==

Eberhard Karls Universität Tübingen

Mathematisch-Naturwissenschaftliche Fakultät
Institut für Neurobiologie
Lehrstuhl für Kognitive Neurowissenschaften

Bachelorarbeit

A sparse coding model of zebrafish ego-motion detection from optic flow

Johannes Hölscher

Gutachter: **Prof. Dr. rer. nat. Hanspeter A. Mallot**
Institut für Neurobiologie
Kognitive Neurowissenschaften

Betreuer: **Gerrit Ecke**
Institut für Neurobiologie
Kognitive Neurowissenschaften

Hölscher, Johannes:
A sparse coding model of zebrafish ego-motion detection from optic flow
Bachelorarbeit Kognitionswissenschaft
Eberhard Karls Universität Tübingen
Bearbeitungszeitraum: 2. Juni 2018 – 1. Oktober 2018

Erklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Tübingen, den 1. Oktober 2018

Abstract

Zebrafish (*Danio rero*) adjust the movement of their eyes and their body responding to motion. Kubo et al. (2014) described neurons in their area pretectalis, that are sensitive for translation and rotation directions and thus could implement the necessary detection of ego-motion from optic flow. Characteristic properties of those cells were recreated by Ecke et al. (2018) using a sparse coding network on optic flow input generated from a virtual reality simulation.
To examine, whether such sparse coding could be the basis for the ego-motion detection, we calculated likelihood functions for the activation of the model neurons and applied a maximum likelihood estimation. In this way, we obtained estimates of the translation axis containing high variation and a systematic bias, but fairly reliable estimates of the rotation axis.
These results indicate, that the present model does not suffice to recreate the process of ego-motion approximation in zebrafish. Yet there are positive aspects, that are suited to motivate further research on this application of sparse networks. Therefore, we provide starting points to improve the model and further examine the potential of sparse representations for modelling specific tasks like ego-motion detection.

Acknowledgements

First of all, I would like to thank my supervisor, Gerrit Ecke, for his understanding guidance and his friendly ear for my problems. His suggestions and his assistance were fundamental to my work.
I would also like to thank my reviewer, Prof. Dr. Hanspeter A. Mallot for the possibility to work on such an interesting project. He always had plenty of ideas when I came up to him with questions.
Further I want to thank all members of the department of cognitive neuroscience. The pleasant working atmosphere eased my efforts as well as the manifold exchange that has broadened by mind.
Moreover, I want to acknowledge those, who proofread my thesis in the last weeks: besides my supervisor, PD Dr. Gregor Hardiess, Sebastian Kenzler, Raphael Roos and Martin Schlimke substantially helped me to improve both content and language.
In addition, I am glad for the advice and the motivation I obtained from my friends and fellow students, who always made things look easier.
I also want to express my gratitude for the support of the Cusanuswerk foundation that did not only financially support my studies, but also inspired me by the interchange with its fellows.
More particularly, I would like to thank my parents, who assist me whenever it is needed. This accomplishment would not have been possible without their encouragement throughout the years.
Finally I want to thank everyone who has, directly or indirectly, supported me during the process of writing this thesis.

Contents

1. Introduction **1**
1.1. Sparse coding 1
1.2. Ego-motion detection from optic flow 3
1.3. Modelling ego-motion detection in zebrafish 4
1.4. Overview of the approach 5

2. Methods and results **9**
2.1. Experimental simulation and computation of optic flow 9
2.2. Whitening 11
2.3. Sparse coding network 13
2.3.1. Locally competitive algorithm 13
2.3.2. Network implementation 16
2.4. Receptive fields 20
2.5. Ego-motion estimation 21
2.5.1. Likelihood functions for kernel activities 21
2.5.2. Maximum likelihood estimates 27
2.6. Estimate evaluation 28

3. Discussion **39**
3.1. Summary of the procedure 39
3.2. Interpretation of the results 40
3.3. Pros and cons of the approach 41
3.4. Flaws and starting points 43
3.5. Conclusion 46

A. Supplementary material **47**

Bibliography **51**

1. Introduction

1.1. Sparse coding

One of the most promising approaches to modelling properties of neuronal processing are sparse coding networks. The basic principle of sparse codes is that data items are represented by the activation of only a few elements at one time in a large neural network.
What makes this approach so relevant to simulating neural processes, is that sparseness, that is low activity rates, seems to be enforced in animal brains: Froudarakis et al. (2014) could show that, in the visual cortex (V1) of mice, natural movies cause sparser activities than artificially generated ones. The selectivities of the neurons appear to be optimized to exploit statistical properties of natural data such as higher-order correlations: Realistic inputs, that fulfill these properties, are represented more efficiently. Also they found, that the sparsity of the representation predicted a better discriminability of the movie scenes and thus created a computational advantage.
The enforcement of sparsity in cortical areas was also indicated by Baddeley et al. (2001), who examined neuronal firing in the V1 of cats and the inferior temporal area (IT) of maquaque monkeys. They found sparse firing-rate distributions in both areas for realistic visual inputs and concluded, that the neural codes are designed to maximize the carried information and at the same time minimize the firing rate.
Inspired by the biological occurance of sparseness, artificial neural networks have been implemented in several models in order to bring deeper understanding of its functions. A groundbreaking application was presented by Olshausen & Field (1996): They designed a network to enforce just the two

principles information preservation and sparsity of activities, very similar to what Baddeley et al. (2001) suggested for neurons. This model sufficed to let receptive fields, that fulfill all the important response properties of biological neurons, emerge.

While these findings suggest the usage of sparse coding in classification systems, the higher computational costs do not necessarily pay off by enhancing the performance. Rigamonti et al. (2011) found that the recognition rate of their image classifying network was at least as good using a simple convolutional layer as it was using a sparse layer. Yet sparse coding was useful in learning the relevant features of an image and produced better filters than handcrafting or randomization.

While sparse coding networks apparently do not guarantee an improved performance, they provide other benefits that make them advantageous for neural representation and therefore relevant to models of neural processes. One important aspect is their efficiency in terms of energy consumption. Baddeley et al. (2001) proposed low firing rates as a restriction for the maximization of carried information. The importance of this constraint is strongly supported by Attwell & Laughlin (2001), who collected physiological data from the grey matter in rodents, in order to size up the components of their energy demand. Even for low firing rates of four spikes per second, action potentials cause 47% of the overall consumption, accounting for a higher fraction than the resting potential and other maintenance costs. As this amount increases with higher spike frequencies, firing rate broadly determines the overall demand. Therefore distributed neural codes with low activity ratios (e.g. $\leq$ 15% of the neurons active at the same time), are advantageous compared to codes employing less neurons, but higher firing rates. As sparse coding matches this description, it leads to less need for energy supply. Due to the high proportion of overall energy consumption by the brain, reductions of this challenge provide a vast benefit for animals in terms of survival.

A further advantage of sparse coding lies in their suitability for deriving statistical knowledge about the environment: for learning about probabilistic dependencies, events have to be reliably counted. Gardner-Medwin & Barlow (2001) proved, that for this task high redundancy is needed. Precise

counting requires much more neurons than just unambiguously representing these events. Also the overlap between the elements active for different events has to be small, which can be achieved by enforcing sparseness.
Based on these results, Barlow (2001) suggested sparse codes as the appropriate form of representation to detect redundancy in natural sensory inputs and to exploit its statistical implications. Furthermore, he argued that, in sparse networks, the information carried by a single neuronal activity is more relevant to related tasks and can be processed more efficiently.
Taking together these arguments and the advantage of sparseness in terms of energy efficiency, the findings of sparse codes in animal brains appear comprehensible. Therefore, and because they have already been used successfully for modelling neural properties, we decided to use a sparse coding approach for the present model.

1.2. Ego-motion detection from optic flow

We will apply the sparse coding approach motivated above to the simulation of ego-motion detection in zebrafish (*Danio rerio*) on the base of optic flow. This task, that is knowing in which direction one moves at any time, is a crucial skill for controlling position and posture while navigating a complex environment. Animals have developed different ways to collect information about their own motions, for example the vestibular system in most mammals. In many of them, the direction of movement is inferred from the shifts of the retinal image, called the optic flow (Nakayama, 1985).
For the computational solution of this task, several different models were developed to approximate ego-motion using analytical solutions or numerical optimization techniques (Raudies & Neumann, 2012). State-of-the-art models use unsupervised convolutional neural networks to estimate features such as depth and camera motion (Zhou et al., 2017).
Still the question arises, how this task is carried out on a neuronal level in animals nervous systems. In the case of bowflies, Krapp & Hengstenberg (1996) examined receptive fields of tangential neurons and found that some of them

extract patterns caused by either translations in a particular direction or rotations around a specific axis. Their receptive fields strongly resemble the optic flow fields caused by those ego-motion components. It appears questionable, whether a simple system like this could also be used in vertebrates.

1.3. Modelling ego-motion detection in zebrafish

To understand, and then model, the processes underlying ego-motion detection in animals, the responses of the relevant neuronal populations have to be examined in detail. This has been done for the larvae of zebrafish (*Danio rerio*), that use the detection of ego-motion from optic flow to generate optokinetic and optomotor responses. i. e. reactions to motions by moving eyes and body. The necessary processing seems to be performed in the area pretectalis of the fish. Kubo et al. (2014) examined the activation patterns of neurons in this area and found hierarchically organized, monocular and binocular neurons that perform specific operations in order to discriminate rotations and translations and to distinguish their direction. These neurons were categorized into a number of response classes to give an overview of their activation selectivities.
In order to model the response properties of the zebrafish pretectum, Ecke et al. (2018) made use of a sparse coding network that developed several filters extracting components of the optic flow. As the work treated in this paper is a resumption of that study, their methods will be further discussed in section II. Their results revealed that the features extracted by the networks kernels strongly resemble the findings of Kubo et al. (2014). Thus they showed that the ability of sparse coding to recreate neural properties also applies to the processing of optic flow.
Since Ecke et al. (2018) found similar response types as Kubo et al. (2014), it seems reasonable to assume their sparse coding network as a proper approach to model zebrafishs ego-motion detection. As this method is designed to make hidden correlations in the data explicit, it is likely to produce neurons that extract optic flow components caused by specific translations and rotations.

This kind of selectivities would provide a basis for the detection of the ego-motion direction, i. e. of the current axes of translation and rotation. If we could infer reliable estimates of these directions from the sparse representation, this would demonstrate that sparse coding is a promising approach not only to recreate neuronal response properties, but also to solve specific tasks using its representation of the input signal.

1.4. Overview of the approach

To answer the open question, whether sparse coding is the proper tool to model ego-motion detection in zebrafish, we employed a modified version of the sparse coding network used by Ecke et al. (2018) and evaluated the response of the model neuron with respect to the ego-motion direction. To obtain estimates of the direction, we then applied a maximum likelihood method. This delivered the axes of translation and rotation that are most likely to produce a given activity pattern.
This approach is biologically more plausible than others, as it reveals parallels to the model proposed by Burge & Geisler (2014) for the estimation of disparity in natural images. They created neurons representing Bayesian likelihoods for different disparities and took the one with the highest activation as an estimate. They could show, that this model only requires operations that neurons can perform. They also proved its optimality, as it increases selectivity to relevant and invariability to irrelevant aspects of the stimuli.
The processing steps of our model are visualized in figure 1.1. Its basic steps correspond to real-world counterparts: The virtual reality simulation is designed to resemble a realistic surrounding of zebrafish and the images taken during the simulated movements of the fish are projected on a schematic model of its retina. The local optic flow vectors calculated from these images parallel the retinal ganglion cells of zebrafish, which are selective for local movement. According to Kubo et al. (2014), these cells are directly wired to neurons of the area pretectalis (APT), which is modelled by the means of a

sparse coding network. However, we added a principal components analysis as a preprocessing step to facilitate the sparse coding. The main steps we added in the present study are, firstly, the computation likelihood functions of the neural activities from the output of the sparse network, and, secondly, the estimation of the translation and rotation axis of the ego-motion by maximum likelihood estimation. By these means, we examine, whether sparse coding can simulate the ego-motion detection from optic flow in zebrafish in an effective way.

The details of the model and the results we obtained from our simulation will be described in the following chapter.

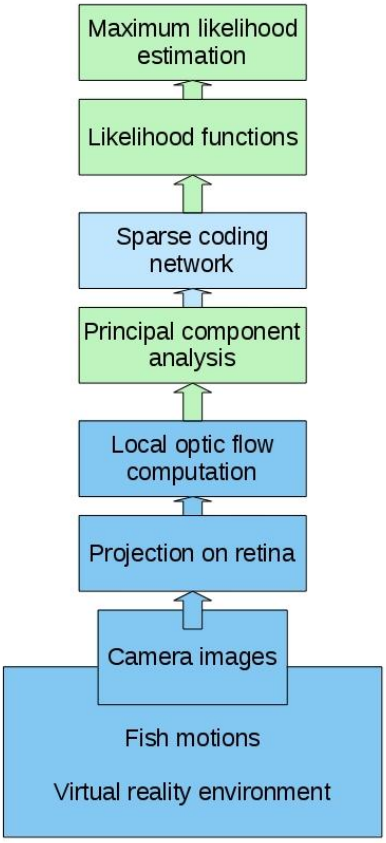


Figure 1.1.: Overview of the model steps

For the steps coloured deep blue, we worked on data created by Ecke et al. (2018), as a recomputation would require high computational effort. For the sparse coding network, the data were recomputed in a similar manner to their model, changing only details of the implementation. The green steps were added by us to enhance performance and to generate the direction estimates.

