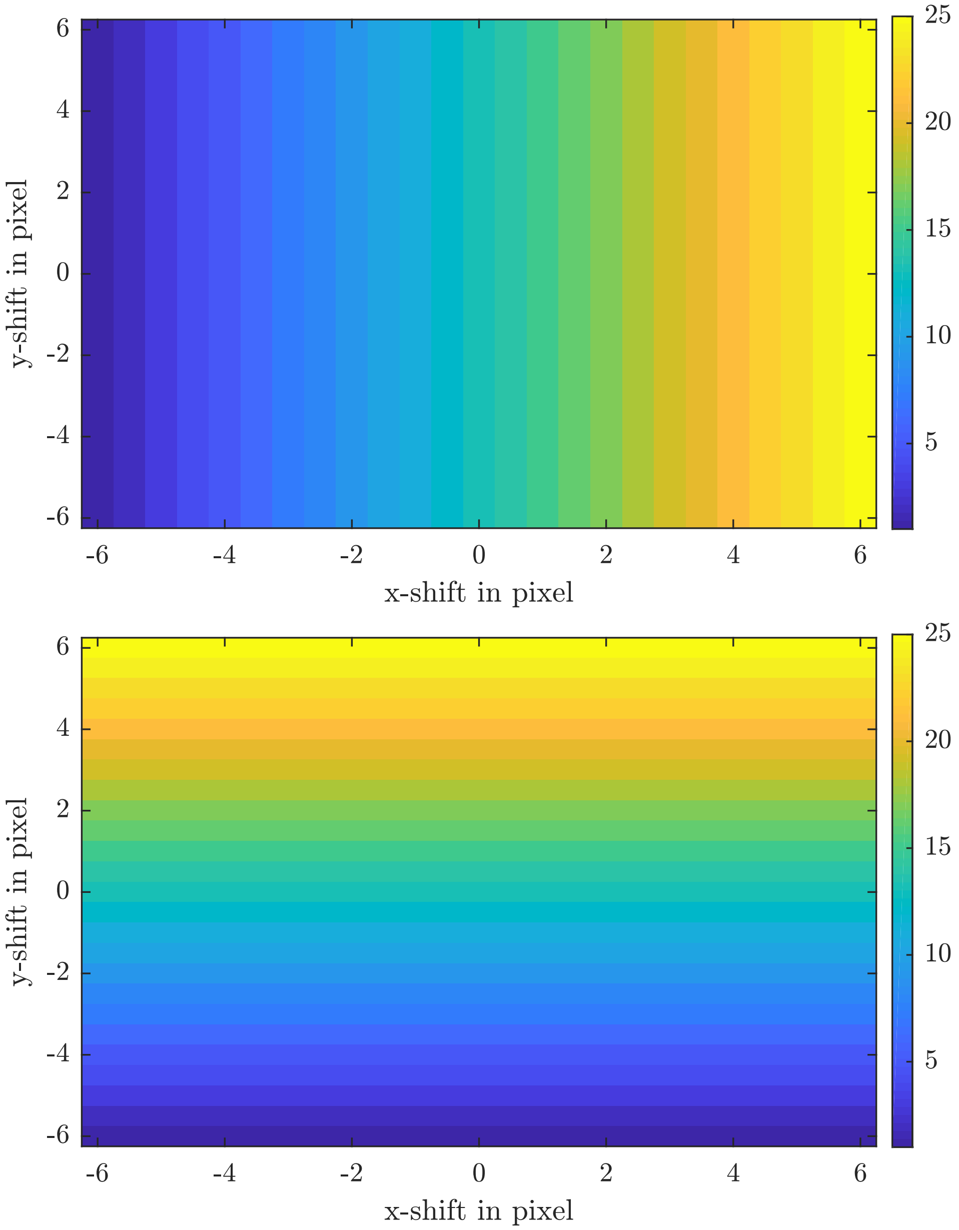


Figure 3.35: *top:* Modes of x-shift estimation at all stimulus classes. Color encodes the x-index of the most frequent estimation; *bottom:* Modes of y-shift estimation at all stimulus classes. Color encodes the y-index of the most frquent estimation.

Chapter 4

Discussion

Unencoded Stimuli and Significant Differences in Mean Encoding Neurons

A neuron is only active, if a feature on the stimulus matches the feature its receptive field is selective for. Featureless stimuli, as would be images of the clear blue sky or very smooth house facades, therefore remain unencoded. Since the neuronal sets take into account only features from a relatively small portion of an already small portion of a 256 × 256 pixel image, the chance of finding no feature at all, rises. This at least accounts for the 10% overall unencoded images across all stimulus classes.

Unexplained remain the abnormally many unencoded images around $\phi = 60°$ and $\phi = 240°$ for the highest slant. The fact, that these are directly opposing directions enforces, that there is some underlying regularity. As could be seen (in the mean tuning maps), the tilts at exactly those stimulus classes, were especially unfavored.

Looking at the distribution of the rotational parameter of the Gabor fitted the basis functions (Figure 4.1) may shed some light on the problem.

There are no basis functions, which encode features, which are rotated at around 75°. That however, is not exactly the range, where there are many unencoded images. In addition to that, encoding seems to have worked for lower slant levels.

Besides, it raises the question, why no such basis functions were learned in the learning phase of the neural network (see section 2.2.2 in the first place. At least the many basis functions for 90° and 180° could be explained, by the image dataset, containing many images of man-made objects, which usually agglomerate features around those rotations.

Considering, that *encoded* images from the same stimulus classes, were then encoded by especially many neurons, is furthermore curious. A more extensive analysis of the image statistics of the 1989900 stimuli is needed.

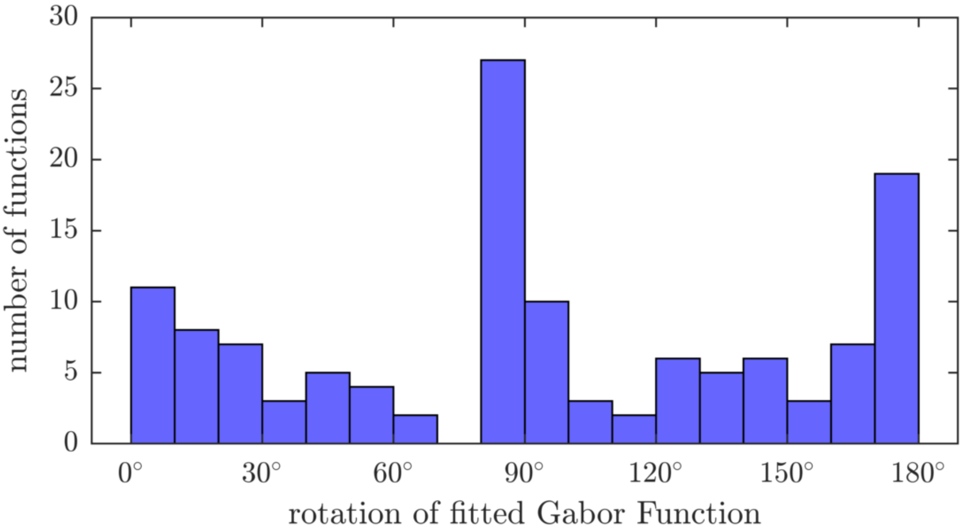


Figure 4.1: Histogram of the distribution of Gabor rotation parameter ϕ for the Gabor-fitted basis functions, underlying the feature maps of the O1 model.

Problems with Overcompleteness Factor 8

Tuning maps that stem from the neurons of the V1 model of overcompleteness factor 8, made any kind of inference impossible. It was anticipated, that neurons from O8 are more specialized, encoding more exotic features. The results pointed out the contrary: It seems, as if the neurons were too unspecific, being active all the time, for (almost) all stimulus classes. This makes an inconsistency at feature map learning highly probable.

