Mathematisch-Naturwissenschaftliche Fakultät
Lehrstuhl für Kognitive Neurowissenschaft

Bachelor Thesis

Self-organization of motion-sensitive receptive fields in the zebrafish optokinetic system

Author: Fabian Mikulasch
fmikulasch@googlemail.com

Supervisors: Gerrit Ecke
Prof. Dr. Hanspeter Mallot
Prof. Dr. Aristides Arrenberg

Date of submission: 09.03.2017

I. Abstract

Many animals rely on motion cues in their field of view to control their behavior. Zebrafish for example show an optomotor and optokinetik reflex when presented with certain optic flow stimuli. We reasoned that binocular whole field optic flow, originating from egomotion in a natural environment, can be described by a small number of statistically independent components. These sparse components could be involved when neurons in the pretectum of the zebrafish, which are supposed to be responsible for the horizontal optokinetik reflex, specify their receptive fields.

To test the hypothesis we created plausibly natural optic flow stimuli from the viewpoint of a zebrafish in a virtual reality simulation and computed the sparse components of the data with the local competitive algorithm. We then compared the resulting "receptive fields" and their response characteristics to responses to horizontal whole field optic flow of neurons in real zebrafish, measured by Kubo et al. (2014). Though the model still has room for improvements, we could indeed find sparse components, as we expected, which seem to be specific for certain events in the surrounding of the fish. The majority of these components appear to extract optic flow cues originating from either rotation or translation of the fish, or from moving objects in the world. These findings qualitatively accord to some of the measurements by Kubo et al. (2014) and invite to further research in this direction.

II Table of contents

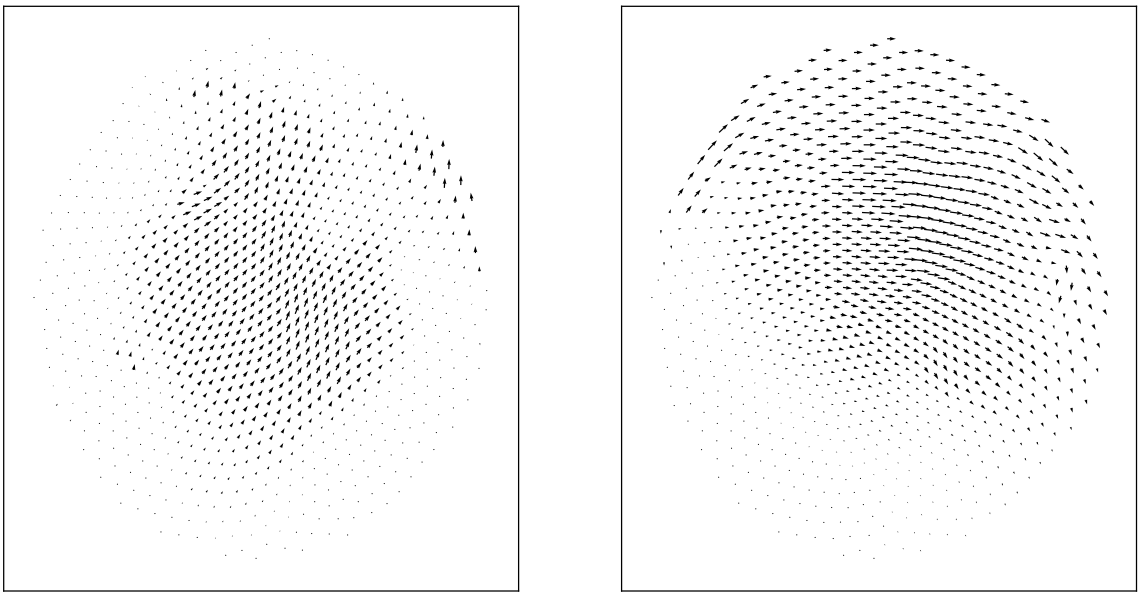

Figure 1: Simulated binocular whole field optic flow from the viewpoint of a zebrafish. Arrows point in the direction of motions and their length denotes the speed.

1. Introduction

Many animals rely on motion cues in their field of view as indicators for important events in the world. A lot of them also show specialized responses when presented with certain optic flow signals (Figure 1). A very common observation in different animals is the use of such signals, originating from egomotion in a fixed environment, to orient and compensate for displacements in their location. Water striders for instance are able to recover their position after a period of combined translation and rotation with a single jumping motion. It could be shown that this optomotor reflex, and therefore the discrimination between translation and rotation, continues to be successful even if the stimuli are restricted to a monocular field of view (Junger and Dahmen, 1991).

Newer studies are still concerned with the question what these optic flow cues, used in animal behavior control, look like, especially for rotation and translation discrimination. The perception of optic flow is implemented by direction sensitive cells which show dedicated responses to movements in a certain direction. Such cells can be found at many levels of the visual system. Krapp et al. (1996) for example investigated local motion tuning of cells in the third visual neuropile of the blowfly with intracellular recording. Their study suggests that the fly develops cells that specifically extract very large optic

flow components of either rotation around a particular axis or translation.

With recent methods of calcium imaging the processing of these informations in a large part of the neural system can be analyzed on a neural level and with a high temporal resolution (Grienberger and Konnerth, 2012). Especially in translucent zebrafish embryos it is possible to optically observe and manipulate the firing of genetically altered neurons throughout the brain (Baier and Scott, 2009). In addition, zebrafish show an optomotor reflex, by stabilizing their position in respect to a moving background (Neuhauss et al., 1999), and an optokinetik reflex, i.e. the eyes of the fish move to keep the field of view steady (Brockerhoff et al., 1995), making them a convenient research subject for this matter. A range of lesion and electrical stimulation studies suggest that cells in the pretectum and accessory optic system are direction sensitive and mainly responsible for the horizontal optokinetik reflex (Masseck and Hoffmann, 2009). Kubo et al. (2014) managed to measure functionally different response types to horizontal whole field motion of these cells with calcium imaging. While the exact stimuli that elicit or inhibit the firing of these cells still remain unclear, the study gives a detailed insight into how the information from both eyes of the fish could be processed to compute the measured responses.

1.1. Sparse coding as a model for receptive fields

In another aspect of visual processing in animals, however, the response characteristics of neurons at intermediate levels of processing are well known. Receptive fields of cortical simple cells can be described extraordinarily well by two-dimensional gabor wavelets (Marĉelja, 1980). With sparse coding there also exists a robust mathematical model which describes the formation of very similar receptive fields from natural image data (Olshausen and Field, 1996).

One basic assumption of the model is that an image I can be represented in terms of a linear superposition of basis functions ϕ_i:

$$I(x, y) = \sum_i a_i \phi_i(x, y) \tag{1.1}$$

It turns out that these basis functions ϕ_i resemble the receptive fields of cortical neurons if they are sparse, meaning the statistical dependencies between the functions are low or, in other words, the number of nonzero coefficients a_i for all images in a set of images is as small as possible. Such functions ϕ_i can be found by minimizing the energy function for all images:

$$
\begin{aligned}
E(a_i, \phi_i) &= -[\text{preserve information}] - \lambda[\text{sparseness of } a_i] \\
&= \sum_{x,y} \left[I(x,y) - \sum_i a_i \phi_i(x,y) \right]^2 + \lambda \sum_i S(a_i)
\end{aligned}
\tag{1.2}
$$

The first part of the equation ensures that the images are captured well by the reconstruction with the basis functions. For this the distance between the original and the reconstructed image has to be minimized. The minimization of the second part forces the representation to be sparse. S is a nonlinear, generally monotonic increasing "sparsity" function. The sparsity term is multiplied with a positive sparsity factor λ which determines the importance of the sparseness in comparison to the first term.

There are several biological reasons instanced for why sparse coding might be a principle employed by the visual system, e.g. metabolic efficiency (Baddeley, 1996). Olshausen and Field mention from a statistical perspective that natural images have sparse structure, as they can be described in terms of a small number of primitives like lines and edges, so there are statistical dependencies between parts of the image (Olshausen and Field, 1997). This is also an important reason for why sparse coding can find statistically independent basis functions. If the structure of the images were different (and not sparse) no such components could possibly be found.

The basis functions generally are better at capturing such sparse components if the basis is overcomplete. This means there are more functions in the basis than linearly independent functions would be needed to perfectly reconstruct the input. For example, if a black and white image is to be reconstructed by a linear combination of images of the same size, a complete basis would need as many functions as the image has pixels. The ratio between the size of the sparse coding basis and the complete basis is also termed the *overcompleteness* of the basis and is an important parameter for the output of the algorithm (see also Olshausen and Field, 1997).