2. Methods and results

2.1. Experimental simulation and computation of optic flow

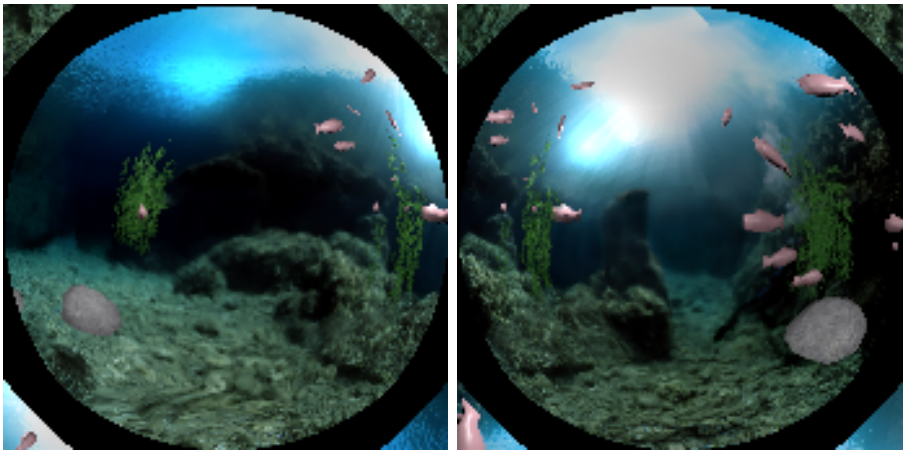


Figure 2.1.: The fishs' view on its virtual environment.

The two pictures show the images taken by the virtual cameras representing the left and the right eye of the fish, after being projected onto the simulated retina and therefore distorted.

In order to generate realistic input data for their sparse coding model, Ecke et al. (2018) simulated realistic images of the zebrafishs view on typical scenes. For this purpose, they created a virtual reality simulation of a fish tank containing objects like stones and plants as well as other fish and muddy or clear water. The scenario was programmed using the 3D creation suite Blender (release v2.76)[1]. The eyes of the fish were modelled by two cameras that captured the scenes as shown in figure 2.1. The observer in the simulated

[1] https://www.blender.org/

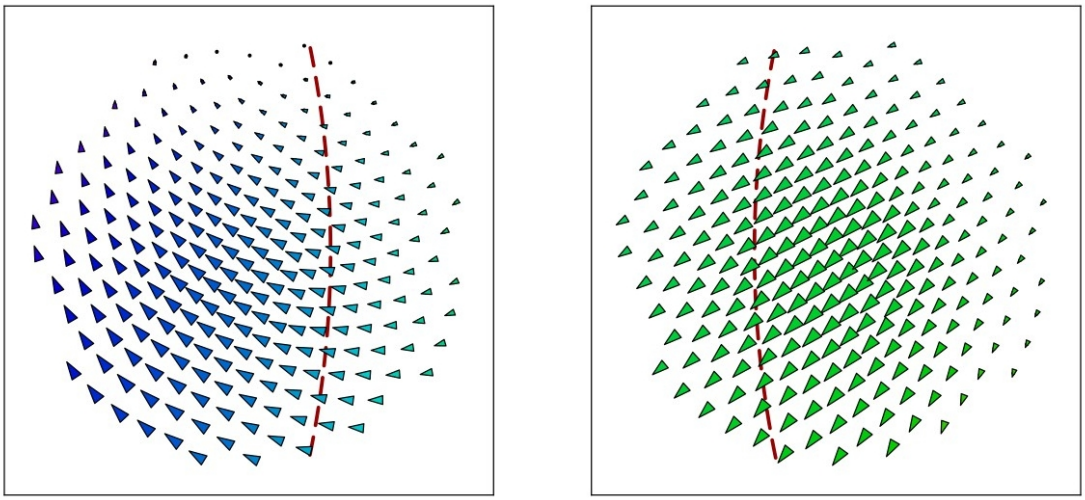


Figure 2.2.: Flowfield computed for a motion of the fish in the virtual reality.

The local motion computed from the input of the simulated retina is illustrated by vectors at the locations of retinal ganglion cells. The colour corresponds to the orientation, the size to the norm of the vectors. The images show the optic flow for the left and the right eye, with a dashed red line indicating the overlap of the fields of view.

scene was placed randomly and moved by a random force composed of six values for the three dimensions of rotation and translation, that were drawn independently from a uniform distribution with zero mean. After the impulse driven by the resulting force, the movement abates exponentially, while two sequent camera shots of the scene are captured.

The zebrafish retina was modelled by a half-sphere covering 160 degrees of view on which 256 points are equally distributed by a simple repellance algorithm. The image was projected onto the points by stereographic projection and sampling, resulting in inputs as displayed in figure 2.1.

Corresponding to the direction-tuned retinal ganglion cells of zebrafish, the local optic flow between the two images was computed using the software FlowNet 2.0[2] created by Ilg et al. (2017). It was represented by two signed variables, one for up and down and one for left and right. The values resulting from a movement can be depicted in a flow field as in figure 2.2 by combining the two variables in one vector for every retinal ganglion cell.

[2]https://github.com/lmb-freiburg/flownet2

2.2. Whitening

In order to make the sparse coding algorithm more efficient, we preprocessed the optic flow output before feeding it into the sparse coding network. One up-down and one left-right value for each of the 256 simulated optic flow detectors of each eye lead to data vectors of $2 \cdot 2 \cdot 256$ elements. These raw data contained correlated variables with differences in individual variance and a high amount of noise. As this makes their proper representation a complex task, we whitened them using a principal components analysis (PCA).
Like our further computations, unless specified differently, the whitening was carried out in the statistics software MATLAB (release 2018a)[3]. The build-in function *pca* contains a centering of the data, i. e. subtracting the means of each variable from the values and thus bringing them to zero mean. Subsequently, the PCA is implemented by the means of a single value decomposition (SVD). Hyvärinen & Oja (2000) used the same method as preprocessing for their independent component analysis algorithm, which has some parallels to sparse coding approaches. The description of the SVD is inspired by their publication.
As first step of the SVD whitening, the covariance matrix $E(xx^T)$ for the input data set x is computed and then decomposed into the orthogonal matrix E containing the eigenvectors and the diagonal matrix D of the eigenvalues, giving $E(xx^T) = EDE^T$. Finally, the data vectors are whitened by the linear transformation

$$\tilde{x} = ED^{-1/2}E^T x. \tag{2.1}$$

It can be shown that then one has $E(\tilde{x}\tilde{x}^T) = I$, i.e. the variance of each variable is 1 and the covariance between different variables is always 0. Hyvärinen & Oja (2000) demonstrated that the further analysis of correlations by their algorithm, or analogously by a sparse coding network, is substantially simplified, as the number of parameters is reduced.
The complexity was further diminished by discarding components with low eigenvalues, that is those that explain only a small amount of the overall

[3] https://de.mathworks.com/products/matlab.html

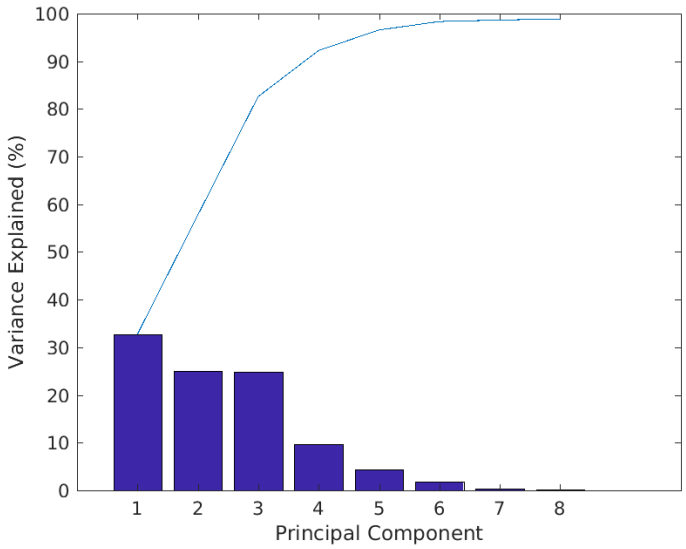


Figure 2.3.: Amount of variance explained by the highest-ranked components.

variance. We found that a small subset of the components could account for most of the variance, as signified by figure 2.3. As the 64 components explaining the highest amount of variance together accounted for 99.8% of the overall variance, the remaining components were considered noise and discarded from the representation. This did not only provide a more compact representation, but also removed noise from the optic flow data.
In order to illustrate the components resulting from the PCA, we computed the coefficients of the original flow variables for the calculation of each component. These weights can be depicted in a similar way as the original flow fields resulting from each movement. The illustrations for the six components explaining the highest amount of variance are supplied in figure A.1. They match very well with flow fields resulting from specific translations and rotations and thus appear suited for efficiently describing our flow field data.
After the whitening had removed covariance between the variables, brought the individual variances to the same level, removed noise factors and reduced the number of variables, the data constituted proper input to be efficiently processed by the sparse coding network.

2.3. Sparse coding network

2.3.1. Locally competitive algorithm

For the purpose of simulating the activities of pretectal neurons in zebrafish, Ecke et al. (2018) compared the results of two theoretical approaches: a sparse coding network for unsupervised learning and a backpropagation network with three hidden layers for supervised learning. The second approach thus contained estimates for ego-motion, while the sparse one did not. As the distribution of response types in the sparse coding network fitted much better with the ones found by Kubo et al. (2014), we decided to generate estimates based on the activites of that network.
The sparse coding network is based on the locally competitive algorithm (LCA) as proposed by Rozell et al. (2008), who modified the approach of Olshausen & Field (1996). The original authors used it to model receptive field properties of simple-cells in image processing and therefore assumed images, i. e. patterns of pixels, as input for their formulation of the algorithm. However, the implementation in this model processes input patterns consisting of the coefficients of the PCA components presented above.
The starting point for the LCA is the assumption, that any input pattern $I(x, y)$ can be represented by a linear superposition of basis functions $\phi_i(x, y)$:

$$I(x, y) = \sum_i a_i \phi_i(x, y) \tag{2.2}$$

The goal of the algorithm is to find such basis functions that lead to minimal entropies of the coefficients a_i for sets of natural input patterns, making the statistical dependencies in the input explicit. The LCA enforces two principles: For one thing, maximizing the quality of reconstruction of the original pattern, and for another thing maximizing the sparseness of its representation. The second principle implies minimizing the number of non-zero coefficients a_i used for the reconstruction of the input by the basic functions.