A viable alternative for the future, would be to raise the hard threshold of activation λ at feature map learning. Consequently, neurons would be more picky in their activation, leading to more specific tuning maps. However, there are many parameters, and interaction is complex at the learning phase. A more in-depth analysis of the binocular SCANN would be useful.

Hence, my first hypothesis: An overcompleteness factor of 8 leads to better inference, is contradicted, as far as neuronal sets for O1 and O8 are learned with the *same* SCANN-parameters.

Estimations

Slant

All in all slant estimation performance was underwhelming. Quantitatively, no condition could satisfy any requirements. Admittedly, with bigger slant level, a more precise and more accurate estimation *seems* to be possible. However, it remains unclear if the fact, that estimations of slant level 6 are so overrepresented, does not distort the results: slant level 6 is not correctly estimated, if all slant levels are estimated as level 6. Still, besides slant level 6, small shifts in estimation-distributions could be observed.

One possible source of constantly estimating the highest slant level, is the overrepresentation of tuning maps, highly selective for slant level 6: the chance of a stimulus to be encoded by many slant level 6 selective neurons is simply higher.

Still it is possible to say, that the bigger neuronal set of $m = 7$ qualitatively outperformed the smaller set of $m = 5$ in terms of accuracy and precision. Hereby, the without-bias approach of estimation proved itself the most accurate. It had the smallest distortion of always-estimating slant level 6.

This outcome might be due to the fact, that more excentric neurons carry more information, especially at small slants. There, disparity is stronger and more distinct.

Another possible reason is the small area under observation: For the slant level of $\alpha = 6.04°$, neurons near the center of the stimulus only perceive disparity in the sub-pixel domain.

Tilt

Inference of the tilt parameter performed quantitatively adequate - beginning from a slant level of $\alpha = 38.25°$. From there on, up to higher slant levels, estimations became more accurate and more precise. Especially the $m = 7$, with-bias condition was sharp in its estimations. At the slant level of $\alpha = 55.2°$, even only (mean) deviations of 10° were present.

One of the reasons for the good performance might be, that tuning map selectivity was more uniformly distributed across tilt stimulus classes. So, the chances of stimuli, being encoded by meaningful tuning maps, was higher.

Another reason might be, that the tilt parameter is closely related to the rotational parameter of the basis functions - in fact they are highly correlated.

The earlier called pivot points seem to emerge at tilt levels, where there are many selective tuning maps for. This is coherent with the high number of tuning maps, selective for slant level 6, leading to always estimating slant level 6.

Summary

The big neuronal set of O1,$m = 7$ (irrespective of with-bias or without-bias approach) performed better, than both approaches for O1,$m = 5$. Hence, my second hypothesis, namely that a bigger area under observation, leads to better inference, could not be contradicted.

Depth Information *is* in the model

The mediocre performance for slant estimations, might lead to wrong conclusions. Remembering the selectivity behaviour of neurons across the same feature map: much of that information remains unused. The manner in which selectivity changes with position, might allow to compute the gradient and thusly directly infer first-order depth. At least the selectivity-dynamic holds some kind of information, simply because of the spacial organization.
Another source of yet unused information, is the fact that neuronal activity strength is completely ignored. For building tuning maps, it is only counted *if* a neuron is active. Having more nuanced neuronal activation, possibly leads to more potent tuning maps.
A last possibility to gain more information, would be to use a more elaborated method of inference. By employing the knowledge of how the model is construed or adding a neuronal layer with complex wiring, surely better results could be obtained, than by simple log-likelihood-summation.

Zeroth Order Depth and First Order Depth

While a whole experiment was simulated, stimuli were created and models were learned, only to show that inference of first order depth information is possible, at the heart of this work stands the claim, that a sparse stereo-representation is a useful and powerful format to hold sensory data in.
And indeed, while estimation of (at least) the slant parameter was not satisfactory, tilt could be acceptibly inferred - especially in the O1,$m = 7$ condition - leading to at least, a proximate estimation of first order depth.
The crucial point is that in parallel, zeroth order depth could also be inferred with fairly better results. However, no model alterations were rendered, no extra parameters were set. Just by observing stimuli of different nature (with even the same underlying basis functions), neurons showed different selectivities, enabling inference of both orders of depth.

Limitations

The sparse representation also brings methodological problems. Due to the sparseness, the amount of stimuli needed, to obtain sufficiently dense-sampled tuning maps, is very high. In a first try of the method employed in this work, the stimulus database comprised about 200000 stimuli, what turned out to be too few: no selectivity patterns emerged on the tuning maps.

Is the Experiment too Abstract?

Many simplifications were undertaken, beginning with a linear retina, up to the scarce stimulus presentation. Through texturing of a flat surface, depth manipulation then led to structures, and deformations, which seldomly appear in reality. Depth was only mediated through disparity. No other information, from any of the many depth cues, entered the sparse representation. That could of course be interpreted as a well controlled experiment, but on the other hand it is not surprising that depth inference did not work perfectly, with such withered stimuli.
No one would expect a human, to correctly tell the rotation of a square-centimeter surface in space, after looking from one meter, disregarding the edges. So maybe a little more proximity to literal natural images, would boost the performance of inference.

Outlook

This work made a first assessment of the role of a sparse stereo-representation for depth estimation. First order depth was inferred with mediocre results, but all in all, it could be shown, that information needed for such an inference is availible.
For the future I propose to change the SCANN parameters in such a form, that useful, more specific neurons emerge for high overcompleteness factors. Selectivity on tuning maps should thereby be uniformly distributed across all stimulus classes.
Creating a stimulus data base, with constant mean encoding neurons across all stimulus classes, could further minimize bias and optimize tuning maps. Another option would be, to reconstruct a 3D environment, so that some more natural cues are present in the stimuli.
In addition the model's sparsity factor could have a strong impact on the information representation and is worth investigating.
With such enhancements of the method, finally, second order depth stimuli - convex and concave shapes could be created. Showing that a sparse stereo-representation, also allows inference of such stimulus parameters, would be thrilling.

References

Abbasi-Asl, R., Pehlevan, C., Yu, B., & Chklovskii, D. (2016). Do retinal ganglion cells project natural scenes to their principal subspace and whiten them? In *Signals, systems and computers, 2016 50th asilomar conference on* (pp. 1641–1645).

Anzai, A., & DeAngelis, G. C. (2010). Neural computations underlying depth perception. *Current opinion in neurobiology*, *20*(3), 367–375.

Atick, J. J., & Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural computation*, *4*(2), 196–210.

Barlow, H. B. (1961). Possible principles underlying the transformations of sensory messages.

Berens, P., et al. (2009). Circstat: a matlab toolbox for circular statistics. *J Stat Softw*, *31*(10), 1–21.

Bosking, W. H. (2008). V1 neurons: in tune with the neighbors. *Neuron*, *57*(5), 627–628.

Ecke, G. (n.d.). *not yet availible.* (unpublished)

Ferris, S. H. (1972). Motion parallax and absolute distance. *Journal of experimental psychology*, *95*(2), 258.

Fisher, N. I. (1995). *Statistical analysis of circular data.* Cambridge University Press.

Heil, C. (2010). *A basis theory primer: expanded edition.* Springer Science & Business Media.

Huang, S. Y. (n.d.). *Savitzkygolay2d in matlab.*

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215–243.

Janssen, P., Vogels, R., Liu, Y., & Orban, G. A. (2003). At least at the level of inferior temporal cortex, the stereo correspondence problem is solved. *Neuron*, *37*(4), 693–701.

Jones, J. P., & Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, *58*(6), 1233–1258.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S. A., & Hudspeth, A. J. (2000). *Principles of neural science* (Vol. 4). McGraw-hill New York.

Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., ... Wiskott, L. (2013). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1847–1871.

Lipton, L. (1982). *Foundations of the stereoscopic cinema: a study in depth.* Van Nostrand Reinhold.

Lundquist, S. Y., Paiton, D. M., Schultz, P. F., & Kenyon, G. T. (2016). Sparse encoding of binocular images for depth inference. In *Image analysis and interpretation (ssiai), 2016 ieee southwest symposium on* (pp. 121–124).

Mallot, H. A. (2000). *Computational vision: information processing in perception and visual behaviour.* MIT Press.

Murphy, K. P., et al. (2006). Naive bayes classifiers. *University of British Columbia*, *18*.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607.

Orban, G. A. (2011). The extraction of 3d shape in the visual system of human and nonhuman primates. *Annual review of neuroscience*, *34*, 361–388.

Pack, C. C., Livingstone, M. S., Duffy, K. R., & Born, R. T. (2003). End-stopping and the aperture problem: two-dimensional motion signals in macaque v1. *Neuron*, *39*(4), 671–680.

Parker, A. J. (2007). Binocular depth perception and the cerebral cortex. *Nature Reviews Neuroscience*, *8*(5), 379.

Petavision. (n.d.). http://sourceforge.net/p/petavision/code/HEAD/tree/.

Pierrot-Deseilligny, C., Milea, D., & Müri, R. M. (2004). Eye movement control by the cerebral cortex. *Current opinion in neurology*, *17*(1), 17–25.

Reich, K. (2017). *Binokulare bildstatistik mit virtueller vergenz* (Unpublished master's thesis). Eberhard Karls Universit"at T"uebingen.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., & Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, *20*(10), 2526–2563.

Ruderman, D. L., & Bialek, W. (1994). Statistics of natural images: Scaling in the woods. In *Advances in neural information processing systems* (pp. 551–558).

Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, *36*(8), 1627–1639.

Schultz, P. F., Paiton, D. M., Lu, W., & Kenyon, G. T. (2014). Replicating kernels with a short stride allows sparse reconstructions with fewer independent kernels. *arXiv preprint arXiv:1406.4205*.

Simoncelli, E. P. (2003). Vision and the statistics of the visual environment. *Current opinion in neurobiology*, *13*(2), 144–149.

Srivastava, S., Orban, G. A., De Mazière, P. A., & Janssen, P. (2009). A distinct representation of three-dimensional shape in macaque anterior intraparietal area: fast, metric, and coarse. *Journal of Neuroscience*, *29*(34), 10613–10626.

Taira, M., Tsutsui, K.-I., Jiang, M., Yara, K., & Sakata, H. (2000). Parietal neurons represent surface orientation from the gradient of binocular disparity. *Journal of neurophysiology*, *83*(5), 3140–3146.

Tsutsui, K.-I., Jiang, M., Yara, K., Sakata, H., & Taira, M. (2001). Integration of perspective and disparity cues in surface-orientation–selective neurons of area cip. *Journal of Neurophysiology*, *86*(6), 2856–2867.

Verhoef, B.-E., Vogels, R., & Janssen, P. (2010). Contribution of inferior temporal and posterior parietal activity to three-dimensional shape perception. *Current Biology*, *20*(10), 909–913.

Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In *Computer vision and pattern recognition (cvpr), 2010 ieee conference on* (pp. 2528–2535).

Zhaoping, L. (2006). Theoretical understanding of the early visual processes by data compression and data selection. *Network: computation in neural systems*, *17*(4), 301–334.

Appendix A

My Additional Information

Derivation of Geometric Formulas

This section describes how the geometry from Figure 2.1 was derived, up to the mentioned formula 2.2.
With a and b given, angle γ can be calculated:

$$\gamma = \arctan \frac{b}{2a} \tag{A.1}$$

With γ one can calculate δ

$$\delta = \frac{\pi}{2} - \gamma \tag{A.2}$$

With a, b given s can be calculated with pythagoras:

$$s = \sqrt{a^2 + \frac{b^2}{4}} \tag{A.3}$$

The distances of x_r and x_l are sought. The simple equations stand:

$$x_r = f_e \cdot \tan \beta_r \tag{A.4}$$

$$x_l = f_e \cdot \tan \beta_l \tag{A.5}$$

On the other side of the lense one can again find β_r and β_l (opposite angles of the inner β). Because a dependence of α is sought to compute the β, one needs c_r, respectively c_l. For now, I write out the calculation of c_r.
Through the law of cosines one gets:

$$c_r^2 = s^2 + l^2 - 2 \cdot s \cdot l \cdot \cos\left(\alpha + \frac{\pi}{2} - \arctan \frac{b}{2s}\right) \tag{A.6}$$

$$c_r = \sqrt{s^2 + l^2 - 2 \cdot s \cdot l \cdot \cos\left(\alpha + \frac{\pi}{2} - \arctan \frac{b}{2s}\right)} \tag{A.7}$$

Because of the law of sines one then gets:

$$\frac{sin\beta_r}{l} = \frac{\sin\left(\frac{\pi}{2} - \arctan\left(\frac{b}{2a}\right) + \alpha\right)}{\sqrt{s^2 + l^2 - 2 \cdot s \cdot l \cdot \cos\left(\alpha + \frac{\pi}{2} - \arctan\frac{b}{2s}\right)}} \tag{A.8}$$

with

$$\sin\frac{\pi}{2} - x = \cos x$$

this makes

$$\frac{sin\beta_r}{l} = \frac{\cos\left(\arctan\left(\frac{b}{2a}\right) - \alpha\right)}{\sqrt{s^2 + l^2 - 2 \cdot s \cdot l \cdot \sin\left(\arctan\left(\frac{b}{2s}\right) - \alpha\right)}} \tag{A.9}$$

the trigonometric addition formulas state that:

$$\begin{aligned}
\sin(\alpha+\beta) &= \sin\alpha \cdot \cos\beta + \sin\beta \cdot \cos\alpha && \text{(A.10)}\\
\sin(\alpha-\beta) &= \sin\alpha \cdot \cos\beta - \sin\beta \cdot \cos\alpha && \text{(A.11)}\\
\cos(\alpha+\beta) &= \cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta && \text{(A.12)}\\
\cos(\alpha-\beta) &= \cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta && \text{(A.13)}\\
& && \text{(A.14)}
\end{aligned}$$

Therefore I can resolve the sums in the trigonometric functions:

$$\sin(\beta_r) = \frac{l \cdot \Bigg[\cos\Bigg(\arctan\bigg(\frac{b}{2a}\bigg)\Bigg) \cdot \cos(\alpha) + \sin\Bigg(\arctan\bigg(\frac{b}{2a}\bigg)\Bigg) \cdot \sin(\alpha)\Bigg]}{\sqrt{s^2 + l^2 - 2 \cdot s \cdot l \cdot \Bigg[\sin\Bigg(\arctan\bigg(\frac{b}{2a}\bigg) \cdot \cos(\alpha) - \sin(\alpha) \cdot \cos\Bigg(\arctan\bigg(\frac{b}{2a}\bigg)\Bigg)\Bigg)}\Bigg]} \tag{A.15}$$

In addition to that, the following stands:

$$\begin{aligned}
\tan(x) &= \frac{\sin(x)}{\sqrt{1-\sin^2(x)}}\\
\Rightarrow \tan^2(x) &= \frac{\sin^2(x)}{1-\sin^2(x)}\\
\Rightarrow \tan^2(x) - tan^2(x) \cdot \sin^2(x) &= \sin^2(x)\\
\Rightarrow \tan^2(x) &= sin^2(x) \cdot (1 + \tan^2(x))\\
\Rightarrow \sin(x) &= \frac{\tan(x)}{\sqrt{1+\tan(x)}}
\end{aligned}$$