1.2. Hypothesis and content

In the case of whole field optic flow there is good reason to assume that there are such sparse components to be found. Originating from egomotion, highly statistically dependent motions in the visual field can be expected. In addition, moving objects in the world typically are spatially extended and sparse and thus optic flow from these motions should carry localized sparse information. Therefore we reason that it is certainly possible

for neurons, which are reacting to optic flow stimuli, to employ a similar mechanism to determine their receptive fields.

We furthermore expect functionally different sparse components, similar to the responses measured by Kubo et al. There should be basis functions that are monocular, spatially localized and unidirectional, specific for optic flow elicited by moving objects. From egomotion we expect basis functions that are specific for either translation or rotation. These functions should be more complex and spatially more extended than the other ones. Also they should not only be restricted to monocular stimuli, since in the case of egomotion optic flow on both sides of the fish is correlated.

To test this hypothesis this thesis will be concerned with the computation of sparse components in natural optic flow data. In addition, we will qualitatively compare the results with the data measured by Kubo et al. (2014). Therefore, the whole part of the model that produces the optic flow data will approximate the visual circumstances for zebrafish. A big part of the thesis will cover how an appropriate dataset was constructed.

1.3. The locally competitive algorithm

In the thesis the locally competitive algorithm (LCA) will be used instead of the original formulation by Olshausen et al. (Rozell et al., 2008). The LCA involves neural network components and thus can be computed very efficiently on modern graphic cards. It solves a family of sparse approximation problems and its neural elements "could potentially be mapped onto the neural circuitry of sensory cortical areas, such as V1"[1]. To emphasize this resemblance we will refer to the layer of neurons which stand for the activations a_i of weights ϕ_i[2] as the *V1-layer*[3]. In the same sense we will use the terms *weights*, *basis functions* and *receptive fields* interchangeably. Activity u_m of neurons in this layer, when presented with an input stimulus, evolves in accordance with the ordinary differential equation:

$$\tau \dot{u}_m(t) = -u_m(t) + b_m(t) - \sum_{n \neq m} G_{m,n} a_n(t) \tag{1.3}$$

Here τ is a universal time constant determining the speed of the development. b_m describes how well m fits the input and $-u_m(t)$ takes the role of a leaky integrator. This whole first

[1] Rozell et al. (2008)

[2] The weights of the connections between the image layer and the V1-layer are equivalent to the basis functions, hence we use the same symbol.

[3] Even though we model the neurons in the pretectum of the fish.

part of the equation can be seen mostly analogue to the error minimization term of (1.2). Neurons and their weights that do not fit well on the input will fire less in an equilibrium state than neurons that do. The last part of the equation is analogue to the sparsity term of (1.2). $G_{m,n}$ denotes the similarity between the receptive fields of two neurons m and n, and $a_n(t)$ is the activation of neuron n after applying an activation function to its activity $u_m(t)$[4]. This function ultimately determines the cost-function of the LCA and typically is linear except for a threshold λ below which the neuron is inactive. λ then acts in a similar fashion as the sparsity factor in equation (1.2). Similar active neurons inhibit each other, forcing the representation to be sparse; the higher λ, the sparser the representation. The exact architecture of the neural network will be described in the next section.

It has to be added, that the notion of a receptive field is to be used with a grain of salt in this context. As there is inhibition between cells when the activations are calculated, every cell can be thought to have an additional inhibitory part in its receptive field that cannot be observed in the weights to this cell, but rather is a property of the whole set of weights of cells. In other words, the weight ϕ_i of a cell does not determine the activation during a stimulus completely by itself and therefore only gives an approximation of the entire receptive field of the cell. This should be kept in mind when the resulting weights are analyzed.

2. Methods

The model that was used to compute the receptive fields can be roughly divided into three stages. In order to provide meaningful input for the sparse-coding network a realistic virtual-reality simulation of the natural environment of zebrafish was created. In the first stage this simulation generates binocular image-sequences from the viewpoint of a fish. In the second stage these images are used to calculate the input for the sparse-coding model. For this purpose an algorithm computes the optic flow in these sequences, which in turn is mapped onto two virtual retinas to generate the input-data. The last stage of the model is a neural-network implementation of the sparse-coding model.

As a remark, these stages have not been chosen arbitrarily, but rather can be seen to approximately correspond to stages in the visual processing of real zebrafish. The 3D-simulation is obviously analogue to the surrounding world. The second stage is a counterpart to the eyes of the zebrafish. It is known that the firing of retinal ganglion cells

[4]We distinguish between the activity $u_n(t)$ of a neuron n, denoting its general state, and its activation $a_n(t)$, equivalent to a_i, which determines the contribution of the neuron to the reconstructed image.

already contains information about perceived optic flow (Barlow and Hill, 1963; Lowe et al., 2013), thus we assume a very basic representation of the optic flow here. In this stage we will furthermore model the anatomy of the eye to a certain extent to create meaningful data. The last stage, the neural network, can be seen as representing parts of the pretectum of the fish. Within the diencephalon and the accessory optic system direction sensitive cells are abundant and, as has been shown in a range of vertebrate species, receive direct input from retinal ganglion cells (Scalia, 1972). These cells are the ones of which reactions to optic flow stimuli have been measured by Kubo et al. (2014). In section 3 we will test these virtual neurons in the same manner.

2.1. Simulating a moving fish in a realistic environment

We set several requirements for a dataset of realistic image sequences: 1. It should reproduce most of the statistical properties of the (standard) motions of zebrafish. To keep the model simple we refrained from modeling special movements like prey hunting. We also chose to keep the eyes of the fish fixed to a certain orientation. 2. There should be a separation between foreground and background, resulting in different optic flow speeds. 3. There should be motions in the images, independent from the motions of the zebrafish. 4. There should occasionally be spots without motion due to ambiguity of the stimulus material.

The simulation was programmed entirely in *blender*[5], an open source 3D graphics and animation software and game engine. Two cameras attached to a simple fish model capture an image sequence of two binocular images in the course of a short physics simulation.

2.1.1. Camera parameters

The cameras are placed nasalward of the rotation (i.e. weight) center of the fish. Each camera is also shifted outwards by a short distance. The ratios of these parameters have been estimated in accordance with rough anatomical data (Westerfield, 1995; Kimmel et al., 1995) where the center of rotation of real fish was estimated to lie slightly caudal of the eyes. Zebrafish have a viewing angle of approximately 160°. The cameras record a square image, so the angle from the midpoints of the edges to one another was set accordingly. As we did not model eye motions, the cameras were set to a relatively neutral orientation. The cameras are rotated to the front in the horizontal plane, so that

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

there is an 45° overlap in the fields of view, which was also assumed by Kubo et al. for their tests.

2.1.2. Modeling motion

There were few sources available that helped modeling motion trajectories for our case. The majority of parameters for this very important part of the simulation have been carefully estimated. We also decided to model the movement behavior of adult fish, even though it is different from the behavior of the measured larval zebrafish.

During neutral behavior the distribution function of both translational and rotational speeds of zebrafish is approximately exponentially decaying, e.g. there is a high probability of a fish moving with low speed that drops significantly for faster motions[6]. The fish alone almost exclusively generate forward translation and their rotations are mainly restricted to the horizontal plane (Dunn et al., 2016; Li et al., 2013; Palmér et al., 2016). However considering a natural habitat, there is a probability of the fish being driven by external forces, which can result in anomalous motions in any direction.

All of these considerations were condensed into a simple physical simulation. The model of the fish is placed randomly and accelerated by a random rotational and translational impulse. The sizes of the impulses are uniformly distributed with their maxima adjusted to the values found in the literature. Their directions follow the estimates of the preceding paragraph and also have impact on the strength of the impulse[7]. After this initialization of the simulation the fish's motion is dampened for a short while before starting the recording of the image sequence. This results in a higher probability of slower motions, just as it is observed for real fish. The dampening is proportional to the velocity of motion, thus the speed of motion decays exponentially.

2.1.3. Surrounding elements

Zebrafish can commonly be found in shallow, slow floating waters, often mentioned are rice fields (Spence et al., 2008). To account for this natural habitat we used two different boundaries in which the simulation takes place and which mostly define the optic flow elicited by distant regions. In one case clear water permits the visibility of these distant regions, whereas in the other case muddy water occludes them (Figure 2).

[6] It is to mention that the speed of motion is irrelevant for the sparse coding, as the algorithm itself is scale-invariant. However, the ratios of translational and rotational speeds and of independent motions, as well as the probability distribution of different kinds of motions is relevant.