To combine those goals, both of them are captured in the error function

$$E = -\frac{1}{2}\,[preserve\ information\ ] - \lambda\,[sparseness\ of\ a_i], \tag{2.3}$$

that has to be minimized during the learning process. The factor λ, sometimes called sparsity, determines the balance between the two parts. The reconstruction part is evaluated by the mean squared error between reconstructed pattern and the input $I(x, y)$:

$$[preserve\ information\ ] = -\sum_{xy}[I(x,y) - \sum_i a_i\phi_i(x,y)]^2. \tag{2.4}$$

The sparsity part of the error function is the sum of the coefficients after modifying them through a cost function $C(x)$:

$$[sparseness\ of\ a_i] = -\sum_i C(a_i) \tag{2.5}$$

Olshausen & Field (1996) choose the cost function with the aim to minimize the ℓ_1-norm, i. e. the sum of the coefficient values. However, we deploy

$$C_\lambda(x) = \frac{\lambda}{2}\mathcal{H}(x-\lambda) = \begin{cases} \frac{\lambda}{2} & \text{for } x > \lambda \\ 0 & \text{else.} \end{cases} \tag{2.6}$$

This is taken from the hard-thresholding LCA from Rozell et al. (2008) and simplified under the assumption of positive a_i. This cost function leads to minimizing the ℓ_0-norm, i.e. the number of coefficients greater than zero. Eventually we obtain the error function

$$E = \frac{1}{2}\sum_{xy}[I(x,y) - \sum_i a_i\phi_i(x,y)]^2 + \frac{\lambda^2}{2}\sum_i \mathcal{H}(a_i - \lambda). \tag{2.7}$$

In the network implementation of the algorithm, each basis function ϕ_i is associated with a computational unit, called neuron. Each neuron has an internal state u_i, which is changed continuously until the network reaches an

equilibrium solving the differential equation

$$\dot{u_i} = \frac{1}{\tau}[b_i - \sum_{j \neq i} C_{ij} a_j - u_i], \tag{2.8}$$

where τ controls, how fast the u_i are adapted. This equation also contains the principles mentioned above: the positive summand $b_i = \sum_{x,y} \phi_i(x,y) I(x,y)$ represents the fitting of the original pattern I and the basis function ϕ_i. In order to maximize the quality of reconstruction, the contribution of those basis functions fitting the original pattern have to be raised. The first negative summand $-\sum_j C_{ij} a_j = -\sum_j [\sum_{x,y} \phi_i(x,y)\phi_j(x,y)] a_j$ expresses in network terms a lateral inhibition that guarantees for not employing several basis functions for the same properties of the original pattern. The last summand $-u_i$ gives the neuron the properties of a leaky integrator, i. e. it looses a certain amount of its activation over time. By the interaction of these summands, the adjustment of the inner states minimizes the error function: $\dot{u_i} \propto -\frac{\delta E}{\delta a_i}$
The coefficients a_i are derived from the inner state u_i of the model neuron similarly to the action potential in biological neurons: as soon as the inner excitation exceeds a certain threshold, the coefficient is raised to a non-zero value. This behaviour is defined by the thresholding function $T_\lambda(x)$, which is connected to the cost function $C(x)$. In the case of the hard-thresholding LCA, it is

$$a_i = T_\lambda(u_i) = \frac{u_i}{1 + e^{-\gamma(u_i - \lambda)}.} \tag{2.9}$$

As this function is monotonically increasing, the a_i are changed to minimize the error function, because they follow the u_i. For $\gamma \to \infty$, the chosen thresholding function results in $a_i = 0$ for $u_i < \lambda$ and $a_i = u_i$ for $u_i > \lambda$. Thus, when an input pattern is presented, the rise of the u_i is faster for those ϕ_i that match better with the input. Therefore, they are the first to reach the threshold and cause non-zero coefficients a_i and subsequently start to inhibit units with similar basis functions. This mechanism is what Rozell et al. (2008) call local competition.
Once the activations have reached the equilibrium state, the reconstruction $\hat{I}(x,y) = \sum_i a_i \phi_i(x,y)$ build by them is used to adjust the basis functions ϕ_i.

This is implemented by the learning rule

$$\Delta\phi_i(x, y) = \eta \cdot a_i[I(x, y) - \hat{I}(x, y)]. \tag{2.10}$$

This means that every basis function ϕ_i is changed by the difference between reconstruction and original pattern, and in proportion to the learning rate η and to its contribution to the reconstruction manifested in the coefficient a_i. Thus, those basis functions that are most similar to the input, adapt to it and thereby reduce aspects that are represented incomplete by the ϕ_i.
By means of this procedure, basis functions are learned that enable a sound, but sparse representation of the original optic flow component patterns.

2.3.2. Network implementation

The locally competitive algorithm, with the goal to build a sparse representation of the optic flow stimuli, was carried out by a modification of the artificial neural network build by Ecke et al. (2018) in the neural simulation toolbox Petavision[4], that was developed by Schultz et al. (2014).
In order to represent the input, the basis functions, their coefficients and the equations of the algorithm, the network is constructed of several layers and their connections, as depicted in figure 2.4. The inner states u_i are represented as the activations of the elements of the simulated pretectum, which in petavision is called the V1 layer. The basis functions ϕ_i are represented as weights between the V1 layer and the error layer. There is a separate weight stored for the connection between each V1 neuron and each neuron of the error layer in order to obtain receptive fields capturing the whole binocular field of view.
The input data were the 64 optic flow components delivered by the principal components analysis for each of the 77076 motion fields. We tested networks with different numbers of simulated pretectal neurons: One dataset was created using 512 elements, giving an overcompleteness level of 0.5, i.e. the number of neurons is 0.5 times the number of explained variables, here the

[4]http://petavision.github.io

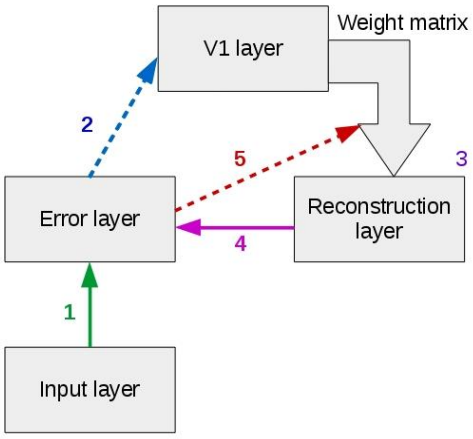
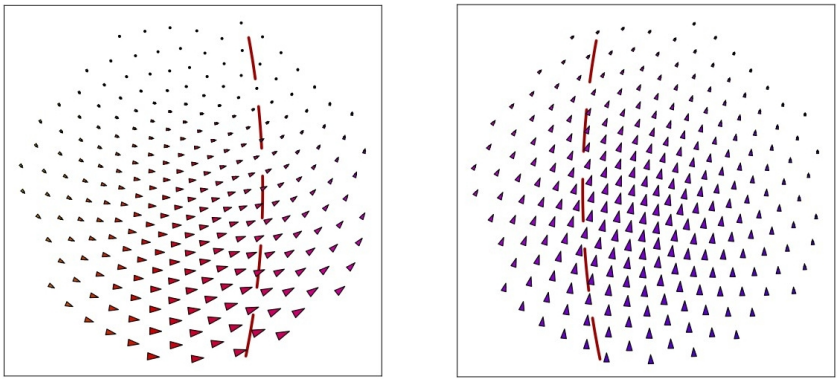


Figure 2.4.: Components and processes of the sparse coding layer

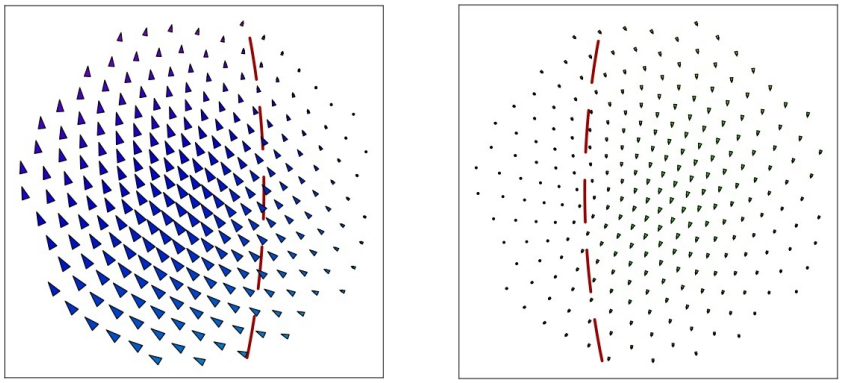
Illustration of the components of the sparse coding network and the processes carried out on each input. When an input pattern is given to the network, first it is copied into the error layer (step 1). In step 2 the activations of V1 are calculated using the basis functions stored in the weight matrix, until they reach the equilibrium from equation (2.8). After the V1 activations are stable, the coefficients derived form them are used together with the stored basis functions to build a reconstruction $\hat{I}$ of the original pattern (step 3). This reconstruction is afterwards subtracted from the activations of the error layer, with the remaining values representing the difference between original and reconstructed pattern (step 4). In a last step (step 5) this error pattern is used to modify the stored basis functions according to learning rule of the locally competitive algorithm (2.10).

$2 \cdot 2 \cdot 256$ up-down and left-right components of the optic flow data points of both eyes. The other run was carried out with 4096 neurons, resulting in an overcompleteness of 4.0 and thus enabling more redundant representations. In the further steps of evaluation we will compare the results of these different settings and thereby analyze, whether additional redundancy offers a benefit to the representation.

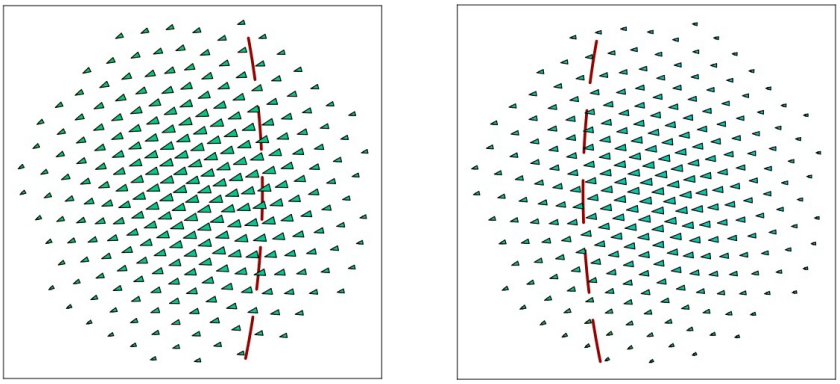
By the computational steps depicted in figure 2.4 that are carried out for several presentations of the principal component representations of all the motion fields in the dataset, the basis functions adapt to the structures of the input and form a sparse code for it. Their properties and their use for computing an estimate of translation direction and rotation axis will be treated in the following chapters.



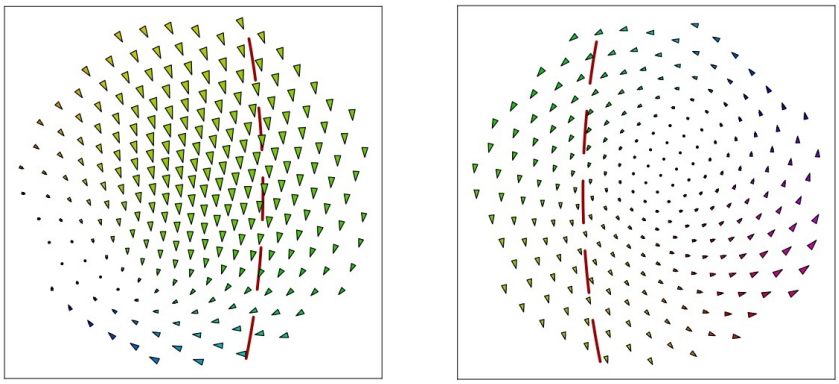
(a) The averagely most active neuron shows a typical receptive field: it is binocular, almost global, for each eye quasi unidirectional and there is no obvious coherence between the directions for the two eyes.



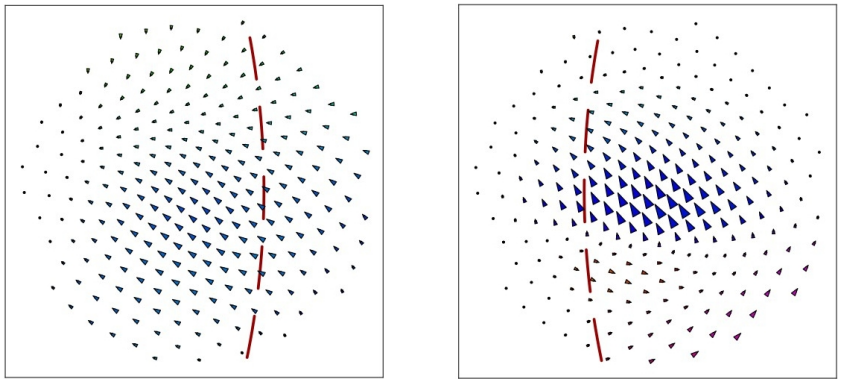
(b) The neuron in sixth rank of the most active kernels reveals an monocular, almost global, unidirectional pattern. These kind of flowfields appear about once in ten neurons.



(c) Neuron number eight exhibits a binocular, global, and across both eyes monodirectional flow field. This kind of patterns also makes roughly one-tenth of the dataset.



(d) A binocular flow field, with a mostly unidirectional pattern for the left eye and a spiral one for the right eye. Patterns like this one from the neuron ranked twelfth appear occasionally in the dataset.



(e) Some of the flowfields show smaller areas of high sensitivity, like these fields from the neuron in rank 60 according to mean activity.

Figure 2.5.: Typical optimal flowfields of model neurons

Flowfields of five of the most active model neurons for overcompleteness level 4.0, demonstrating properties generally common across the whole population. The amount, to which the retinal ganglion cells contribute to the activation of the particular neuron, is illustrated by the size of the vector in the respective location, while its orientation and colour represent the locally preferred direction. The fields are drawn for the left and the right eye, with dotched red lines indicating the overlap of the fields of view.

2.4. Receptive fields

The response characteristics of biological neurons are usually described by specifying their receptive fields, i.e. the subset of the sensory space, for which the neuron shows a response by firing stronger. We can also examine this for the simulated APT neurons of our model, where the stimulus, that would cause a maximum activation in the neuron, is defined by values of the $2 \cdot 2 \cdot 256$ optic flow variables corresponding to the simulated retinal ganglion cells of both eyes.

They can be represented by a field of flow vectors, taking together the up-down and left-right component and placing a two-dimensional vector at the position of the respective motion detector, resulting in a motion field as depicted in figure 2.5.