The same can be done with the cosine:

$$\begin{aligned}\tan(x) &= \frac{\sqrt{1-\cos^2(x)}}{\cos(x)}\\ &...\\ \cos(x) &= \frac{1}{\sqrt{1+\tan^2(x)}}\end{aligned}$$

Following these two rules equation A.15 can be rewritten:

$$\sin\left(\beta_r\right) = \frac{l \cdot \left[\frac{1}{\sqrt{1+(\frac{b}{2a})^2}} \cdot \cos\left(\alpha\right) + \frac{\frac{b}{2a}}{\sqrt{1+(\frac{b}{2a})^2}} \cdot \sin(\alpha)\right]}{\sqrt{s^2 + l^2 - 2 \cdot s \cdot l \cdot \left[\frac{\frac{b}{2a}}{\sqrt{1+(\frac{b}{2a})^2}} \cdot \cos(\alpha) - \sin(\alpha) \cdot \frac{1}{\sqrt{1+(\frac{b}{2a})^2}}\right]}} \tag{A.16}$$

As in equation A.4 stated, I'm interested in $\tan(\beta_r)$. Therefore I make use of the rule:

$$\tan^2(x) = \frac{\sin^2(x)}{1 - \sin^2} \tag{A.17}$$

For that, equation A.16 is now squared:

$$\sin^2(\beta_r) = \frac{l^2 \cdot \dfrac{\cos^2(\alpha) + (\frac{b}{2s})^2 \cdot \sin^2(\alpha) + \cancel{2} \cdot \sin(\alpha)\cos(\alpha) \cdot \frac{b}{\cancel{2}a}}{1 + (\frac{b}{2a})^2}}{s^2 + l^2 - 2 \cdot s \cdot l \cdot \dfrac{\frac{b}{2a} \cdot \cos(\alpha) - \sin(\alpha)}{\sqrt{1 + (\frac{b}{2a})^2}}} \tag{A.18}$$

$$= \frac{l^2 \cdot \left(\cos^2(\alpha) + (\frac{b}{2a})^2 \cdot \sin^2(\alpha) + \frac{b}{a} \cdot \sin(\alpha) \cdot \cos(\alpha)\right)}{\left[(s^2 + l^2) \cdot \sqrt{1 + (\frac{b}{2a})^2} - 2 \cdot s \cdot l \cdot \left(\frac{b}{2a} \cdot \cos(\alpha) - \sin(\alpha)\right)\right] \cdot \sqrt{1 + (\frac{b}{2a})^2}} \tag{A.19}$$

$$= \frac{l^2 \cdot \left(\cos^2(\alpha) + (\frac{b}{2a})^2 \cdot \sin^2(\alpha) + \frac{b}{a} \cdot \sin(\alpha) \cdot \cos(\alpha)\right)}{(s^2 + l^2) \cdot (1 + (\frac{b}{2a})^2) - 2 \cdot s \cdot l \cdot \left(\frac{b}{2a} \cdot \cos(\alpha) - \sin(\alpha)\right) \cdot \sqrt{1 + (\frac{b}{2a})^2}} \tag{A.20}$$

(A.21)

With the short term

$$T = (s^2 + l^2) \cdot (1 + (\frac{b}{2a})^2) - 2 \cdot s \cdot l \cdot \left(\frac{b}{2a} \cdot \cos(\alpha) - \sin(\alpha)\right) \cdot \sqrt{1 + (\frac{b}{2a})^2} \tag{A.22}$$

A.17 can be applied and immediately simplified:

$$\tan^2(\beta_r) = \frac{\frac{l^2 \cdot \left(\cos^2(\alpha) + (\frac{b}{2a})^2 \cdot \sin^2(\alpha) + \frac{b}{a} \cdot \sin(\alpha) \cdot \cos(\alpha)\right)}{\cancel{T}}}{\frac{T - l^2 \cdot \left(\cos^2(\alpha) + (\frac{b}{2a})^2 \cdot \sin^2(\alpha) + \frac{b}{a} \cdot \sin(\alpha) \cdot \cos(\alpha)\right)}{\cancel{T}}} \tag{A.23}$$

$$= \frac{\left[l \cdot \left(\cos(\alpha) + \frac{b}{2a} \cdot \sin(\alpha)\right)\right]^2}{U - V - l^2 \cos^2(\alpha) - l^2(\frac{b}{2a})^2 \sin^2(\alpha) - l^2 \frac{b}{a} \sin(\alpha) \cos(\alpha)} \tag{A.24}$$

with

$$U = s^2 + l^2 + s^2 \cdot (\frac{b}{2a})^2 + l^2 \cdot (\frac{b}{2a})^2 \tag{A.25}$$

$$V = 2 \cdot s \cdot l \cdot \frac{b}{2a} \cdot \cos(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2} + 2 \cdot s \cdot l \cdot \sin(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2} \tag{A.26}$$

In a similar fashion $\sin(\beta_l)$ can be calculated with the help of c_l, leading to:

$$\tan^2(\beta_l) = \frac{\left[l \cdot \left(\cos(\alpha) - \frac{b}{2a} \cdot \sin(\alpha)\right)\right]^2}{U + V - l^2 \cos^2(\alpha) - l^2(\frac{b}{2a})^2 \sin^2(\alpha) + l^2 \frac{b}{a} \sin(\alpha) \cos(\alpha)} \tag{A.27}$$

with the same U and V.
Coming back to the beginning of the calculations it is now possible to describe A.4 and A.5:

$$x_r(\alpha, l) = f_e \cdot l \cdot \frac{\cos(\alpha) + \frac{b}{2a} \cdot \sin(\alpha)}{\sqrt{U - V - \left(l \cdot (\cos(\alpha) + (\frac{b}{2a}) \sin(\alpha))\right)^2}} \tag{A.28}$$

$$x_l(\alpha, l) = f_e \cdot l \cdot \frac{\cos(\alpha) - \frac{b}{2a} \cdot \sin(\alpha)}{\sqrt{U + V - \left(l \cdot (\cos(\alpha) - (\frac{b}{2a}) \sin(\alpha))\right)^2}} \tag{A.29}$$

with

$$U = s^2 + l^2 + s^2 \cdot (\frac{b}{2a})^2 + l^2 \cdot (\frac{b}{2a})^2$$
$$V = 2 \cdot s \cdot l \cdot \frac{b}{2a} \cdot \cos(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2} + 2 \cdot s \cdot l \cdot \sin(\alpha) \cdot \sqrt{1 + (\frac{b}{2a})^2}$$

Basis Functions at Different Overcompleteness levels

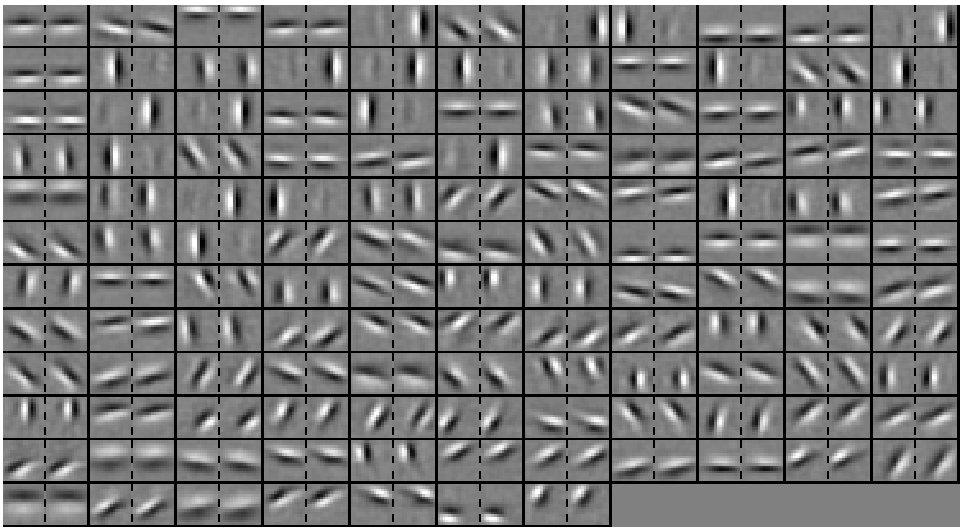


Figure A.1: The binocular basis functions, learned with the virtual vergence database after (Reich, 2017). The basis functions are sorted after their usage in reconstruction, the first being used most frequent. Through the overcompleteness level of 1 for stride $s = 8$ in both spacial directions, 128 basis function pairs emerge.