[7] The calculation of these random vectors is noted down in appendix B.

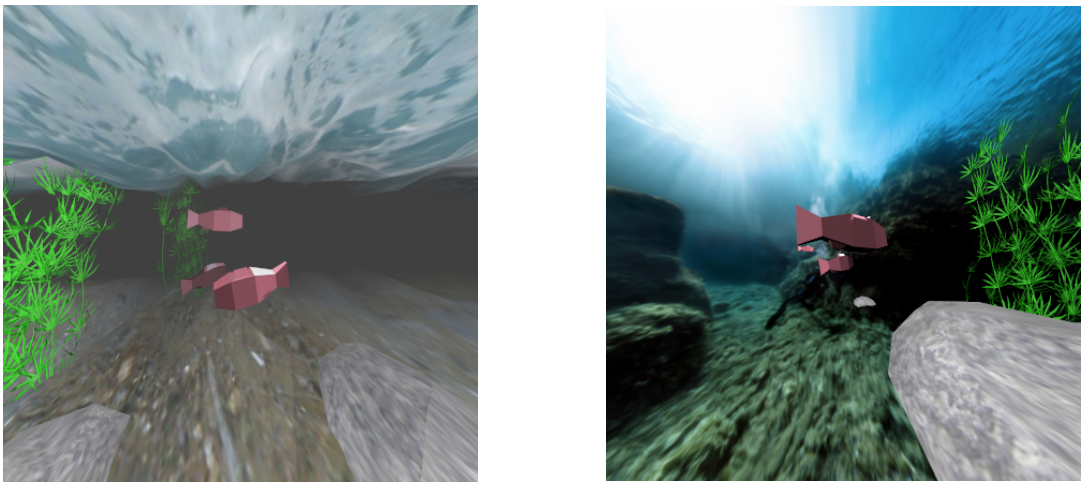

Figure 2
In one simulation setup the fish is surrounded by clear water and swims in a textured sphere with flattened out floor (right), in the other case upper and lower boundaries are flat and distant objects are occluded (left).

To generate independent motion and motion parallax in the simulation we introduced three additional objects to the simulation. Stone-like and a seaweed-like objects allow motion parallax to be perceived. Several simple fish are simulated similarly to the model of the zebrafish in its vicinity. This provides optic flow features independent of the zebrafish's motions. All additional objects are scaled and distributed randomly in the environment without overlapping with the cameras (Figure 2).

2.2. The virtual eye

After creating the images, the data needs to be prepared to be fed into the sparse-coding model. This part of the model has proven to be troublesome for various reasons. To understand why, it is useful to clearly state what we have and what we want.

The input for the sparse-coding model should ideally be a set of vectors corresponding to the firing of direction sensitive ganglion cells in the retina of the fish. These virtual ganglion cells should be plausibly distributed in the field of view of the fish. In order to create these vectors we will use the images created earlier and some opportune algorithm to calculate the optic flow on the image plane.

This raises two major problems: 1. The optic flow algorithms tested only compute the flow reliably if the moving parts of the image sequence are approximately equally large. If this is not the case very small and very large motions will generally be treated differently by the algorithms, resulting in unstable behavior in our case of application. Unfortunately,

the images which have been rendered by blender are heavily distorted due to the large viewing angle. This is something we have to compensate for before calculating the optic flow. 2. The ganglion cells in the fish are distributed on an approximately spherical retina. The projection from a plane to a sphere is a nontrivial one in which compromises have to be made.

These difficulties could have been avoided by calculating the optic flow directly on a sphere, however, this approach is computationally expensive and would have taken too much time to be calculated on a large dataset. The approach ultimately chosen is not flawless, but the compromises made will be justified.

2.2.1. Compensating for distortion

The images are transformed in order to enhance the performance of optic flow algorithms. This is not yet the mapping from the plane to the retinal sphere even though a sphere projection provides the idea for the transformation. However this transformation will be affecting the vectors, resulting from the optic flow algorithms, and therefore will be important for the following parts of the model.

The images rendered in the simulation are calculated with a pinhole camera in respect to a plane as the projection screen. This explains the distortion, increasing with the distance to the center of the projection (Appendix E.2, figure 10). Consider a sphere as the projection screen of the camera with the pinhole in the center of the sphere. In this case no distortion occurs; all rays hit the surface orthogonally. This suggests a sphere projection can be used to compensate for the distortion of the rendering. We first project the image onto the sphere, then project it back onto the plane with another sphere projection with less distortion in respect to the sphere.

Sphere projections can have one ore none of the two following properties which seemed to be relevant in this case: conformal (angles are locally preserved) or equal-areal (sizes of areas are preserved). To account for the way the pinhole camera operates, an azimuthal projection was chosen. In this kind of projection the azimuth of the sphere coordinate in respect to a central point is preserved. The projection used in the thesis is an azimuthal conformal projection, the stereographic projection[8]. The advantage is that motion vectors, mostly describing small motions, are locally orthogonal if the motions on the sphere are orthogonal. However, vectors on the sphere that are the same length do not have to be of the same length in the projection, which will be corrected later on.

[8] Against first intuitions the equal-area projection distorts too strong in areas distant to the center.

Let (r_1, ϕ_1) be the polar coordinate on the rendered image, f the focal length of the camera[9]. We obtain the coordinate on the sphere[10] (f, θ_s, ϕ_s) with radius f by

$$(f, \theta_s, \phi_s) = (f, \arctan(\frac{r_1}{f}), \phi_1) \tag{2.1}$$

To get the coordinate on the resulting image (r_2, ϕ_2) we apply the stereographic projection:

$$(r_2, \phi_2) = \left(2f \cdot \tan\left(\frac{\theta_s}{2}\right), \phi_s\right) \tag{2.2}$$

2.1 and 2.2 together yield:

$$(r_2, \phi_2) = \left(2f \cdot \tan\left(\frac{\arctan(r_1/f)}{2}\right), \phi_1\right) \tag{2.3}$$

This transformation is periodic, hence the images were cropped. While the corners of the resulting images show parts of this periodic recurrence (e.g. appendix E.2, figure 11), it makes no difference for the optic flow calculation. For the processing of the images *OpenCV*[11] was used.

2.2.2. Optic flow calculation

We decided to use a dense optic flow algorithm, meaning the algorithm estimates the motion vector for every pixel in the image. For the final results the *deep flow* algorithm (Weinzaepfel et al., 2013) was used, as it provided the most stable and a very fast calculation. *Deep flow* is based on a deep convolutional neural network architecture. It is preceded by *deep matching*, a deep convolutional neural network as well, which finds matching spots in images even if they are distant. The algorithm thus is able to handle large displacements in image sequences.

2.2.3. Virtual ganglion cells

To simulate the location of receptive fields of ganglion cells in the field of view of the fish we first find the coordinates of corresponding points on a simplified retina. We then find the position of these points on the image, where the optic flow has been calculated.

[9] The distance between the pinhole and the image plane, a constant determined by the 3d-simulation.
[10] The point of intersection of the same ray of light coming through the pinhole.
[11] An open source image processing library: http://opencv.org/

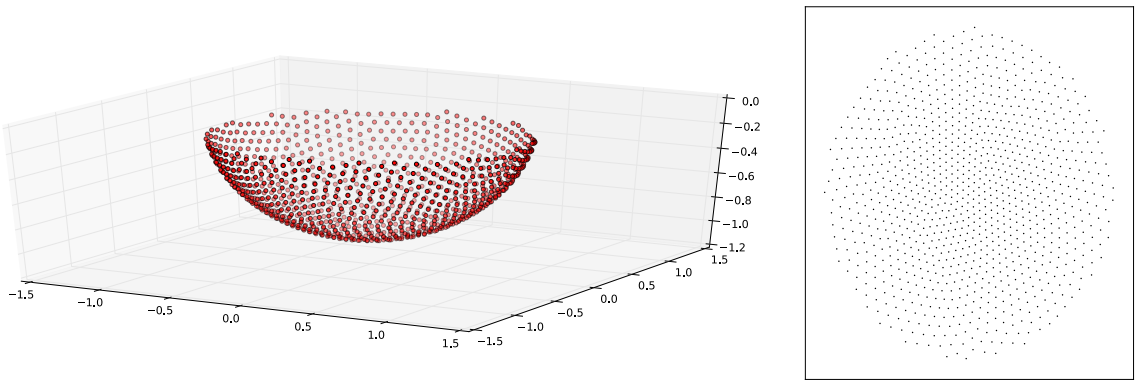

Figure 3
Left: 3D plot of the positions of the virtual retinal ganglion cells.
Right: Distribution of virtual ganglion cells on the projected sphere.