The basis for computing these motion fields are the weights from the pretectum layer to the error layer, which form the representation of the basis functions of the locally competitive algorithm. Rozell et al. (2008) showed that these weights can be interpreted as receptive fields of the sparse layer neurons.

As we did not perform the sparse coding algorithm directly on the optic flow data, but performed a principal component analysis before, we only obtained the influencing principal components from the weight matrix. We therefore reversed the scaling and converted them back into the original variables, which can be done easily as all the operations performed on the data were linear transformations.

Among the receptive fields obtained by this procedure there are quite different forms, but some tendencies are noticeable. Five typical examples for the flowfields generated with overcompleteness ratio 4.0 are shown in figure 2.5. Some of the neurons only react to signals derived from one eye, but most of them are binocular, sometimes showing similar patterns on both eyes, yet largely without any obvious relation between the patterns of left and right eye. The flow fields were mostly unidirectional, while a few contained circular or helical patterns around a center. They usually spanned broad parts of the visual field, more local ones formed exceptions. The observations for

overcompleteness 0.5 are generally similar, but contain more monocular patterns, yet not as many as binocular ones.
The flow fields of the artificial neurons broadly resemble the actual flow fields derived from the simulation. This fact demonstrates that the basis functions adapt to the input data. To examine, if this adaptation leads to a code suited for inferring the direction of translation and rotation from the activation of the neurons, we applied the maximum likelihood method discussed in the chapters below.

2.5. Ego-motion estimation

2.5.1. Likelihood functions for kernel activities

After our network implementation of the locally competitive algorithm resulted in a sparse activity code for the optic flow, the directions of translation and rotation were estimated just from the activity pattern of the sparse layer. This was done by a maximum likelihood method, with the probabilities calculated by the naive Bayesian rule. As described below, the respective direction of translation and rotation, that is most likely to account for the given activity pattern, is derived from the likelihood of every kernel to be active if the fish is moved in a particular direction.
Thus the estimation requires a likelihood function for every neurons activity, i. e. the likelihood of neuron x_i to be active given the direction ω_k. This is derived from the relative frequency of the activation of the neuron for the movements in each direction:

$$p(x_i|\omega_k) = \frac{h(x_i \wedge \omega_k)}{h(\omega_k)} \tag{2.11}$$

Under this approach, the magnitude of activation, that is the value of a_i in the locally competitive algorithm, is ignored. The only matter is, if a_i is zero or greater than that. The goal is to simplify the computation of the probabilities.

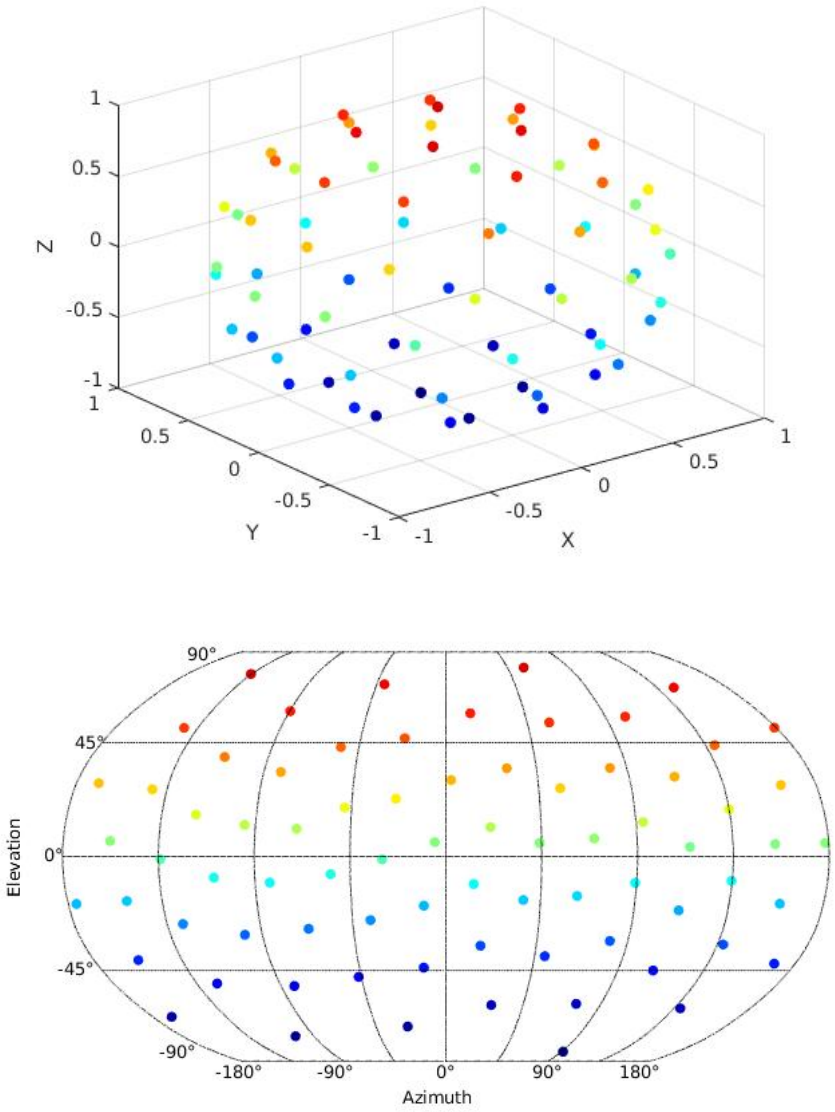


Figure 2.6.: Points selected as centres for binning

The centres of the 75 bins on the unit sphere, plotted in the 3D space (top) and in a Robinson projection on a 2D plain. While the origin of the projection represents axes pointing forward, positive x indicates axes going to the left, the horizontal borders show those going backwards, and positive y specify upwards axes pointing. This projection, usually applied to global maps, was used to illustrate several properties of the spherical data.

In order to obtain the relative frequencies, the data had to be split in distinct directions x_i, for which the neural activities could be counted. As the dimensions of translation and rotation had been drawn from continuous distributions, there were no such discrete categories. Therefore, the data were binned in groups of similar direction.
A simple repellence algorithm generated 75 points evenly spread on the surface of the unit ball, which was necessary in order to form groups occupying equal amounts of all possible directions. The result was the set of points shown in figure 2.6, that constituted the centres of the bins.
Afterwards, we computed the axes of translation and rotation from the position data of each motion in our Blender simulation. We then assigned the translational and rotational component of the motion independently to the group of that centre point, which was located closest to the unit vector of the respective axis.
Since in the simulation of the motion sequences the values for the six dimensions of translation and rotation were drawn independently from uniform distributions, the overall distribution of translation and rotation was a cube, not a sphere. This lead to a higher amount of translations and rotations with axes in the direction of the edges of the unit cubes. As a result, the bins that were located there contained more data, as shown in figure 2.7.
With the translations and rotations assigned to 75 distinct groups, one translation and one rotation likelihood function for each neuron were computed according to equation (2.11). These functions can be visualized as spherical heatmaps, like in figure 2.8.
For the run with overcompleteness level 4.0, the heatmaps for the likelihood function of rotations usually possess one clear local region of high activity, but rarely two or more, including about one to five of the 75 bins for which the neuron is active clearly more often than for the rest. By contrast, the translation fields rather contain two to five such regions that stand out less clearly, and only infrequently a more clear pattern with just one.

This difference also became visible, when we took the value of the most likely direction for each kernel and calculated the mean across all directions: For

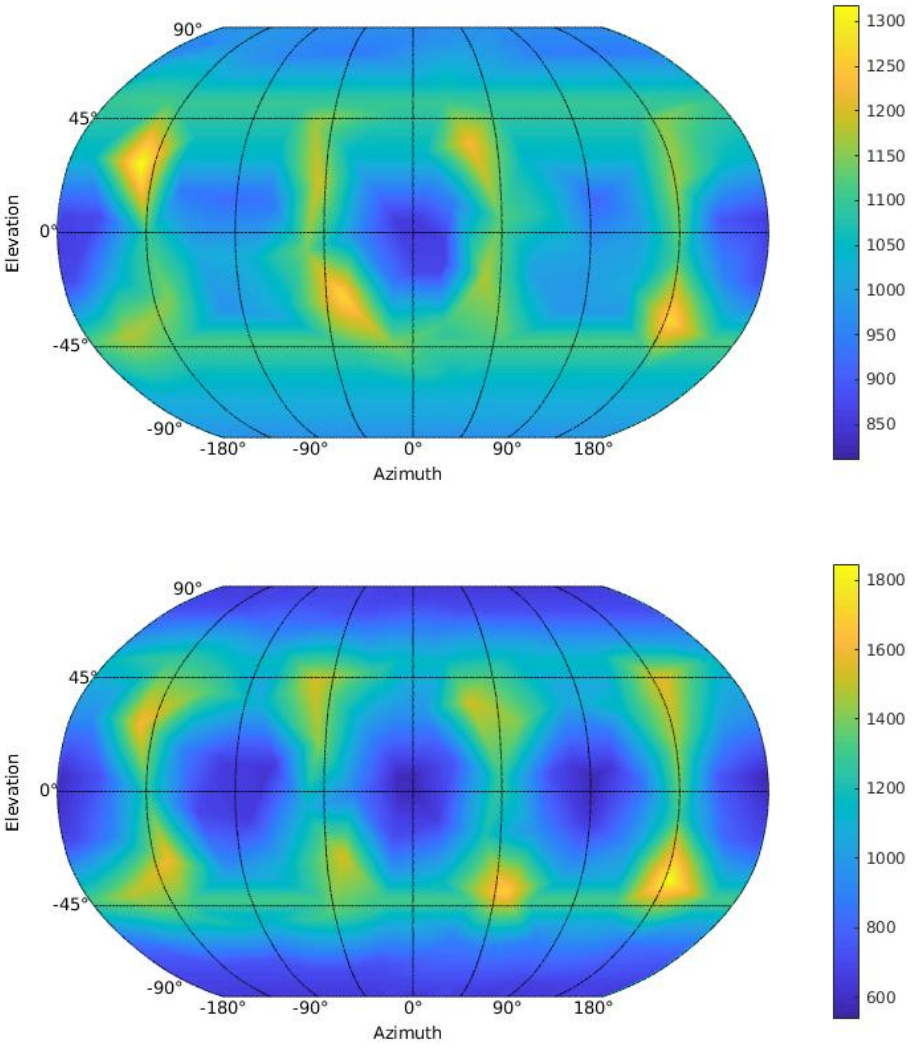


Figure 2.7.: Distribution of the directions of translation (top) and rotation axes (bottom) within the input data.

The plots were created by interpolating between the centres of the bins on the unit sphere and projecting onto a plain as mentioned in figure 2.6. The colour represents the absolute number of input data per bin.

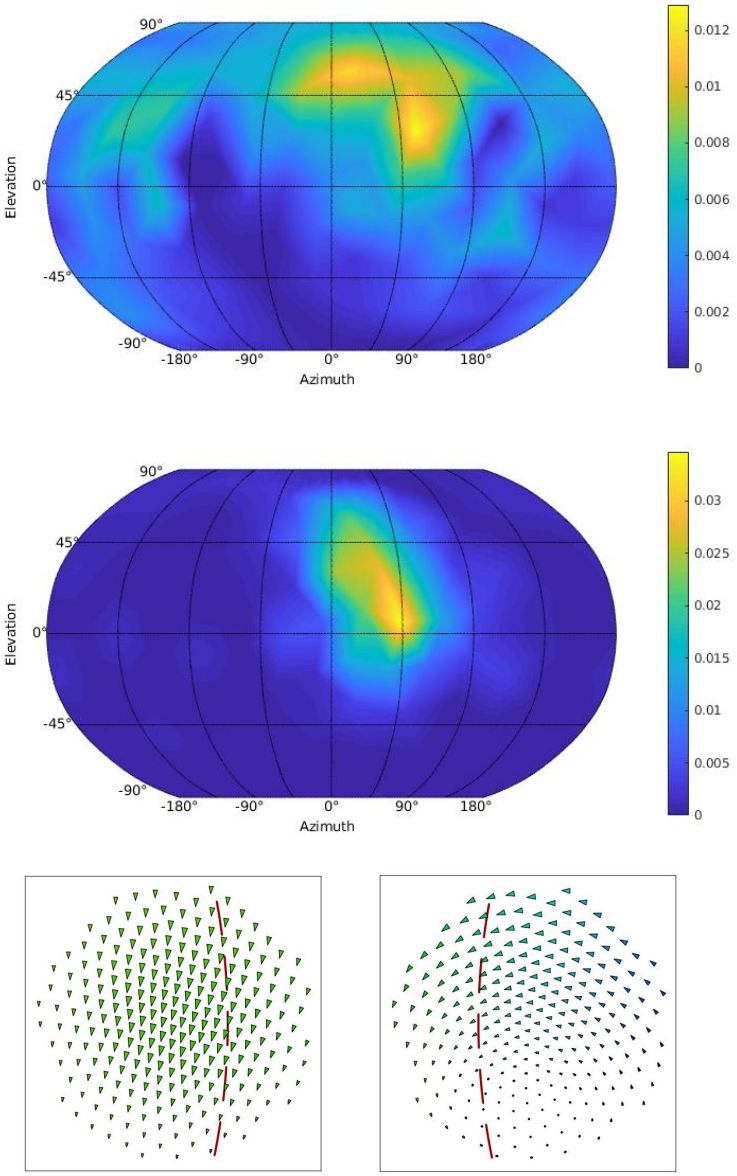


Figure 2.8.: Typical selectivity heatmap of a model neuron for translation (up) and rotation plus flowfield for left and right eye for comparison

From the results of overcompleteness 4.0. The colour represents the interpolated relative frequency of activation for the inputs assigned to each bin. Being ranked eighth in average activity, this neuron is selective for rotations around axes pointing forward and shows weaker translations going forward, slightly to the right and upwards. The directions of high selectivity broadly match the patterns in the flowfields: the right eye field is a cyclic pattern as from a rotation forward and to the right, which would also match the downward pattern for the left eye. This also fits to downwards translations, while the stronger part of the right eye pattern is directed to the left, matching with translations to the right.