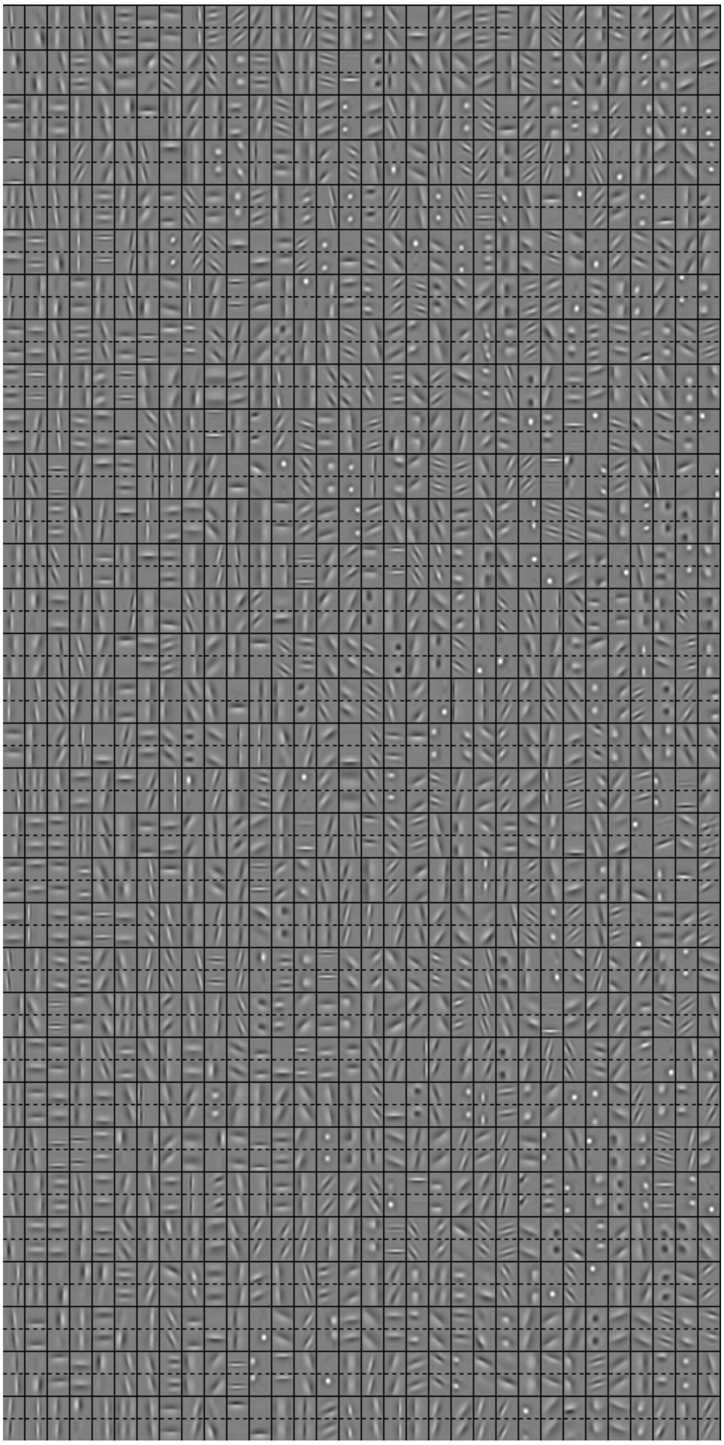


Figure A.2: The binocular basis functions, learned with the virtual vergence database after (Reich, 2017). The basis functions are sorted after their usage in reconstruction, the first being used most frequent. Through the overcompleteness level of 8 for stride $s = 8$ in both spacial directions, 1024 basis function pairs emerge.

Exemplatory ZED Stereo Camera Images

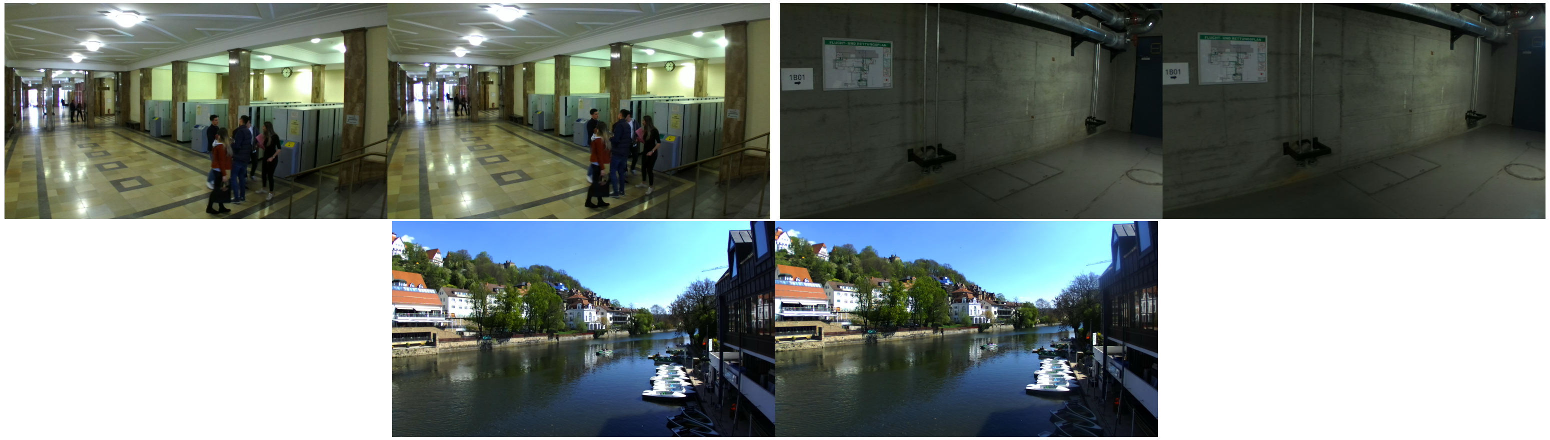


Figure A.3: Three exemplatory glued-together half-images from Reich (2017)

Additional Results

To test the significance of difference between the two groups of mean encoding neurons, first, the samples were z-transformed. The following Figures show the z-transformed data for all four conditions.