Zebrafish have a rather complex retina in terms of a 3d-simulation (Haug et al., 2010; Schmitt and Dowling, 1999). We therefore made some simplifying assumptions: 1. The retina is shaped like a perfect sphere with the pinhole in the center of the sphere – in fact the shape of the zebrafish's eye is close to a hemisphere. 2. All ganglion cells cover about the same area. 3. The fish has 1024 groups of four ganglion cells, which we will treat as one point for now[12]. 4. There is evidence that retinal ganglion cells in zebrafish are specific for one of three directions of optic flow (Lowe et al., 2013). We simplified the calculation by assuming four principal directions[13].

For this purpose n points are placed on the unit sphere, repelling each other with forces reciprocal to the square of the distance of the points. The force $\vec{f_p}$ on point p from the set of points P is thereby:

$$\vec{f_p} = \sum^{q \in P \setminus \{p\}} -\frac{\frac{\overline{pq}}{\|\overline{pq}\|}}{\|\overline{pq}\|^2} = \sum^{q \in P \setminus \{p\}} -\frac{\overline{pq}}{\|\overline{pq}\|^3} \tag{2.4}$$

The viewing angle of the simulated zebrafish is 160°. We choose n in such a way that there will be 1024 points in an area A_R, a spherical cap which covers 160°, resembling the retina. As the cells are all assumed to cover the same area in the end, the expected number of points in the retina is:

[12] This number is hard to estimate, especially as the number of direction sensitive ganglion cells seems to be changing depending on the direction (see Lowe et al., 2013). However we don't expect this to have a big influence on the results, as long as the number of ganglion cells is not too small.

[13] We figured, the most important fact at this point is that there are no negative vectors, i.e. two nonzero vectors in different directions cannot add up to zero. The vectors of the optic flow can admittedly be reconstructed by addition in a different way in both cases.

$$n_R = \frac{A_R}{A} \cdot n \tag{2.5}$$

with the surface area of the unit sphere $A = 4\pi$ and

$$A_R = 2\pi \left(1 - \cos \frac{160^\circ}{2}\right) \tag{2.6}$$

Rearranging 2.5 for n and substituting results in:

$$n = \frac{1024}{1/2\,(1 - \cos 160^\circ/2)} \approx 2478.3 \tag{2.7}$$

n is rounded to 2479 to reduce the probability of a point lying outside. After the simulation settles down to an equilibrium we take the 1024 points with the lowest z-coordinate to represent the cells on the retina P_R (Figure 3, left). A quick test reveals that $\max_{p \in P_R}(\alpha(p)) \approx 79.9^\circ$ with $\alpha(p)$ the angle of p to the z-axis, thus there are no outliers.

2.2.4. Preparing the vectors for the network

The optic flow calculated in 2.2.2 is already smoothed out by *deep flow*. The vector for every ganglion cell is taken from the center of the cell at the corresponding location on the image (see figure 3). The vectors at these points are split up into 4 values: The up, down, left and right component of the vector on the image.

With the projection these vectors are scaled and rotated differently compared to corresponding vectors calculated on the sphere. Under rotation the sparse coding is invariant (see appendix A), but the scaling has to be corrected since it would alter the results.

We normalize each vector $\vec{v}_i$ at the point of the cell $\mathbf{x}_i$, before splitting it up, by the directional derivative of the inverse projection F^{-1} in the direction of $\vec{v}_i$

$$\vec{u}_i = \left\| \nabla_{\vec{v}_i} F^{-1}(\mathbf{x}_i) \right\| \cdot \vec{v}_i \approx \left\| \frac{F^{-1}(\mathbf{x}_i + \epsilon \frac{\vec{v}_i}{\|\vec{v}_i\|}) - F^{-1}(\mathbf{x}_i)}{\epsilon} \right\| \cdot \vec{v}_i \tag{2.8}$$

with ϵ being very small. The split up vectors are then normalized altogether to values between 0 and 255 and saved in 8 *.png*-files – 4 for each eye – as input for the sparse coding.

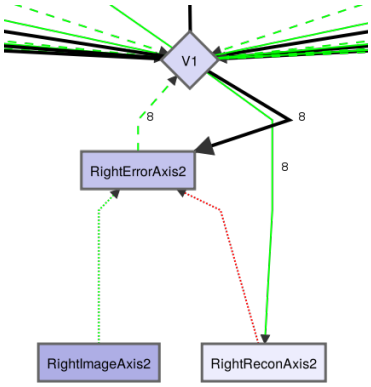

Figure 4:
Graphical illustration of a part of the sparse coding network. Rectangular and diamond shaped boxes denote neural layers while arrows represent connections between the layers. Numbers on the arrows indicate if the same weights for different connections are used.

2.3. Sparse-coding neural network

In the model we used *Petavision*, an open-source neural-network simulation toolbox containing an implementation of the LCA. The architecture of the network is outlined in figure 4. For each of the 8 input-files the same V1-layer (in other terms, the same coefficients a_i) was used to calculate the sparse representation of the input, but with different connection weights (ϕ_i) for each input. In the figure only one of these 8 symmetrical inputs is depicted, while the others are only indicated by their connections to V1 in the upper part of the picture.

The computation of the network on one input proceeds in 5 steps: 1. The input is copied over from the input layer (*RightImageAxis2* in figure 4) to the error layer (*RightErrorAxis2*). 2. The sparse representation of the activations in the error layer is calculated by V1 in accordance with equation (1.3) until an equilibrium is reached. The weights (ϕ_i) for this calculation are saved in a connection from V1 to the error layer (bold black arrow) and only copied over to the connection in the other direction. 3. The sparse representation is projected out to the reconstruction layer (*RightReconAxis2*) using the weights ϕ_i. 4. The error between the reconstruction of the image and the original in the error layer is calculated by subtracting (inhibiting) the layers (red arrow). 5. The error gives the gradient which allows for an adaptation of the weights stored in the plastic connection between V1 and the error layer via a hebbian learning rule.

The connection between the error layer and V1 is often chosen to be a convolution, so one neuron in V1 is connected only to a small patch of neurons in the other layers and shares its weights with neurons covering other patches. This has the advantage that less weights are necessary to reach a certain overcompleteness factor. As we wanted one receptive field for the whole binocular field of view, in our case one neuron in V1 is connected to all neurons in the other layers.

With the LCA it is possible to retain the activations of the neurons in V1 from one input to the next. This can be used to speed up computations by using one image as the prior for another image if they are related, e.g. images from a movie. As our inputs were unrelated we chose to disable this feature.

3. Results

Before recreating the experiment of Kubo et al. the performance of the LCA was analyzed and enhanced. The two main parameters that were adjusted to configure the performance of the network are 1. the number of neurons in V1 and 2. the threshold of these neurons, i.e. what activity they have to reach until they are considered in the reconstruction of the input. In terms of sparse coding 1. is equivalent to the number of vectors in the sparse coding base and 2. approximately to the sparsity factor λ (equation 1.2).

As of yet there is no literature to be found on whole field sparse coding specificities in binocular optic flow data. The only evidence if the computations have been set up properly, was the general behavior of the network and the resemblance of the measurements when recreating the experiment. As one can see in the work of Schultz et al. (2014) a good predictor for the quality of the weights is the location of reconstructions of inputs in a sparsity vs. error plot (Appendix E.1, figure 9).

In the following tests we considered weights computed by a network with an overcompleteness of 2 (16384 neurons in V1) and a threshold of 0.04. Another network with an overcompleteness of ½ (4096 neurons) and a threshold of 0.02 was also tested and it showed that this second setup was better at reproducing the results of the experiment of Kubo et al. Although it did not find the sparse components we expected as explicit as the other setup, we still considered it as an alternative result (Appendix E.5), showing that the model is capable of a fairly accurate reconstruction of the measurements, apart from some details. The reason for the difference between the setups and what it implies is not clear and could be part of further investigations.

3.1. Resulting sparse components

For the interpretation of the weights that have been computed we displayed them in the same way as the optic flow of the image sequences, projected onto the plane with the stereographic projection. The four vectors are combined to one vector by addition. It is certainly possible, that there could be 2 nonzero vectors on the point of a weight in

opposite directions. This however is very unlikely. As this case never occurs in the dataset it would always increase the error. Since one cell covers the whole binocular field of view we can see the location and direction of the optic flow vectors that increase the activation of the neuron. In appendix E.6 a selection of these weights is displayed.