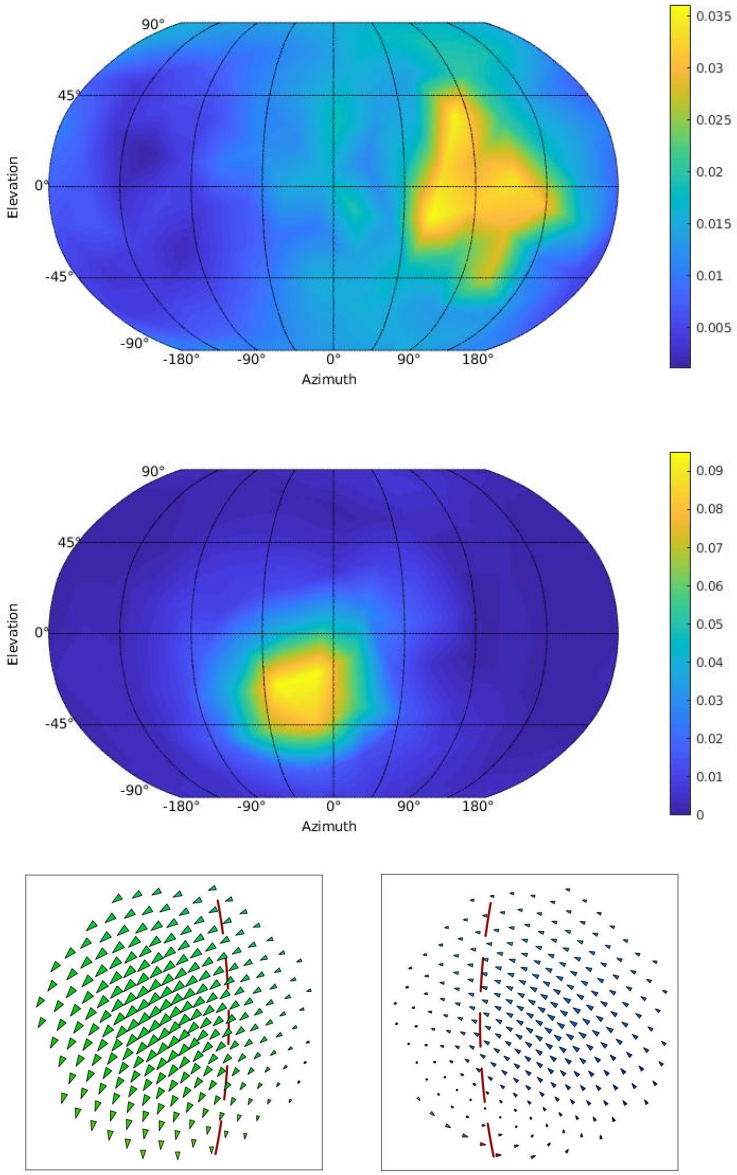


Figure 2.9.: Typical selectivity heatmap of a model neuron from the network with overcomleteness 0.5 plus flowfield for left and right eye for comparison

The selectivity is illustrated in the same way as in 2.8. This neuron (rank 6 in overall activity) is selective for rotations around axes pointing forward, downwards ans slightly to the right. There is also quite weak selectivity for translations to the left. The motion vectors for the right eye go down and to the left, those for the right one up and to the left. This matches a rotation axis pointing forward and downwards. However, translations to the left, as indicated in the tuning map, do do not fit this pattern.

translation, the result was .0100, for rotation .0176, while the overall mean for all directions and kernels was .0028 for both categories.
For overcompleteness 0.5, similar data were obtained. One difference was found in the generally larger regions of high activation frequencies, as visible in the example in figure 2.8. Looking at the numbers, the general average frequency is higher (.0116 for both translation and rotation), while the peaks differ somewhat less from the rest: The average of the maximum relative frequency per kernel is .0292 for translation and 0.0557 for rotation.

2.5.2. Maximum likelihood estimates

The likelihoods for a kernel to be active given an axis of translation, respectively rotation, that we had derived from relative frequencies, formed the basis for the estimation of the direction from an activity pattern of the sparse layer. We used maximum likelihood estimates based on the naive Bayes rule, which is according to Duda et al. (2001):

$$p(\omega_k|x) \propto \prod_{i=1}^{d} p(x_i|\omega_k) \tag{2.12}$$

This is derived from Bayes' theorem by taking the 'naive' assumptions that there are no statistical dependencies between different x_i and that there are no unequal prior probabilities $p(\omega_k)$. Despite its simplifying assumptions, that may not hold for many applications, naive Bayes classifiers have been proven to be a suitable instrument for classification tasks.
Applying the naive Bayes rule to the likelihood functions of the kernels means that the activities of different kernels are assumed as independent, which seemed reasonable as in a sparse coding the network elements are chosen to have minimal covariance. Due to the implementation of the choice of motion direction, there technically are different prior probabilities for different directions, which are ignored here to simplify the model.
Under those assumptions, for obtaining maximum likelihood estimates, it was enough to determine the direction ω_k giving the maximal likelihood $p(\omega_k|x)$

for an activity pattern x by comparing the proportional value $\prod_{i=1}^{d} p(x_i|\omega_k)$, that was easily obtained from the likelihood functions of the kernel. More precise, we calculated the estimator

$$\begin{aligned} \hat{\omega} &= argmax_{\omega_k} \hat{p}(\omega_k|x) \\ &= argmax_{\omega_k} [\prod_{X_i active} p(x_i|\omega_k) \cdot \prod_{X_i inactive} (1 - p(x_i|\omega_k))] \end{aligned} \tag{2.13}$$

taking in account both the likelihoods of active and inactive kernels. This was done separately for the translational and rotational component of a given activity pattern x, independently obtaining estimates of translation and rotation axis.

2.6. Estimate evaluation

In order to analyze the estimation model, we needed a large test set to observe the activity of the kernels for all directions and compute the likelihoods needed for the naive Bayesian classifier, but also a large validation set for applying the maximum likelihood method and thus examining the precision and accuracy of the resulting estimates. Since the creation of the dataset of flow fields used by Ecke et al. (2018) took a waste computational effort, we decided to apply the common method of leave-one-out cross-validation to efficiently exploit the existing data. Under this approach, the likelihoods were recalculated for the estimation of each motion event from the relative frequencies in the remaining set of all events except for the one being estimated.

The estimates we obtained after those computational steps were the directions of the translation axes, respectively rotation axes of the fish's egomotion. As these axes could point in any direction in the three-dimensional space, while their length was neglected, we represented them as unit vectors or equivalently as points on the unit sphere. Thus, for the evaluation of the estimates, we made use of spherical statistics.

To examine the precision of the estimates, we computed the mean vector of

the estimate vectors $\hat{\omega}$ relative to the respective axis of actual rotation or translation ω. For this, we first rotated each estimate $\hat{\omega}$ around the cross product $\omega \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ by the angle between ω and the x-axis. That is the sphere rotation aligning ω with the x axis with the minimal angle. This step enables us to evaluate all estimates across the different original axes of translation and rotation, by comparing their direction with respect to the x-axis as an equivalent of the direction of the estimates $\hat{\omega}$ with respect to the particular original direction ω.
In the next step, we took the sample mean vector $\bar{x}$ of all rotated estimate vectors $x_i = \hat{\omega_i}^*$ for the whole dataset:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \overline{R} \cdot \bar{x}_0 \tag{2.14}$$

The polar representation of $\bar{x}$ consists of the mean direction of the sample $\bar{x}_0$ and the mean resultant length $\overline{R} \in [0,1]$, a measure of precision, where zero indicates a uniform distribution spanning the sphere, whereas one indicates that all vectors are equal. According to Mardia & Jupp (2000), the total variation $1 - \overline{R}^2$ is an appropriate analogue of the sample variance. Due to the rotations performed on the estimate vectors, the mean direction here is no appropriate measure of accuracy. Therefore, we will rather discuss the biases separately for the estimation of motions out of each bin.
As additional measures we determined the mean of the angles between the estimates and the actual directions to demonstrate the average discrepancy. It was computed by

$$\bar{\alpha} = \frac{\sum_{i=1}^{n} \alpha_i}{n} = \frac{\sum_{i=1}^{n} atan2(||\omega_i \times \hat{\omega}_i||, \omega_i \cdot \hat{\omega}_i)}{n}, \tag{2.15}$$

where n denotes the number of estimates. Further we calculated the amount of correct classifications with respect to the bins, *hits*. Table 2.1 shows the mentioned values for different configurations of the model. Additionally, the distributions of the angles α_i are supplied in A.2.
For the contextualization of the obtained measures, we should not neglect,

	Translation				Rotation			
Overcompl.	*hits*	$\bar{\alpha}$	$\overline{R}$	$1-\overline{R}^2$	*hits*	$\bar{\alpha}$	$\overline{R}$	$1-\overline{R}^2$
0.5	.057	66.0°	.290	.916	.219	30.5°	.804	.354
4.0	.047	69.7°	.340	.884	.221	30.4°	.804	.354

Table 2.1.: Measures for the precision of the estimates depending on the models overcompleteness. For translation and rotation estimates, the table includes the amount of correct classifications, the mean angle of discrepancy, the mean resultant length and the total variation relative to the axes of the original motions.

that the model could impossibly deliver perfect estimations, as it could only choose one of the 75 bins and not the exact direction of the motion. Under this constraint, an optimal performance would be achieved, if the model always chose the centre point of that bin, to which that motion is assigned when it is part of the test set. This would lead to $\overline{R}_{opt} = 0.986$, for both rotation and translation.

All the measures of estimate precision provided in 2.1 show a notable difference between translation and rotation: they indicate a higher precision of the rotation estimates compared to the translation estimates. As a result, the total variance is higher for rotation. For translation, the mean resultant length is closer to the value for the uniform distribution than to the value for translation. Examples for the distributions of the estimates are illustrated in figure 2.10.

In the measures from table 2.1, the difference between the runs with overcompleteness ratio 0.5 and 4.0 is small for translation estimates and almost non-existent for rotation. Since the conditions do also not show notable differences in the further examinations, we only describe the results for overcompleteness 4.0, as the precision is slightly higher for translation estimates. Therefore, figures 2.10 to 2.14 display values for that condition.

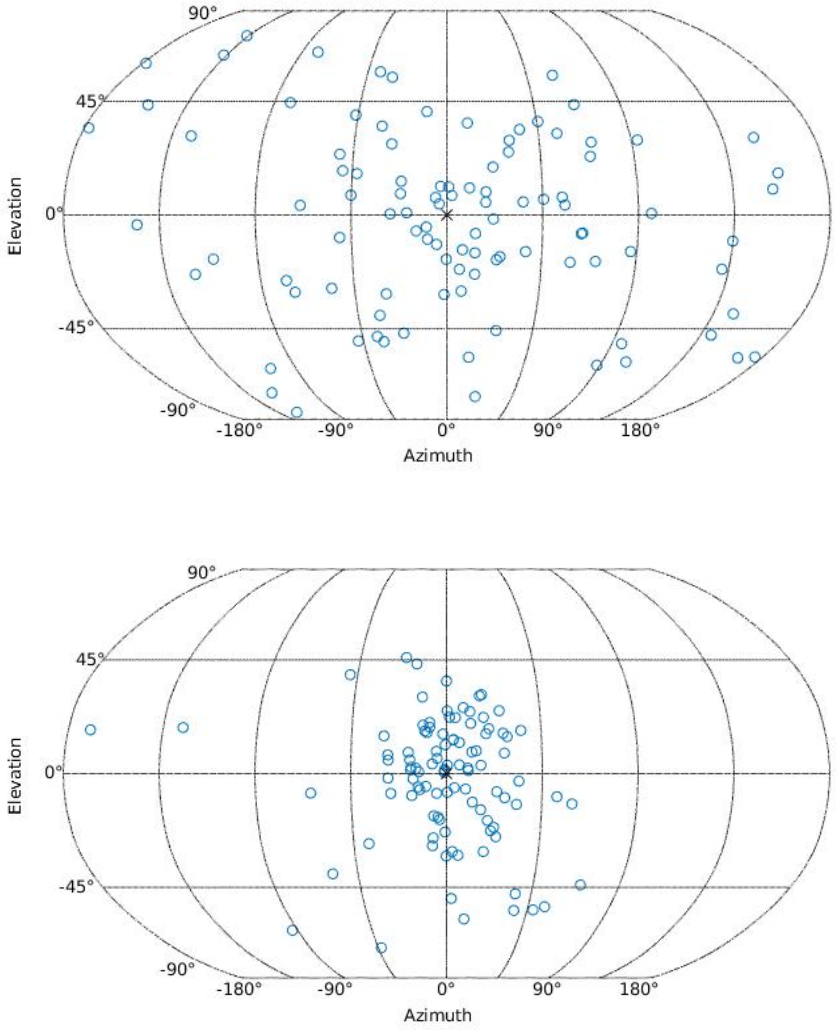


Figure 2.10.: Example for displacement of translation (top) and rotation estimates

Blue circles represent the relative direction 100 randomly chosen estimates, after being rotated so as to align the input motion axes with the origin. They are projected to 2d like the points in figure 2.6.