Mean Kernel Activity

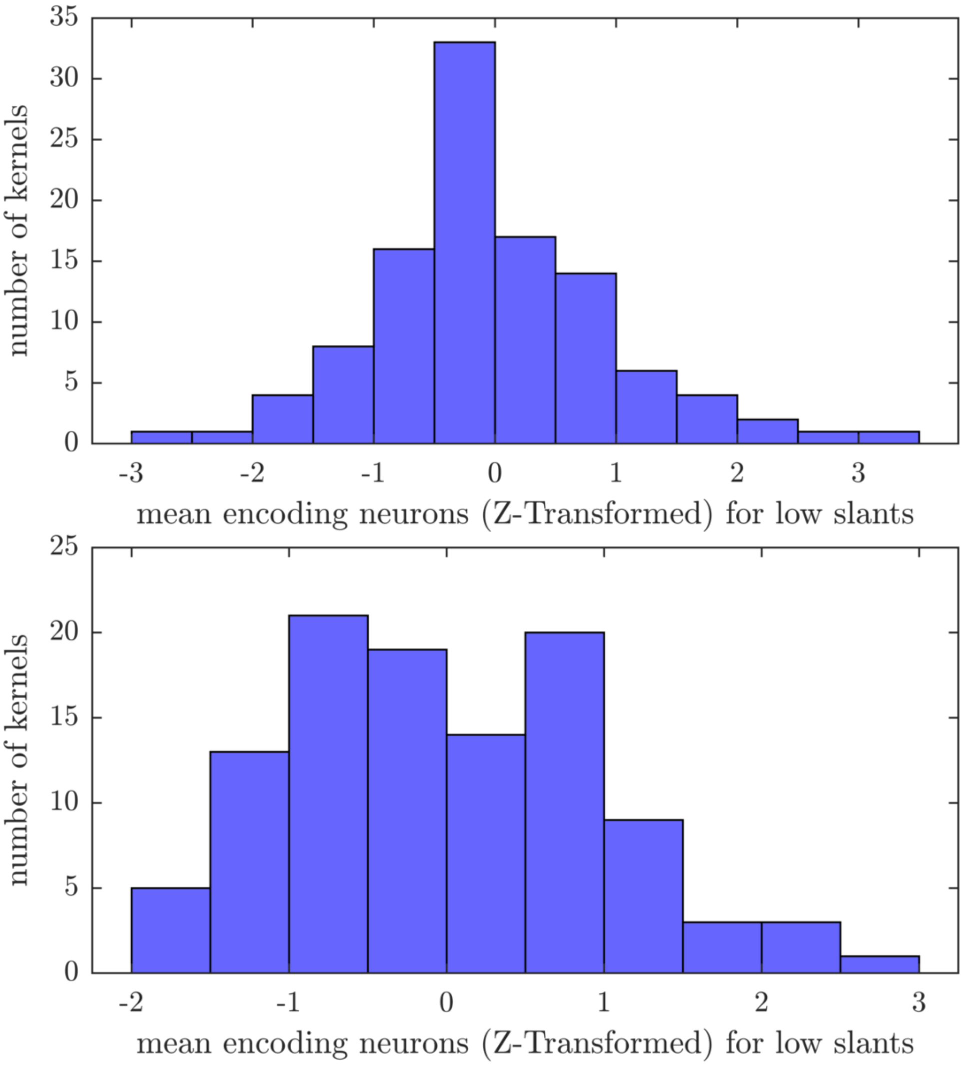


Figure A.4

The z-transformed data were tested on their normality with help of a Kolmogorov-Smirnov Test. Due to the fact, that all samples followed a normal distribution, a conventional two-sampled, one tailed t-test was carried out. Details can be found in the following two tables.

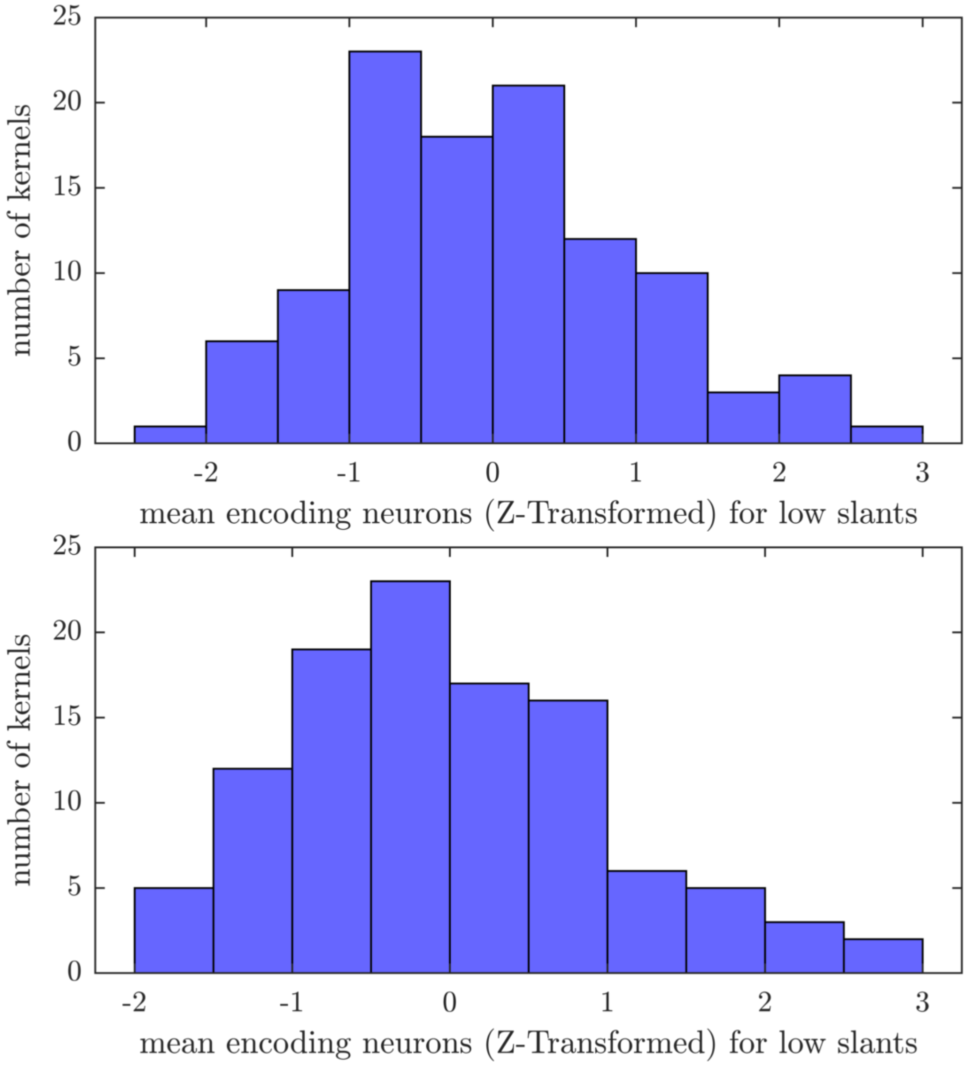

25
20
15
10
5
0
number of kernels
-2
-1
0
1
2
3
mean encoding neurons (Z-Transformed) for low slants
25
20
15
10
5
0
number of kernels
-2
-1
0
1
2
3
mean encoding neurons (Z-Transformed) for low slants