We could observe the formation of weights that are spatially localized and unidirectional, both monocular and binocular, as we hypothesized. They are noisy at times, meaning small vectors in different directions tend to occur randomly in the flow. The sizes of the unidirectional flow components vary between only one vector (i.e. ganglion cell) and a medium sized cluster of vectors. Very large unidirectional components, covering the majority of the field of view as they have been observed by Krapp et al. (1996) in the blowfly, haven't been found. The reason for this should be that subtraction of vectors is not possible, hence there is no use in such large components, because they almost never occur in natural optic flow without spots of slower flow[14].

A bit of displeasure is caused by the large number of weights, that showed combinations of the components we hypothesized, but in a manner that made it very unlikely for these flow fields to occur in a natural environment. Hence, they should not be sparse components of natural optic flow. During the experiments it showed, that these weights become less common if the network is set up properly and the dataset is made more diverse and, even more important, larger. The 16384 neurons each have 8196 weights - 4 to each vector of the input - resulting in a model with about 67 million dimensions. It is very likely that the dataset we used was just too small to adjust these parameters to capture natural sparse components. This should be taken into account when further investigations are planned.

3.2. Testing the receptive fields

To analyze the receptive fields the LCA generated we recreated a part of the experiment of Kubo et al. in virtual reality. Two hemispheres were placed on each side of the model of the fish (Figure 5). In a 45° angle in front of the fish, where the field of views of both eyes are overlapping and no stimuli were presented in the original experiment, the hemispheres were truncated. They were textured with dark vertical lines on a bright background and animated in such a way that a uniform rotational motion either in nasal or temporal direction could be perceived. Image sequences from this second virtual reality setup were

[14]It has to be added that the fly has much less cells available (Krapp et al. (1996) speak of 60 neurons in the third visual neuropile) that compute the desired components. This could explain the need for bigger, but less specific receptive fields.

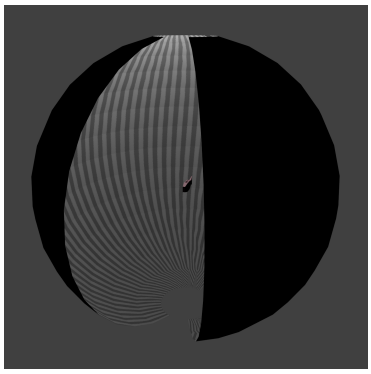

Figure 5: Outside view onto the virtual arena for creating the test stimuli with the fish in the center.

processed in the same manner as the ones of section 2.1[15]. From these simulations 4 different unilateral stimuli and 4 stimuli specific for horizontal rotational or translational motion could be created (Figure 6; see appendix E.3 for an exemplary image sequence).

As the activations of neurons in the V1-layer were determined by the stimulus, the neurons could be grouped based on the combination of reactions (firing or not firing) they showed to the 8 test stimuli. This led to a total of $2^8 = 256$ groups (Figure 7). Kubo et al. determined the belonging of a neuron to a group by finding the highest correlation of its calcium responses, i.e. its firing behavior, with idealized responses of neurons from these groups. In contrast, as there was no time course to be correlated, we determined a neuron to be firing if it crossed the threshold and contributed to the reconstruction of the stimulus. The virtual neurons then were assigned to the according group.[16]

In the course of their paper Kubo et al. decided to categorize and name response types in accordance to their functional implications (Figure 8), as well as describe them in terms of computational complexity.

[15] It turns out that the optic flow resulting from these calculations is not exactly uniform (Appendix E.3, Figure 14). The possible effects of this fact will be discussed later.

[16] As a side note, the major part of the virtual neurons did not respond to any of the stimuli. 12802 of the 16384 neurons have therefore been excluded from the analysis, reactions of only 3582 neurons have been measured. Kubo et al. found about 500 neurons consistently reacting to at least one of the stimuli per fish. In the areas they considered, these make up about 1/20 of the entire population of cells (Appendix C). If the proportion of reacting and non-reacting direction selective cells would be about the same, this would mean that approximately 1/6 of the cells in the considered areas of the fish could be direction selective. Ultimately, this indicates that the proposed model would not need more neurons to conduct the desired computations than the fish has available, if it was set up accordingly.

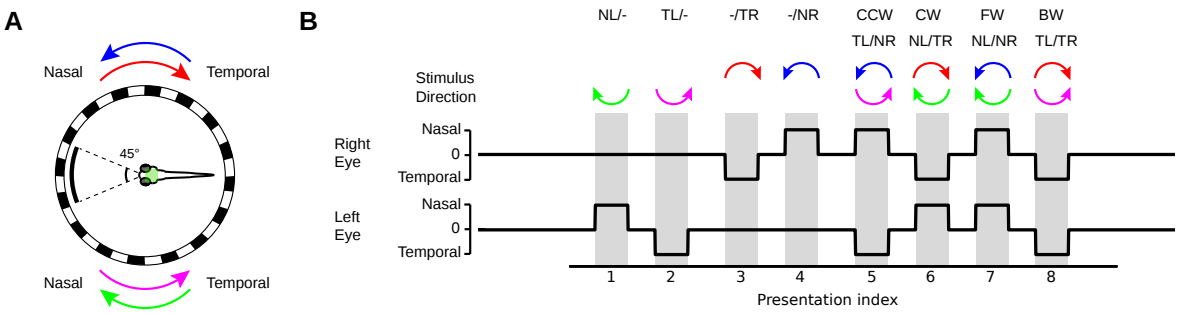

Figure 6: Outline of the test setup.

A. Schematic diagram of the experimental setup that also was used by Kubo et al. In the 45° angle in front of the fish no stimuli were presented.
B. Protocol of the stimuli that were tested. NL/- indicates nasalward motion to left eye; TL/- indicates temporalward motion to left eye; -/TR indicates temporalward motion to right eye; -/NR indicates nasalward motion to right eye; CCW indicates counter-clockwise; CW indicates clockwise; FW indicates forward; BW indicates backward (These are the perceived motions, not the motions of the fish).

3.2.1. Combinational logical complexity of responses

Kubo et al. reasoned that simple logical operators might be involved when the output of retinal ganglion cells is processed by cells in the pretectum. For the purpose of the experiment they used the 4 basic motion stimuli (NL, TL, NR, TR; see Figure 6) and combined them with 3 simple logical operators, AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). The resulting formulas are proposed to describe the neural circuitry and its complexity analogous to the computations in digital electronics. The number of logical operators in them can be seen as an estimate of the degree of complexity of the response types. In the LCA some kind of logical reasoning can be assumed during the calculation of the activations of the V1-neurons, though this admittedly ought to rest on different principles than computations in the fish.

The response types of the neurons then can be described as the output of these formulas. Kubo et al. give several examples for this (Appendix E.4, figure 16C). A neuron reacting to a nasalward motion on the left side (NL) as well to clockwise motion (CW) and forward motion (FW) can be described by a logical formula of the most basic kind, which is simply equivalent to the NL input and involves zero logical operators. If a neuron only reacts to the two stimuli NL and FW and not to CW the formula gets more complex as the firing of the neuron is inhibited in the CW case, resulting in NL $\wedge \neg$TR (i.e. two operators). To determine the simplest formula describing the reactions of the neurons the Quine and McCluskey algorithm was used (Appendix D). The number of logical operators of the response types in this case ranges from 0 to 19, where most of the responses need rather few operators (Figure 7C).

Figure 6 taken and slightly modified from Kubo et al. (2014)