One correlation, that we investigated, was a possible link between the precision of the estimates and the intensity of translation and rotation. The relation between $\bar{R}$ and both translation distance and rotation angle is illustrated in figure 2.11. For rotation, a clear pattern can be seen: The larger the rotation angle and the smaller the translation distance, the higher the precision of estimates. For translation it is the opposite, although the pattern is more noisy.

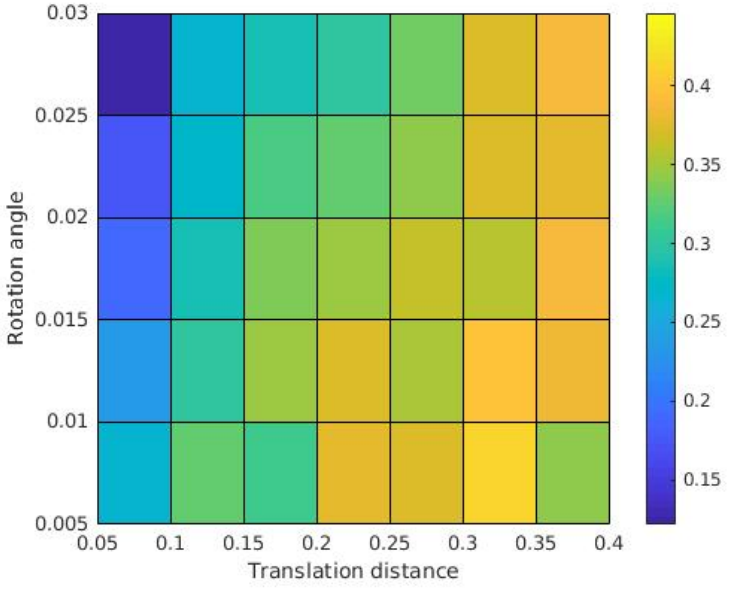
Further we examined, whether there was a connection between the precision and the direction of the motion. Thus we grouped the motions again by the bin they were assigned to when the test set was formed, and computed the *mean resultant length* separately for each bin. The result is illustrated by the heatmaps in figure 2.12. It shows, that the differences between the direction are higher for translation than for rotation, and that they occur more systematically: Estimates for forward and backward translations are very noisy, while they are more precise for the other directions, but still not as precise as the rotation estimates. For the latter, the differences are not as striking and resemble the distribution of rotation axes shown in figure 2.7.

While the previous results described the precision of our estimates, we also tested our model for systematic biases. For this purpose, the estimates, again, were grouped with respect to the bin they were assigned. Then the angle between the *mean direction* $\bar{x_0}$ of that sample and the center of the bin was calculated. This resulted in the heatmap depicted in figure 2.13, indicating the absolute size of the systematic bias. To also illustrate the direction of the bias, arrows from the bin center towards the *mean direction* were added.

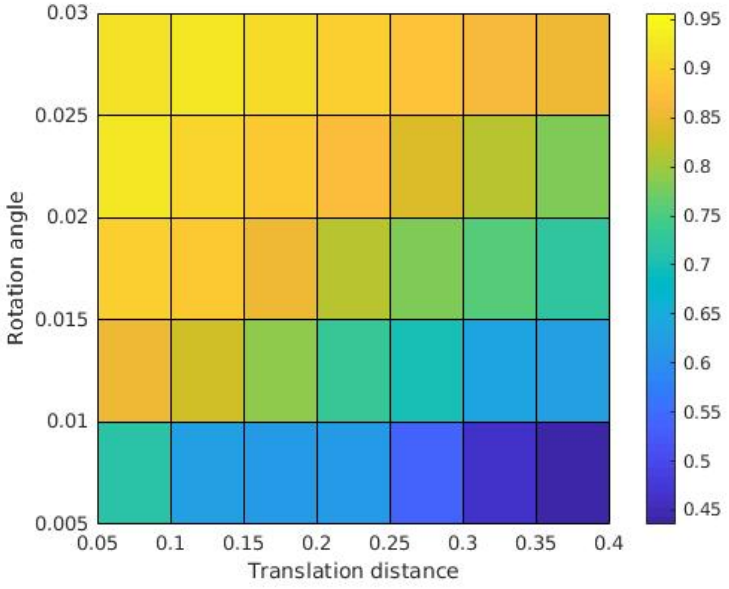
It turns out that, in general, the bias for translation is vastly higher, and also more systematic, than for rotation: estimates of translations forward and somewhat to the right are biased downwards, just like those backwards and somewhat to the left. For motions forwards and slightly to the left or backwards and slightly to the right, it is the opposite effect.

This leads to an overall higher amount of estimates pointing to the right and up, to the left and up, to the right down and to the left and down, whereas less estimates point directly forward or backward. The distribution of all estimates is depicted in figure 2.14.

While the density pattern for translation estimates can be related to the bias, the rotation pattern broadly resembles the distribution of input directions as seen in figure 2.7. However, there is one bin, directed forward and downwards, that contains a vastly higher amount of estimates.



(a) Translation estimates



(b) Rotation estimates

Figure 2.11.: Mean resultant length depending on both translation distance (in m) and rotation angle (given in rad)

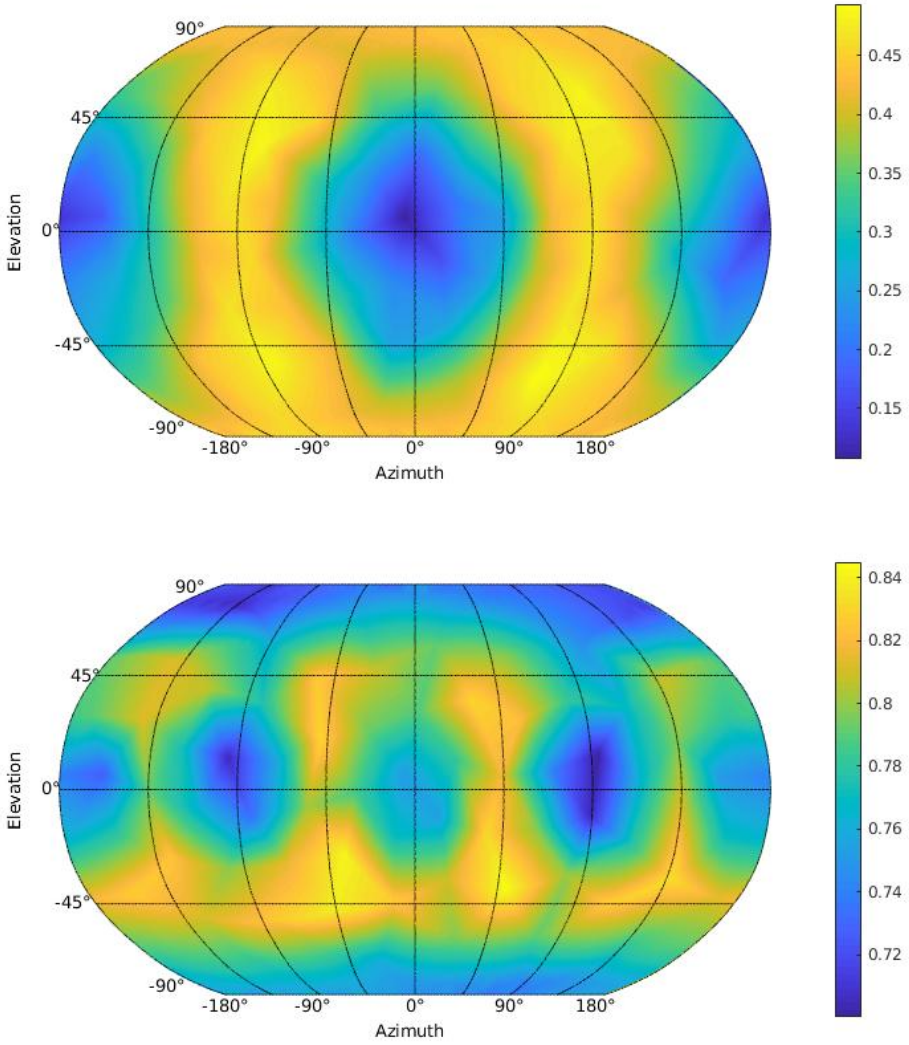


Figure 2.12.: Mean resultant length per bin of translation (top) and rotation

Colour represents the mean resultant length $\bar{R}$ for the motions assigned to each bin. The data are interpolated and projected as described in 2.6.

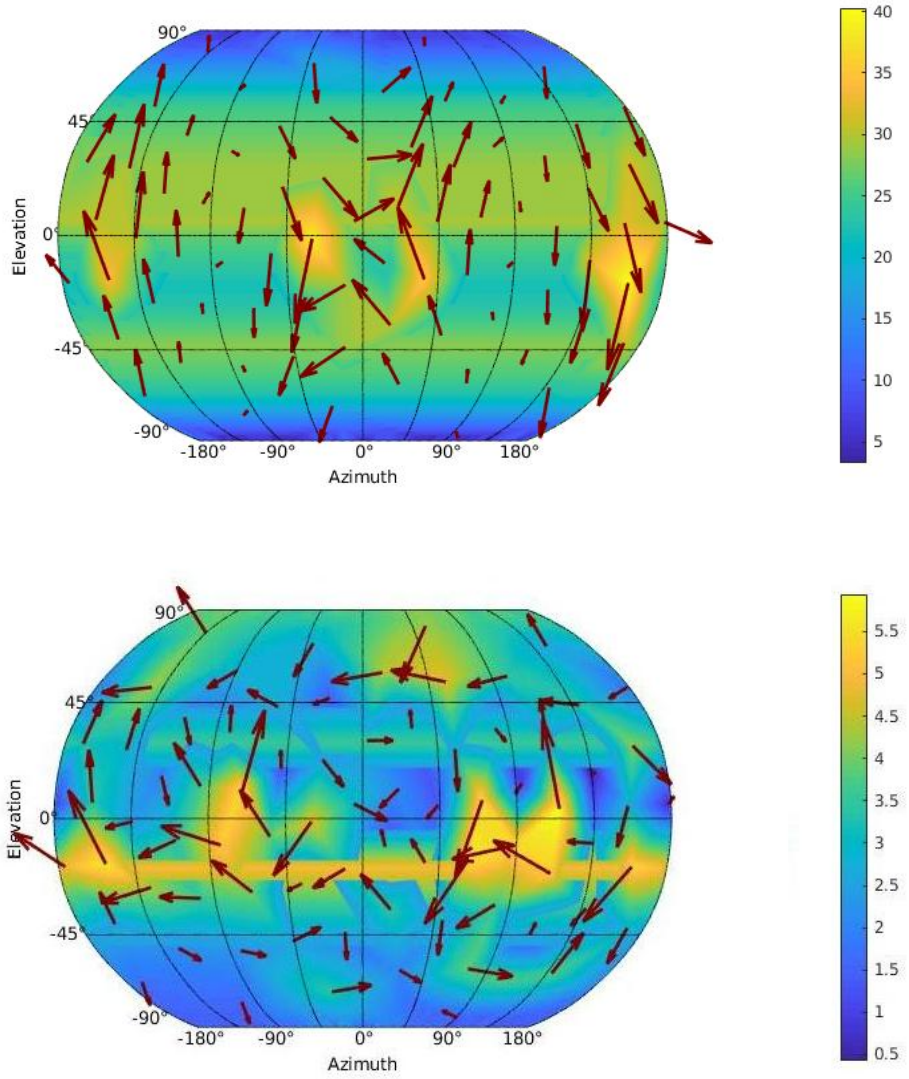


Figure 2.13.: Bias angle and direction per bin for translation (up) and rotation

The color represents the angle (in degree) between the mean direction and the bin center. The arrows start at the center and are directed towards the mean direction. However, the arrowhead is not located at the mean direction, since the arrows were resized in proportion to the size of the bias, in order to make them more discernible and comparable across the map.