Figure A.5

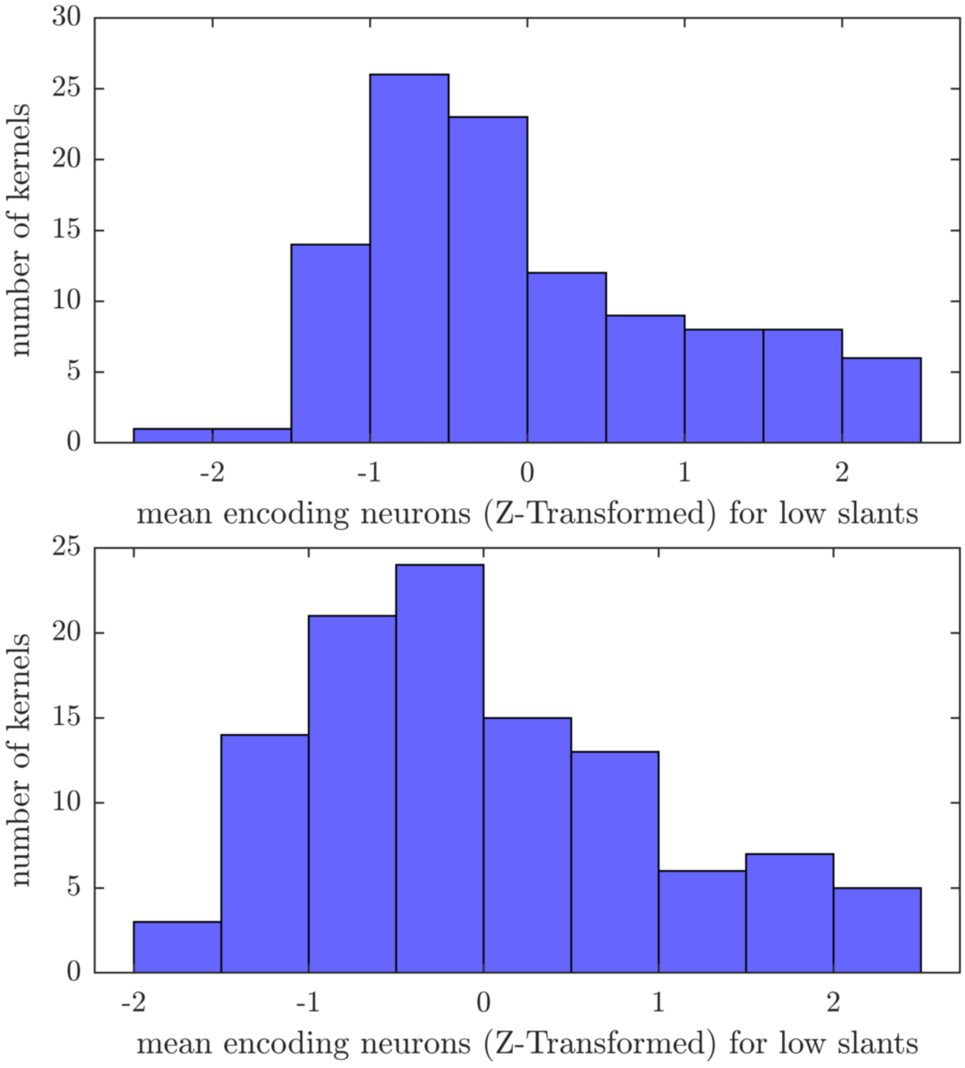

30
25
20
15
10
5
0
number of kernels
-2
-1
0
1
2
mean encoding neurons (Z-Transformed) for low slants
25
20
15
10
5
0
number of kernels
-2
-1
0
1
2
mean encoding neurons (Z-Transformed) for low slants

Figure A.6

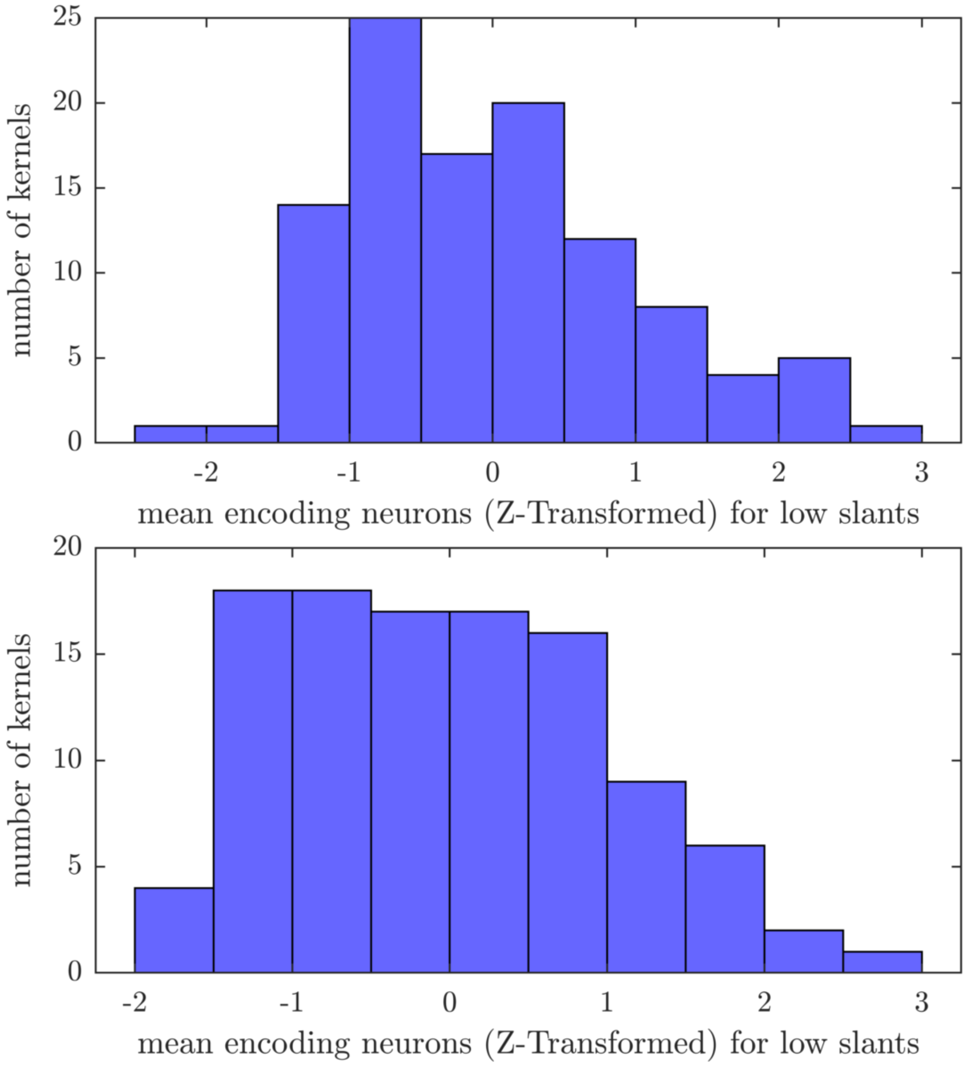

25
20
15
10
5
0
number of kernels
-2
-1
0
1
2
3
mean encoding neurons (Z-Transformed) for low slants
20
15
10
5
0
number of kernels
-2
-1
0
1
2
3
mean encoding neurons (Z-Transformed) for low slants

Figure A.7

	Z-Transformation				Kolmogorov-Smirnov Test			
	μ_{low}	μ_{high}	σ_{low}	σ_{high}	Hypothesis Decision Low	p_{low}	Hypothesis Decision High	p_{high}
O1, 5x5	29.1512	29.8655	0.2677	0.7589	0	0.2857	0	0.4093
O1, 7x7	55.0789	56.5554	0.4948	1.4618	0	0.2794	0	0.5122
O8, 5x5	30.3115	31.1229	0.2964	0.6697	1	0.0295	0	0.4130
O8, 7x7	57.3007	58.6106	0.4456	1.1041	0	0.5246	0	0.7129

	Two Sampled, One-Tailed T Test with unequal Variance			
	Hypothesis Decision	p	Confidence Interval	df
O1, 5x5	1	1.3929×10^{-17}	[0.5864,Inf]	133
O1, 7x7	1	5.2559×10^{-18}	[1.2304,Inf]	131
O8, 5x5	1	1.3842×10^{-22}	[0.6947,Inf]	147
O8, 7x7	1	3.9714×10^{-22}	[1.1201,Inf]	141

Additional Tilt Estimations

Tilt estimations at the lowest two slant levels can be seen in the following two Figures. The slant is too small to allow any inference.