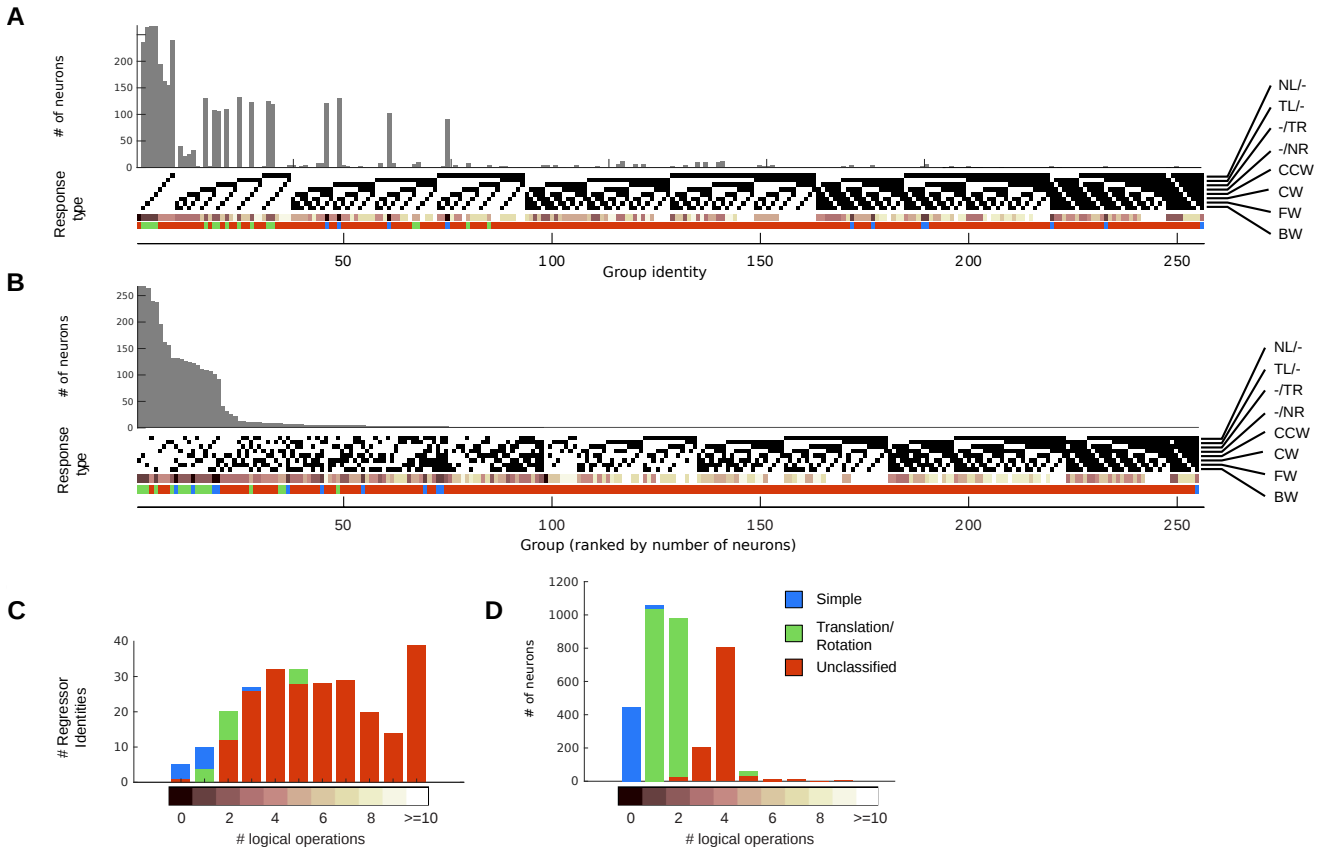

Figure 7: Distribution of response types and computational complexity.

A. A histogram showing the number of virtual neurons classified according to the 256 possible response types. The white-and-black plot below the histogram illustrates the response profile of each group/response type. Each vertical 1 x 8 line represents one response profile, and the squares indicate whether the response type is active (black) or inactive (white) during the stimulus phases indicated on the right. The copper-colored line shows the computational complexity of the response type, as in **C**.
B. The histogram from **A** sorted by the number of neurons found with the particular response type.
C. Quantification of regressor complexity. The 256 possible response types are binned according to the number of logical operations needed.
D. Histogram of the number of neurons versus the number of logical operations.
The color code (blue, green, and red) in (A–D) corresponds to the one used in figure 10.

It seems like for most of the neurons in the zebrafish pretectum only few logical operations are required to describe their response characteristics (Appendix E.4, figure 16E), especially when comparing the distribution of the number of groups and their logical complexity (Figure 7C) to the distribution of single neurons. The virtual neurons in our case show a very similar tendency (Figure 7D). There is, however, in our case a large number of neurons seemingly using four logical operations to calculate their responses. Mainly neurons that are active for only one stimulus contribute to this number. We will discuss these neurons later. Their response characteristics seem to have a different origin than the responses of the other neurons.

Figure 7 and description taken and modified from Kubo et al. (2014)

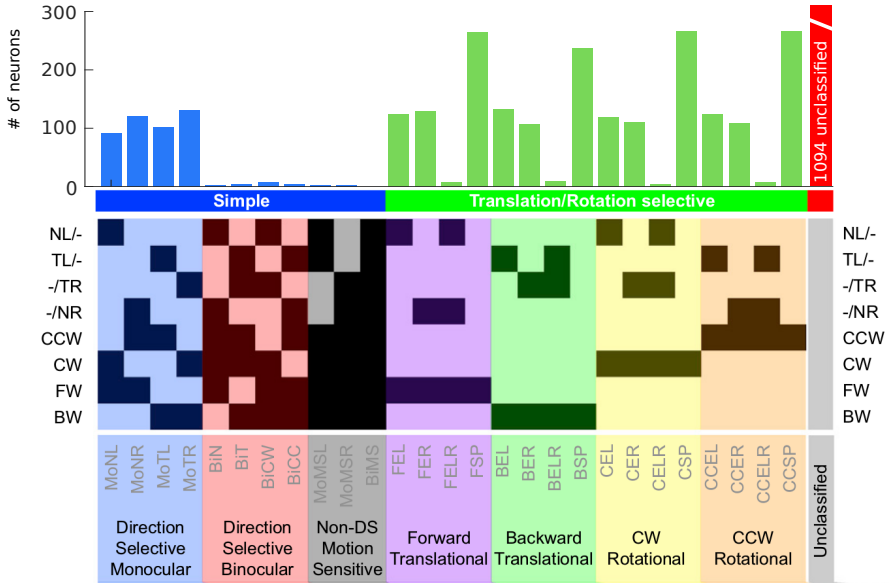

Figure 8: Functional classes of neuronal responses to the test stimuli introduced and named by Kubo et al. A black square means the cell is reacting to a particular stimulus if it is in the group indicated at the bottom.

3.2.2. Functional classes of neural responses

Kubo et al. divided the response types into three categories (Figure 8). "Simple" cells show responses whose formulas do not include any negations[17]: 1. monocular direction sensitive cells, consisting of four response types that respond to either nasalward or temporalward motion presented to either the left or right eye, 2. binocular direction sensitive cells, receiving information from both eyes, but preferring one direction over the other and 3. non-direction sensitive, motion sensitive cells.

They found a higher representation of monocular simple response types than binocular types (Appendix E.4, figure 15). Most simple cells very likely only receive input from one eye. A similar ratio can be observed for the receptive fields created in the simulation (Figure 8), indicating that the vectors of the simple category tend to be specialized for the optic flow on one side. Another observation, which deserves attention as it can explain a range of findings, is that there are more neurons to be found which react to fewer stimuli. This behavior is forced by the sparseness of the representation. While the receptive field of a neuron could be appropriate for the input, it can be inhibited by the reactions of other neurons (Equation 1.3).

The second category of translation and rotation selective responses contains cells that

Figure 8 taken and modified from Kubo et al. (2014)

[17]Their reactions can be thought of being calculated without the use of inhibition.

show responses to only one of the four binocular stimuli (FW, BW, CW, and CCW). In the case of the neuron responding also to a monocular stimulus, the response still is specific for one rotational or translational stimulus. For example a neuron with FEL characteristic responds to NL and FW but not to CW (Figure 8), even though the CW stimulus contains the NL component (Figure 6). In this case the neuron is apparently being inhibited by motion that doesn't fit the rotation or translation it is specific for.

In the zebrafish Kubo et al. found a higher number of cells specific for translation than for rotation (Appendix E.4, figure 15). Since the cells in the pretectum are responsible for the optokinetic reflex, they reason that this reflex in the case of rotation is mainly triggered by monocular cells, hence cells responding to rotation are thought to be monocular in the fish. In contrast, in the case of translation they should respond to binocular stimulation. In the model we couldn't find such a distinction between rotation and translation (Figure 8). Though, it is apparent that neurons that respond to more stimuli (e.g. FELR) are less likely than neurons that respond to fewer stimuli (e.g. FSP). This again seems to be a characteristic of sparse coding.

In addition, Kubo et al. found about 156 neurons that responded to some of the stimuli, but it was not clear what they were specific for, so they were marked as unclassified. This number is very low, considering that these neurons express one of the 229 other response types. In total about ⅓ of the measured cells have not been assigned to one of the specific response types. This matches quite well with the model, where 1094 of 3590 neurons, which were active during the stimulations, were unspecific. In contrast, the biggest part of these unspecific neurons in the model are in one of few response types (Figure 7A; there are only few large unspecific "red" groups). In the fish they are more divers (Appendix E.4, figure 16A).