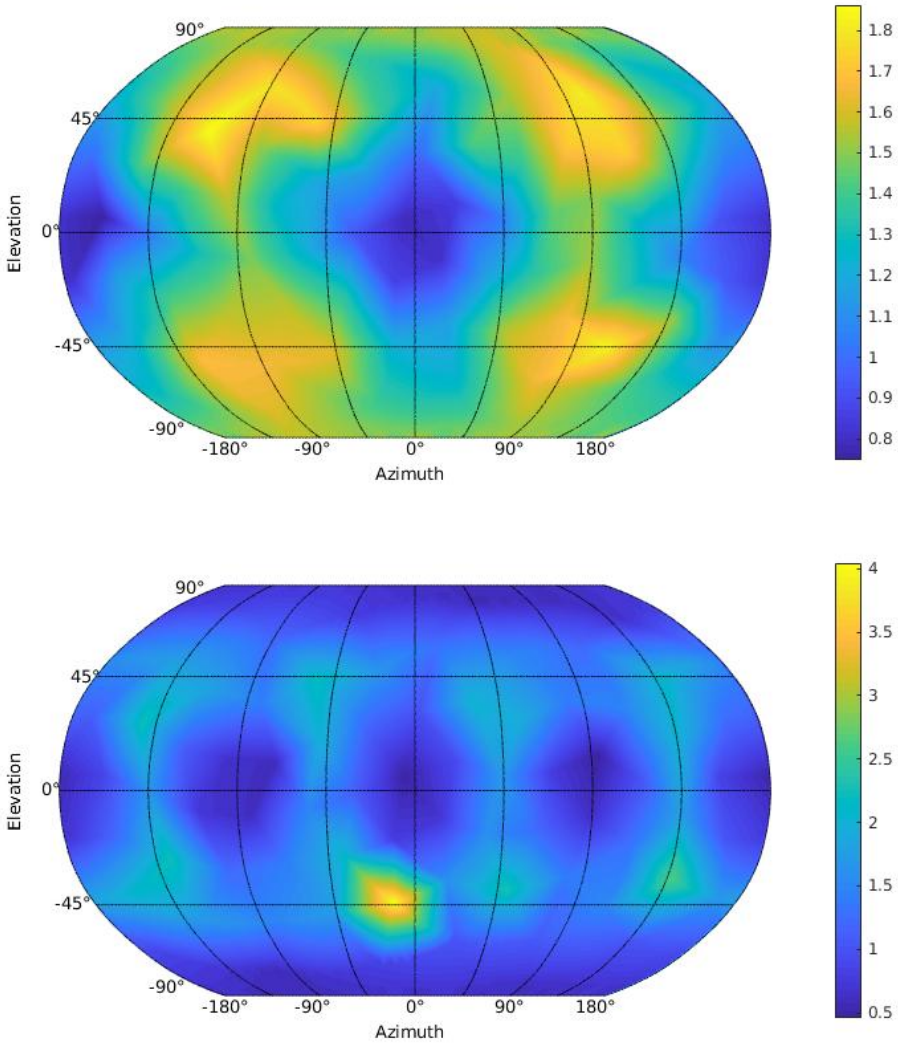


Figure 2.14.: Density of estimates for translation (up) and rotation

The colour represents the amount (in percent) of estimates assigned to each bin.

3. Discussion

3.1. Summary of the procedure

Perceiving the position and motion of their own body is a crucial skill for animals, as it enables them to navigate their surroundings and to react properly to positional changes. For example, zebrafish adjust the movements of their eyes and their body when they detect that they are moved. Kubo et al. (2014) found neurons selective for specific translations and rotations in the pretectum of the fish. Ecke et al. (2018) could reproduce response properties of these neurons by building a sparse coding model of these neurons and training it in an unsupervised fashion, with input from a virtual reality simulation of a realistic environment. Their findings suggest that sparse coding models can be a proper tool to simulate the ego-motion detection in zebrafish.
As the model did not provide estimates of the ego-motion direction, it remained open, whether the sparse coding representation is suitable for this specific task. Therefore, we decided to extend the model by a maximum likelihood estimation based on activity patterns of the sparse layer. The likelihoods were computed by evaluating the response characteristics of the individual neurons and applying the naive Bayesian classifier. Afterwards, we examined the accuracy and precision of the outcomes as well as possible relations to the amount of the translation and rotation component and to the direction of motion. In the following sections, we will provide interpretations of the obtained results, take them as a basis for discussing the approach and, finally, illustrate possible impovements of our model.

3.2. Interpretation of the results

In reflection of the results of the analysis, the model delivers fairly reliable estimates of the rotation direction, and hardly reliable estimates of the translation direction. This indicates, that the sparse coding brings out hidden statistical structure of the input, i. e. components of the optic flow caused by specific translations and rotations. Yet, especially for translations, these properties are not clear enough. The tuning maps of almost all the model neurons show selectivity for certain motion directions, but it appears as if the patterns have to be more clear to provide a basis for reliable estimates. Possible reasons and implications of this performance will be discussed in the following sections, after we provided interpretations for our further findings.
A salient aspect of the results is, that there is much less variance in the rotation estimates than for translation. This matches with the observation of more concentrated regions of high activity in the selectivity heatmaps of the neurons for rotation. A possible explanation for this difference can be found in the properties of flowfields resulting from combined rotation and translation, like all flowfields of our dataset: the rotational component tends to overshadow the translational component, as rotations cause large flow vectors in the whole field of view, while for translations, it depends on the surrounding, since distant objects cause amost no flow vectors. Thus the rotation fields are constant for constant axis and angle, while the translation fields also depend on the position of other objects, making it more difficult for the neurons to adapt to patterns caused by specific translation axes and thus causing more noise in the estimation.
The overall precision and accuracy of the estimates depended scarcely on the number of neurons in the sparse layer. This also applied to the difference between translation and rotation and the dependence on direction or intensity of translation and rotation. Apparently, the redundancy of representation, that was added by the increased population, was not exploited in a way, that would lead to a more suited representation of the data.

Looking closer at the estimates, we found an effect of the ratio of translation and rotation: The higher the amount of the rotation compared to the translation, the more precise were the rotation estimates, and the more scattered the translation estimates. This is not surprising, as the influence of the rotational pattern on the representation is higher and, thus, it can be recognized easier.
Looking at the relation between motion direction and estimate precision, we found a clear pattern: more variance in estimates of forward and backward translations, as well as more precision for the remaining directions. This is a rather odd result, as one might expect forward and backward motion to be recognized better, since they cause large flow vectors on both the left and the right side of the field of view, presumably facilitating their discrimination. By contrast, for movements e. g. to the left, only the large flow vectors in the middle of the field of view can be seen, while the other large vectors are behind the head of the fish.
The root of this unexpected effect seems to be, that estimates of forward and backward translations are biased up or down, leading to certain regions of high density of estimates. The exact reason for this phenomenon is uncertain, as well as the cause for the bias towards one specific bin for the rotation estimates.

3.3. Pros and cons of the approach

On the basis of these findings and their explanations, the success of the present approach can be evaluated with respect to its goal: with the present model, we aimed to provide an explanation for the neural processing behind the ability of zebrafish to discern specific translations and rotations. The mentioned results give an ambivalent answer to the question, whether the approach is suited for the task of estimating ego-motion direction.
One point, that clearly speaks against the approach, are the unsatisfactory outcomes: especially the translation axis estimates produced by our model contained a systematic bias and too much variation to allow reliable reactions

to positional changes. This implies, that under our approach either the model neurons do not develop selectivities clear enough for generating precise translation estimates, or that these selectivities have to be statistically evaluated by more suitable methods. The model definitely has to be improved to give outcomes suited for the challenges of the environment of the fish. If coming efforts to address this problem will not succeed, a different approach may be needed to understand the coding used by the fish to detect ego-motion and to react properly.

Another point, that argues against sparse coding as a proper description of the representations in the zebrafish pretectum is the difference in the number of neurons: While we employed respectively 1024 and 4096 kernels in the sparse coding layer, Kubo et al. (2014) only found approximately 600 cells per fish with selective activity for ego-motion. Thus, at least the more typical, highly overcomplete network has to be considered unrealistic. The contrast in neuron numbers may be due to filtering of relevant information in earlier visual processing steps in the zebrafish brain, while in our model a complete representation of the optic flow input is established by the sparse coding layer. The non-overcomplete network produced estimates that were not much worse, but to examine the performance of a sparse network with a more realistic number of kernels can be a starting point for coming studies. However, with the larger amount of neurons, our model is unlikely to exactly recreate response characteristics of the zebrafish pretectal neurons.

But there are also good reasons to believe, that the sparse coding approach is suited for simulating the ego-motion detection in zebrafish. Our model produced quite reliable estimates for the axis of rotation, being even close to the optimum for large rotation angles and small translations. This shows, that the sparse coding network extracts statistical properties of the input data in a way that is suited for solving specific tasks. This positive part of the results is in a row with various successful approaches of modelling properties of biological neurons under the use of sparsity, like the outcomes of the original algorithm presented by Olshausen & Field (1996).

There is another aspect of the sparse representations that suggests that they are a proper simulation of the neural processes of the fish: The properties

of the receptive fields seem realistic for neurons concerned with ego-motion detection: They partly resemble the fields Krapp & Hengstenberg (1996) found for interneurons in the bowfly, which also are global and which are believed to extract rotatory and translatory flow components. This similarity suggests, that the mechanisms underlying the creation of receptive fields in animals are related to the principles our model is based on. This conjecture is supported by several findings of sparse coding in the optical tracts of the brains of different animals (Baddeley et al., 2001; Froudarakis et al., 2014). Furthermore, the reason for the occurence of sparse representations in animal brains appears not necessarily to be an advantage in classification performance. They appear to have more benefits, as they are suited for deriving probabilities of events (Gardner-Medwin & Barlow, 2001), therefore support knowledge about statistical properties of the environment (Barlow, 2001), and also are efficient in terms of energy (Attwell & Laughlin, 2001). These advantages suggest, that sparse coding could be the best form of representation in the animal brain even for tasks, where it does not lead to optimal performance.

3.4. Flaws and starting points

The positive aspects in the results and the general suitability of sparsity for neural representations encourage us to think about changes to our model, that could lead to more useful estimates of the motion direction. That is why in this section, we will reconsider so some questionable aspects of the model and suggest possible improvements for future approaches.
One of these aspects is, that for training the network, the input is always resulting from a combination of rotation and translation. A downside of these joint motion inputs, that became apparent in the effect of the translation-rotation ratio on estimate precision, is the overshadowing of translation components by the rotation fields. This could be fixed by the introduction of such a threshold, that leads to filtering of those input data for the computation of the rotational likelihood functions, for which the rotational component pre-

dominates, and analogously for translation. This constraint could enhance the accuracy of the likelihood functions and, thus, lead to more reliable estimates. As an alternative solution, one might consider the presentation of purely translative motion fields to facilitate the development of translation selectivities.

Yet in the case of training joint motion fields, the likelihoods, rotation and translation are evaluated separately, and later also estimated separately. It is conceivable, that a model trained only on combined motions might also be better suited to detect such a combination. Implementing this in the approach as it is presented here, however, would require separate likelihoods for each combination of rotation and translation bin and, therefore, need a larger dataset.

Flowfield data for a greater set of motions would anyway be useful as it would enable the employment of separate records for network training and estimation. The cross-validation guaranteed, that an activity pattern is not used to generate the estimate for the same movement, but all of this data was also used before for training the sparse coding layer, which prevents an optimal independence of the different steps. As generating the input data from the virtual reality simulation requires vast efforts, we decided to work with the existing dataset, created by Ecke et al. (2018), for the present project. However, it would be sensible to create a new set in order to improve the validity of future research.

Calculating a new set of flowfields would also bring the opportunity to remove one flaw in the existing dataset, that is the non-uniform distribution of the movement axes across the unit sphere that is displayed in figure 2.7. By applying an appropriate algorithm such as the ones described by Marsaglia (1972), random samples of a uniform distribution across the unit sphere are easily obtained.