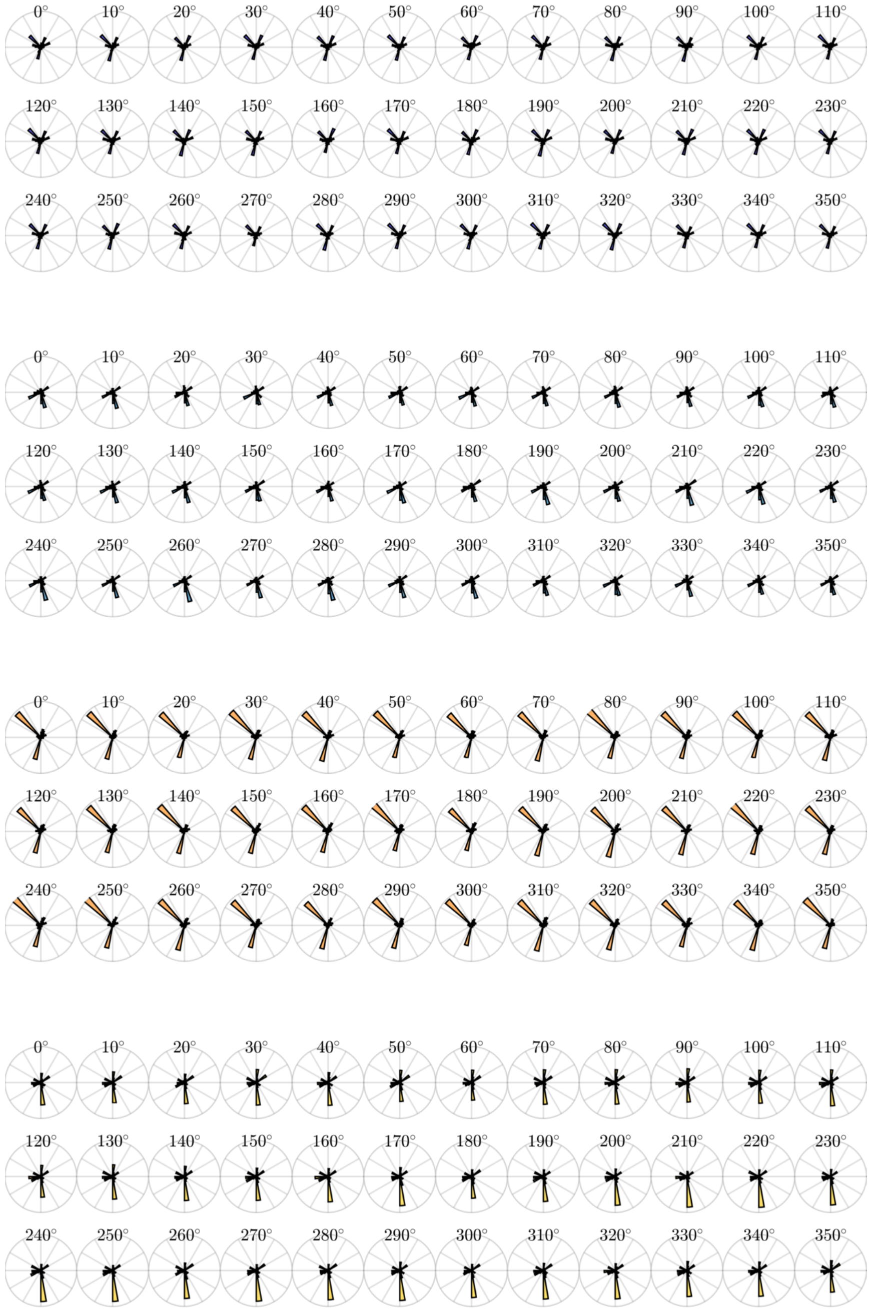


Figure A.8: Histograms of tilt estimations at all tilt levels at slant level 2 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

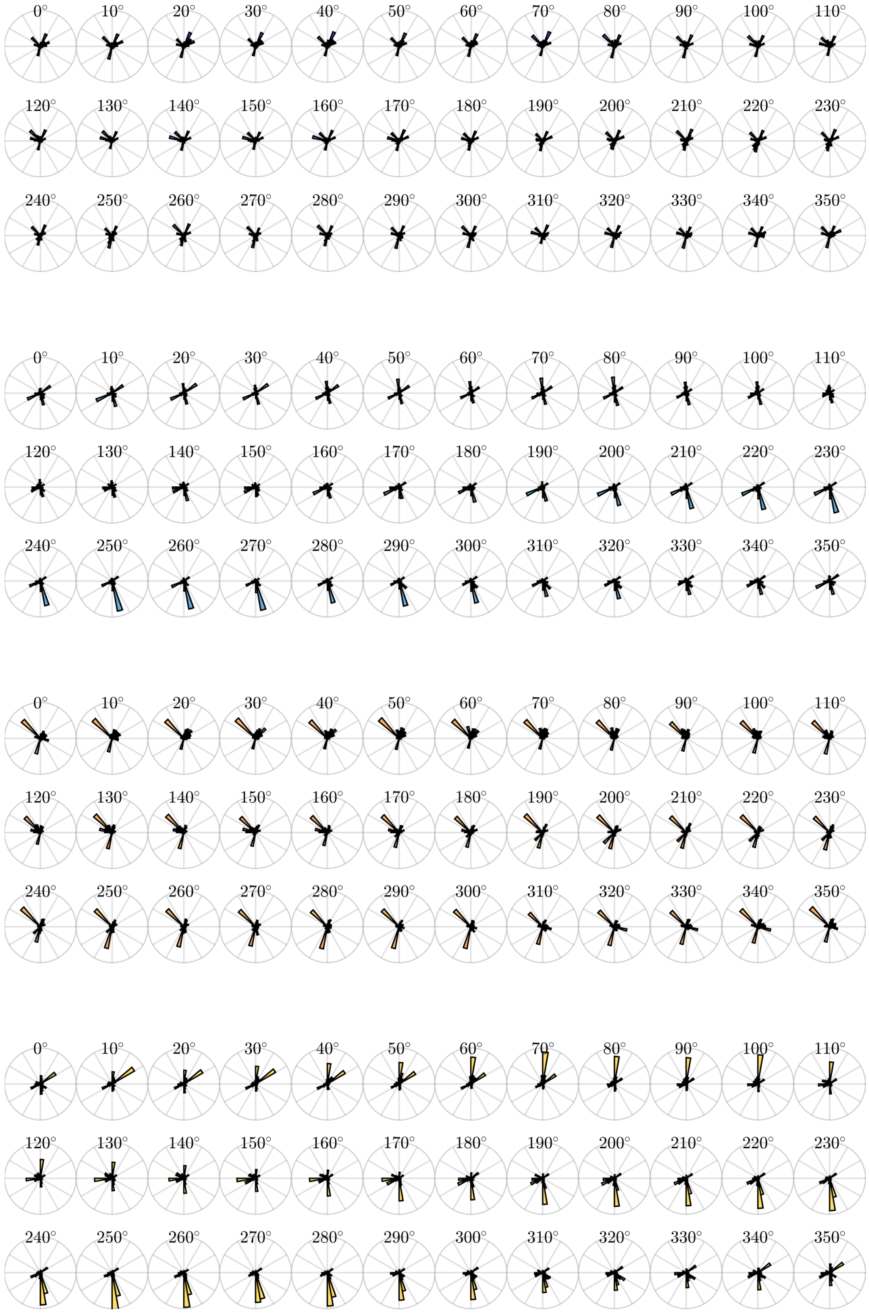


Figure A.9: Histograms of tilt estimations at all tilt levels at slant level 3 *blue*: O1,$m = 5$, with-bias; *green*: O1,$m = 5$, without-bias; *orange*: O1,$m = 7$, with-bias; *yellow*: O1,$m = 7$, without-bias

Appendix B

Contents of CD

1. A digital version of this work
2. A folder with all Matlab Scripts used for processing
3. A template .lua-script for SCANN learning
4. The Blender file with the reconstructed experiment
5. The python scripts used for stimulus creation automatisation
6. The basis functions for O1 and O8