The entire distribution of neurons with their response characteristics is shown in figure 7. The corresponding figure from Kubo et al. can be found in Appendix E.4, figure 16. Both distributions show similar properties. The functionally specialized response types, indicated by blue and green in the colored bar below the distribution, are highly favored over non-specialized responses. When arranging the response types by the number of neurons with this type, this results in a similar distribution (i.e. there is a high probability of a neuron to exhibit one of only few response types, that drops significantly for not so common responses) with the specialized responses being among the most common ones. In appendix E.5 this resemblance is even closer. Moreover, the preference for response types with fewer logical operations is more clearly visible in these plots (The copper-colored bar below the distribution, indicating this number, begins darker at the beginning of the distribution and gets brighter towards the end). The biggest difference is the large

number of neurons showing an unspecific response type with only one response to a single monocular stimulus in the model, that can not be found in the fish.

3.3. Discussion

Based on these and further results, covering the location of functionally specialized cells, Kubo et al. proposed a neural circuit model for the processing of binocular optic flow in the pretectum of the zebrafish, mainly described by logical operations[18]. The output of neurons with simpler receptive fields is thought to be be used later in the processing of more complex neurons. It is apparent that this sort of architecture is not appropriate for our model; the way the receptive fields are computed is known and it is not hierarchical. What can be compared are the functional specifications of the involved receptive fields. If the receptive fields, used to reconstruct (or analyze) the optic flow, are the same in the model and the fish, this would implicate that their functional behavior, i.e. their dedicated responses to certain optic flow circumstances, would be the same. Seeing that this is the case, that in both of them receptive fields with similar functional implications (specific for translation or monocular unidirectional motion) can be found, suggests that the same optic flow components could account for the structure of these receptive fields.

There are several possible reasons for why the artificial neurons of the LCA don't behave exactly like the direction sensitive neurons in the pretectum of the zebrafish, given they would indeed use the exact same mechanism to determine their receptive fields. First of all, the technical components of the model have their flaws, especially the calculation of optic flow is still considered an open problem (this becomes apparent when really the test stimuli should be an uniform motion, but they are not; Appendix E.3, figure 14). However, consistency of the results over a variety of setups (e.g. different optic flow algorithms) suggests that they have no systematical impact.

More important are the properties of the dataset that was created with the 3D-simulation and the virtual retina. It is hard to tell which of the simplifying assumptions is influencing the resulting sparse components. For one thing, the simulation itself could be too simple and more components with different properties and a more diverse behavior of the modeled fish could be required. An indicator for this is the error-sparsity plot (Figure 9). The difference in the error between a setup with $1/2$ and 4 times overcompleteness is very small. This could mean that many of the independent components of the data already can be approximated with few basis functions. Also, the reasons for why the weights at times don't behave as we expected, instanced in section 3.1, could apply here. Even though we

[18] Somewhat like a neural net of McCulloch and Pitts-Neurons.

could find the sparse components we expected, more effort could be necessary before the optic flow in the dataset can be considered natural and sufficiently divers.

Secondly, another great effect could come from the immobility of the eyes. Zebrafish can compensate for rotational egomotion through the optokinetic reflex, which also means that there is less need for binocular receptive fields detecting rotation. As Kubo et al. remark, the optokinetic reflex is likely to be driven by monocularly sensitive neurons since each eye can be controlled individually. It is obvious that there is a discrepancy between the model and the fish, as the model tries to detect optic flow components resulting from rotation, while the fish rather minimizes their effect onto the entire optic flow. One could hypothesize that the distribution of binocular translational and rotational specific neurons become more alike, if the majority of rotational optic flow is compensated for in the simulation by stabilizing the eyes. This however would lead to a lack of monocular neurons specific for rotational motions that seem to exist in the fish. A more complex model (e.g. one that actively compensates for rotational components), not sparse coding alone, could be needed to find better approximations of the receptive fields.

The great number of unspecific neurons responding to only one monocular stimulus and nothing else however are harder to interpret. On the one hand, they could point at a principal difference between sparse coding and the neurons the fish exhibits, namely the inhibition of a fitting receptive field in order to enhance the sparseness of the reconstruction. On the other hand, it could be that they are a mere artifact of the different methods used to observe the activation of the neurons and assign them to a response type. This could result in a higher number of observed neurons reacting randomly to a single stimulus in the model. Answering this question could also be part of future investigations.

4. Conclusion

In the thesis a reasonably detailed model for the optic flow perception in zebrafish was established. We managed to qualitatively reproduce measurements of neural responses in larval zebrafish with the use of sparse coding. This includes the existence and approximate ratios of functionally specialized direction selective cells in the pretectum. Some of the sparse components also express some of the properties we hypothesized. They are spatially localized and monocular if they are specific for motion of other objects and spatially extended and often binocular if they are specific for optic flow originating from egomotion. A major part of the resulting sparse components, however, is noisy and not clearly specified. Though it is likely that this can be prevented by enriching the optic

flow dataset and a better optimized sparse coding algorithm.

There still are some idiosyncrasies of the proposed model that do not fit the empirical data well, like a great amount of binocular receptive fields specific for rotational stimuli that haven't been found in larval zebrafish, or a great number of unspecific cells responding to only one monocular stimulus and nothing else. Nonetheless, the results show interesting resemblances which are hard to explain if no sparse coding process is involved at all in the formation of these direction sensitive cells. It seems plausible that, when the fish searches for appropriate behavioral responses to visual stimuli, it is advantageous to make use of their statistically independent components which are specific for certain events in the world, in order to find meaning in a stimulus; this invites further research in this direction.

Bibliography

Baddeley, R. (1996). An efficient code in v1? *Nature*, 381:560–561.

Baier, H. and Scott, E. K. (2009). Genetic and optical targeting of neural circuits and behavior—zebrafish in the spotlight. *Current opinion in neurobiology*, 19(5):553–560.

Barlow, H. B. and Hill, R. M. (1963). Evidence for a physiological explanation of the waterfall phenomenon and figural after-effects. *Nature*, 200(4913):1345–1347.

Brockerhoff, S. E., Hurley, J. B., Janssen-Bienhold, U., Neuhauss, S., Driever, W., and Dowling, J. E. (1995). A behavioral screen for isolating zebrafish mutants with visual system defects. *Proceedings of the National Academy of Sciences*, 92(23):10545–10549.

Dunn, T. W., Mu, Y., Narayan, S., Randlett, O., Naumann, E. A., Yang, C.-T., Schier, A. F., Freeman, J., Engert, F., and Ahrens, M. B. (2016). Brain-wide mapping of neural activity controlling zebrafish exploratory locomotion. *Elife*, 5:e12741.

Grienberger, C. and Konnerth, A. (2012). Imaging calcium in neurons. *Neuron*, 73(5):862–885.

Haug, M. F., Biehlmaier, O., Mueller, K. P., and Neuhauss, S. C. (2010). Visual acuity in larval zebrafish: behavior and histology. *Frontiers in zoology*, 7(1):8.

Junger, W. and Dahmen, H. J. (1991). Response to self-motion in waterstriders: visual discrimination between rotation and translation. *Journal of Comparative Physiology A*, 169(5):641–646.

Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B., and Schilling, T. F. (1995).

Stages of embryonic development of the zebrafish. *Developmental dynamics*, 203(3):253–310.

Krapp, H. G., Hengstenberg, R., et al. (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, 384(6608):463–466.

Kubo, F., Hablitzel, B., Dal Maschio, M., Driever, W., Baier, H., and Arrenberg, A. B. (2014). Functional architecture of an optic flow-responsive area that drives horizontal eye movements in zebrafish. *Neuron*, 81(6):1344–1359.

Li, Y., Lee, J.-M., Chon, T.-S., Liu, Y., Kim, H., Bae, M.-J., and Park, Y.-S. (2013). Analysis of movement behavior of zebrafish (danio rerio) under chemical stress using hidden markov model. *Modern Physics Letters B*, 27(02):1350014.

Lowe, A. S., Nikolaou, N., Hunter, P. R., Thompson, I. D., and Meyer, M. P. (2013). A systems-based dissection of retinal inputs to the zebrafish tectum reveals different rules for different functional classes during development. *Journal of Neuroscience*, 33(35):13946–13956.

Marĉelja, S. (1980). Mathematical description of the responses of simple cortical cells∗. *J. Opt. Soc. Am.*, 70(11):1297–1300.

Masseck, O. A. and Hoffmann, K.-P. (2009). Comparative neurobiology of the optokinetic reflex. *Annals of the New York Academy of Sciences*, 1164(1):430–439.

Neuhauss, S. C., Biehlmaier, O., Seeliger, M. W., Das, T., Kohler, K., Harris, W. A., and Baier, H. (1999). Genetic disorders of vision revealed by a behavioral screen of 400 essential loci in zebrafish. *Journal of Neuroscience*, 19(19):8603–8615.