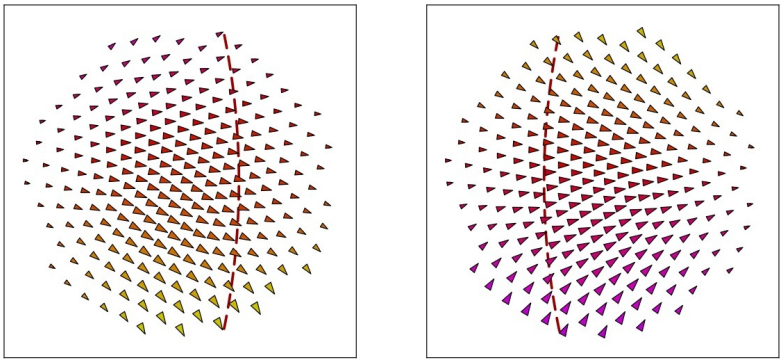
But instead of using a uniform distribution of ego-motion directions, coming approaches could also consider modelling a distribution that fits better the actual stimuli in the environment of zebrafish. It is plausible to assume, for example, that fish turn their bodies more often around vertical axes than around horizontal ones. As sparse coding highly adapts to the properties of

the input, implementation of such constraints would very likely lead to more realistic response characteristics of the kernels.
Another parameter, that was restricted by the limited dataset, was the number of bins. The binning itself brings difficulties for the statistical evaluation of the estimates, but is necessary for the maximum likelihood method we applied here. With smaller bins, we would expect the estimates to be more precise, as the selectivities of the kernels could be captured in greater detail and also the estimation model could choose between more possible directions. We did not run the model with more than 75 bins due to the tradeoff against the number of data for each bin, which is also important for the reliability of the calculated likelihood functions: In our study, at least 537 activity patterns were assigned to one bin. Due to the low activation ratio of the sparse layer, it would be hard to infer informative probabilities from distinctly less data. If future projects work with larger databases, this would be a simple way to enhance the precision of the estimates.
One of the simplifications we used in our model was that, for the computation of the likelihood functions, we only considered, whether a neuron was active or not for a particular motion event, but not the magnitude of its activity. Yet, there might be more information in the differences in magnitude, which could be exploited by a proper statistical model. One approach would be to discriminate different events for a particular neuron, depending on the magnitude of activation, and computing different likelihood functions for these. Implementing the necessary modifications to our model was out of the scope of the present work, but might be another promising possibility for enhancing the results of this approach.

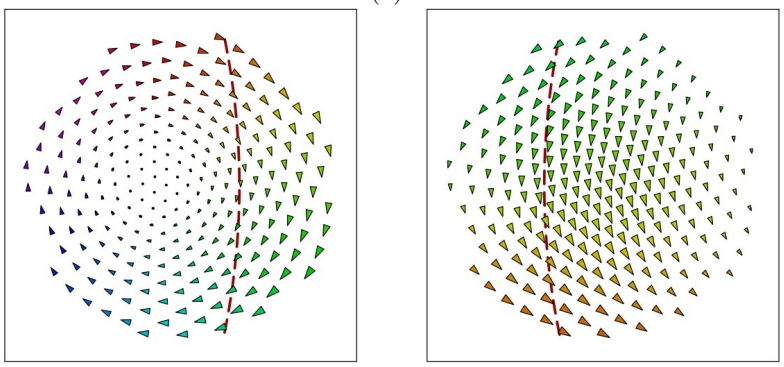
3.5. Conclusion

Considering the analysis of the estimates of ego-motion direction provided by our sparse coding approach, the current model is not capable of reliably detecting translation and rotation axis at the same time. Still, we conclude that the presented approach constitutes a promising attempt to simulate the neural processes underlying ego-motion detection in zebrafish. It did not only, as shown by Ecke et al. (2018), reproduce the response properties of pretectal neurons, found by Kubo et al. (2014), but also led to the emergence of model neurons showing selectivities for particular motion directions. Moreover, the rotation estimates inferred from the resulting representation were fairly accurate. These positive outcomes might encourage further research on the use of sparse coding in this context. There are several aspects of the present model that might be improved in future studies with the aim to show, whether sparse coding can provide the basis for ego-motion detection in zebrafish.

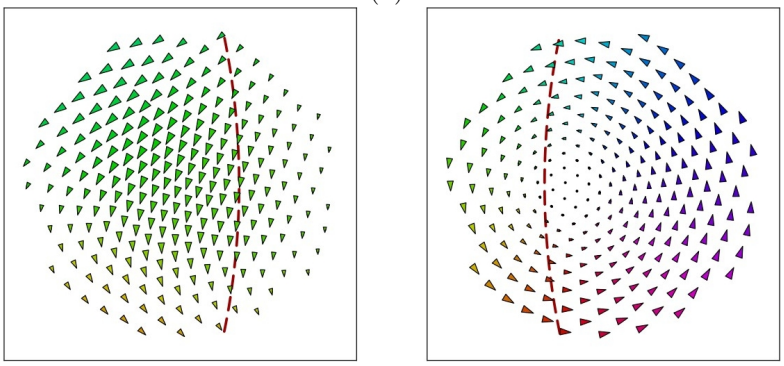
A. Supplementary material



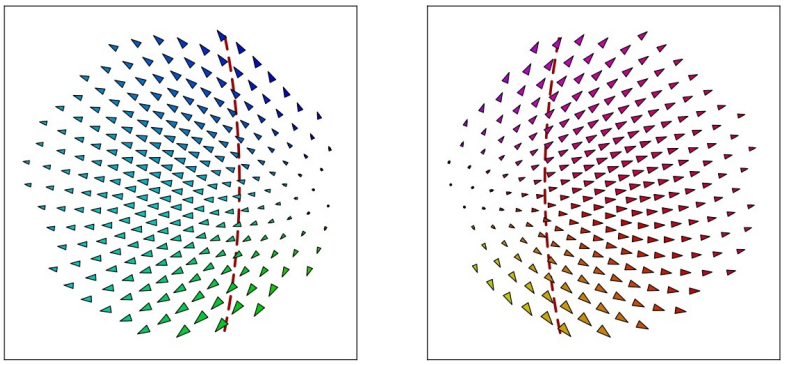
(a)



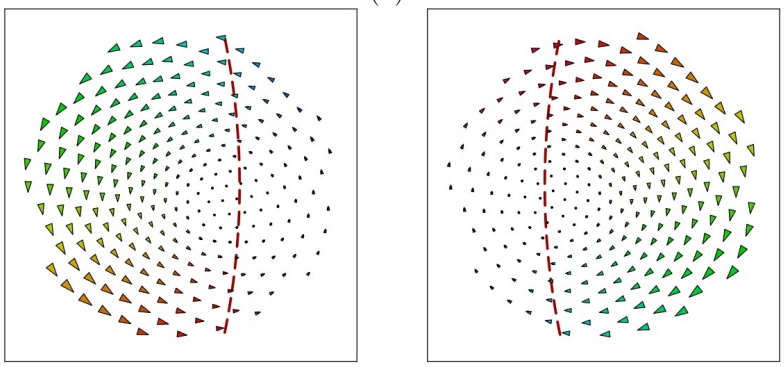
(b)



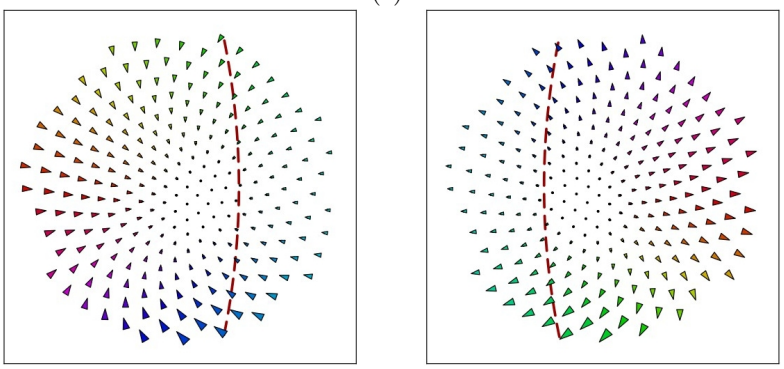
(c)



(d)



(e)



(f)

Figure A.1.: Flowfields of the six components explaining the highest amount of variance, ordered by that amount

The coefficients, that define the contribution of each retinal ganglion cell to the component, are illustrated by a vector in the respective location, with orientation and colour representing the locally preferred direction, and the size showing the amount of contribution. The fields are drawn for the left and the right eye, with dotched red lines indicating the overlap of the fields of view.

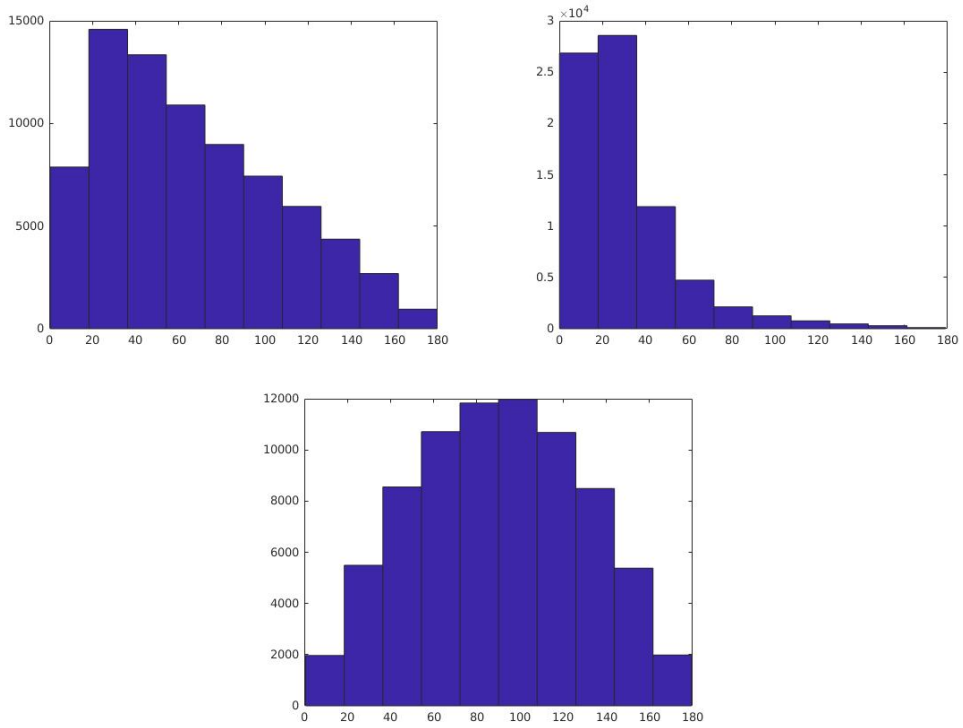


Figure A.2.: Distributions of the angles between estimates and original motions

The distributions for the translation (upper left) and rotation estimates (upper right) and, to compare, for a random distribution (down), angles in degrees.

Bibliography

Attwell, D., & Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow and Metabolism*, *21*, 1133–1145.

Baddeley, R., Abbot, L. F., Booth, M. C. A., Sengpiel, F., Freeman, T., Wakeman, E. A., & Rolls, E. T. (2001). Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society of London B*, *264*, 1775–1783.

Barlow, H. (2001). Redundancy reduction revisited. *Network: Computation in Neural Systems*, *12*, 241–253.

Burge, J., & Geisler, W. S. (2014). Optimal disparity estimation in natural stereo images. *Journal of Vision*, *14*(2), 1–18.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). John Wiley and Sons.

Ecke, G. A., Mikulasch, F. A., Bruijns, S. A., Witschel, T., Arrenbreg, A. B., & Mallot, H. A. (2018). Sparse coding predicts optic flow specifities of zebrafish pretectal neurons. *ArXiv e-prints*. Retrieved 2018-09-09, from `https://arxiv.org/abs/1805.01277`

Froudarakis, E., Berens, P., Ecker, A. S., Cotton, R. J., Sinz, F. H., Yatsenko, D., ... Tolias, A. S. (2014). Population code in mouse v1 facilitates readout of natural scenes through increased sparseness. *Nature Neuroscience*, *17*, 851–857.

Gardner-Medwin, A. R., & Barlow, H. B. (2001). The limits of counting accuracy in distributed neural representations. *Neural Computation*, *13*, 477–504.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, *13*, 411–430.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, HI, USA*, 2462–2470.

Krapp, H. G., & Hengstenberg, R. (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, *384*, 463–466.

Kubo, F., Hablitzel, B., Dal Maschio, M., Driever, W., Baier, H., & Arrenberg, A. B. (2014). Functional architecture of an optic flow-responsive area that dirves horizontal eye movements in zebrafish. *Neuron*, *81*, 1344–1359.

Mardia, K. V., & Jupp, P. E. (2000). *Directional Statistics* (1st ed.). John Wiley and Sons.

Marsaglia, G. (1972). Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, *43*(2), 645–646.

Nakayama, K. (1985). Biological image motion processing: a review. *Vision Research*, *25*(5), 625–660.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.

Raudies, F., & Neumann, H. (2012). A review and evaluation of methods estimating ego-motion. *Computer Vision and Image Understanding*, *116*, 606–633.

Rigamonti, R., Brown, M. A., & Lepetit, V. (2011). Are sparse representations really relevant for image classification? *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011, Colorado Springs, CO, USA*, 1545–1552.

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

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 e-prints*. Retrieved 2018-09-20, from `https://arxiv.org/abs/1406.4205`

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, HI, USA*, 1851–1860.