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

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Palmér, T., Aström, K., Enqvist, O., and Petersson, P. (2016). Visual analysis of zebrafish behavior.

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

Scalia, F. (1972). The termination of retinal axons in the pretectal region of mammals. *Journal of Comparative Neurology*, 145(2):223–257.

Schmitt, E. A. and Dowling, J. E. (1999). Early retinal development in the zebrafish, danio rerio: light and electron microscopic analyses. *Journal of Comparative Neurology*, 404(4):515–536.

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

Spence, R., Gerlach, G., Lawrence, C., and Smith, C. (2008). The behaviour and ecology of the zebrafish, danio rerio. *Biological Reviews*, 83(1):13–34.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia.

Westerfield, M. (1995). *The zebrafish book: a guide for the laboratory use of zebrafish (Brachydanio rerio)*. University of Oregon press.

Appendix

A. Invariance of sparce coding under rotation

We can write the rotations R_i performed on the vectors $\vec{v}_i$ of the optic flow as one rotation matrix as they are not overlapping:

$$\begin{bmatrix} R_1 & \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & R_2 & \dots & \mathbf{0} \\ & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & R_n \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_n \end{bmatrix} = R \cdot \vec{v} \tag{A.1}$$

We rewrite the sparse coding cost function 1.2 with the rotated vectors and show that it remains the same:

$$E(a_i, R\phi_i) = \sum_{j=1}^{m} \left|\left| R\mathbf{x} - \sum_{i=1}^{k} a_i R\phi_i \right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i) \tag{A.2}$$

$$\Leftrightarrow \quad E(a_i, R\phi_i) = \sum_{j=1}^{m} \left|\left| R\left(\mathbf{x} - \sum_{i=1}^{k} a_i \phi_i\right) \right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i) \tag{A.3}$$

and as R is orthogonal:

$$\Leftrightarrow \quad E(a_i, R\phi_i) = \sum_{j=1}^{m} \left|\left| \mathbf{x} - \sum_{i=1}^{k} a_i \phi_i \right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i) \tag{A.4}$$

□

The sparse coding would result in the same vectors ϕ_i, though they would have to be rotated back with R^T.

B. Random vectors for motion simulation

Let R be an independent random number following the discrete uniform distribution over $[0, 1]$. The vector applied as translational impulse is calculated as

$$\vec{v} = \vec{v}_1 + \vec{v}_2 \tag{B.1}$$

$$\vec{v}_1 = \begin{pmatrix} R \\ (R - 0.5) \cdot 0.01 \\ (R - 0.5) \cdot 0.01 \end{pmatrix} \tag{B.2}$$

$$\vec{v}_2 = \begin{pmatrix} (R - 0.5) \cdot 0.1 \\ (R - 0.5) \cdot 0.1 \\ (R - 0.5) \cdot 0.1 \end{pmatrix} \tag{B.3}$$

where $\vec{v}_1$ is the motion generated by the fish and $\vec{v}_2$ a small random impulse. The applied torque is calculated as

$$\vec{u} = \vec{u}_1 + \vec{u}_2 \cdot 0.5 \tag{B.4}$$

$$\vec{u}_1 = \begin{pmatrix} 0 \\ 0 \\ R - 0.5 \end{pmatrix} \tag{B.5}$$

$$\vec{u}_2 = \begin{pmatrix} R - 0.5 \\ R - 0.5 \\ R - 0.5 \end{pmatrix} \tag{B.6}$$

Here a positive z-value denotes a force clockwise around the z-axis of the fish. The vectors $\vec{v}$ and $\vec{u}$ are ultimately scaled to the coordinate system used in the simulation.

C. Estimation of the number of direction sensitive neurons

Neurons measured by Kubo et al. were distributed in four mirror-symmetrical pairs of clusters, which they termed anterior medial cluster (AMC), anterior lateral cluster (ALC), anterior ventral cluster (AVC) and posterior dorsal cluster (PDC). They recorded the coordinates of the cluster to be (in μm):

AMC (x=[-90, 90], y=[-140, -40], z=[-50, 50]), AVC (x=[-45, 45], y=[-120, -40], z=[-80, -20]), ALC (x1=[-130, -90], x2=[90, 130], y=[-140, 0], z=[-50, 0]), PDC (x=[-60, 60], y=[-220, -150] z=[-40, 40]), where x, y and z represent the left-right, rostral-caudal and dorsal-ventral axes, respectively.

We assume them to be approximately ellipsoid, thus we estimate

$$V_{AMC} \approx 0.6 \cdot 180\mu m \cdot 100\mu m \cdot 100\mu m = 1.080.000\mu m^3 \tag{C.1}$$

$$V_{AVC} \approx 0.6 \cdot 90\mu m \cdot 80\mu m \cdot 60\mu m = 259.200\mu m^3 \tag{C.2}$$

$$V_{ALC} \approx 0.6 \cdot 80\mu m \cdot 140\mu m \cdot 50\mu m = 336.000\mu m^3 \tag{C.3}$$

$$V_{PDC} \approx 0.6 \cdot 120\mu m \cdot 70\mu m \cdot 80\mu m = 403.200\mu m^3 \tag{C.4}$$

Then

$$V \approx 2 \cdot 10^6 \mu m^3 \tag{C.5}$$

We estimate the somata of the neurons to be $6\mu m$ wide and take up about 65% of the volume in the clusters. We assume the rest is made up by neuropil. As a rough estimate, the number of cells in the examined areas then is

$$n \approx \frac{V}{4/3\pi \cdot (3\mu m)^3} \cdot 0.65 \approx 11.500 \tag{C.6}$$

from which about 500 have been measured to be sensitive to horizontally moving stimuli by Kubo et al.

D. Quine-McCluskey algorithm

Table 1: Truth table for a neuron specific for forward translational motion; a 1 in the response column means the neuron is firing. On the right the corresponding subterms of the canonical disjunctive normal form are noted.

	NL	TL	TR	NR	Response	
NL/-	1	0	0	0	1	NL $\wedge\neg$ TL $\wedge\neg$ TR $\wedge\neg$ NR
TL/-	0	1	0	0	0	$\neg$ NL $\wedge$ TL $\wedge\neg$ TR $\wedge\neg$ NR
-/TR	0	0	1	0	0	$\neg$ NL $\wedge\neg$ TL $\wedge$ TR $\wedge\neg$ NR
-/NR	0	0	0	1	0	$\neg$ NL $\wedge\neg$ TL $\wedge\neg$ TR $\wedge$ NR
CCW	0	1	0	1	0	$\neg$ NL $\wedge$ TL $\wedge\neg$ TR $\wedge$ NR
CW	1	0	1	0	0	NL $\wedge\neg$ TL $\wedge$ TR $\wedge\neg$ NR
FW	1	0	0	1	1	NL $\wedge\neg$ TL $\wedge\neg$ TR $\wedge$ NR
BW	0	1	1	0	0	$\neg$ NL $\wedge$ TL $\wedge$ TR $\wedge\neg$ NR

The reactions of a neuron are seen as the outputs of a logic formula with the four basic monocular stimuli (NL,TL,NR,TR, see figure 6) as input. In table 1 an exemplary response type (FEL, see figure 8), which is specific for forward translational motion, is written down in the form of a truth table.

The response of the neuron $R_{FEL}(\mathrm{NL}, \mathrm{TL}, \mathrm{TR}, \mathrm{NR})$ can therefore be described in its canonical disjunctive normal form:

$$R_{FEL}(\mathrm{NL}, \mathrm{TL}, \mathrm{TR}, \mathrm{NR}) \Leftrightarrow (\mathrm{NL} \wedge \neg\mathrm{TL} \wedge \neg\mathrm{TR} \wedge \neg\mathrm{NR}) \vee (\mathrm{NL} \wedge \neg\mathrm{TL} \wedge \neg\mathrm{TR} \wedge \mathrm{NR}) \quad \text{(D.1)}$$

With the Quine-McCluskey algorithm this term can be reduced to its minimal form:

$$R_{FEL}(\mathrm{NL}, \mathrm{TL}, \mathrm{TR}, \mathrm{NR}) \Leftrightarrow \mathrm{NL} \wedge \neg\mathrm{TR} \quad \text{(D.2)}$$

This expression can be calculated with the use of two logical operators. The number of logical operators of the response types ranges from 0 to 19, where most of the responses need rather few operators (Figure 7C).

E. Supplementary figures

E.1. Sparse coding performance

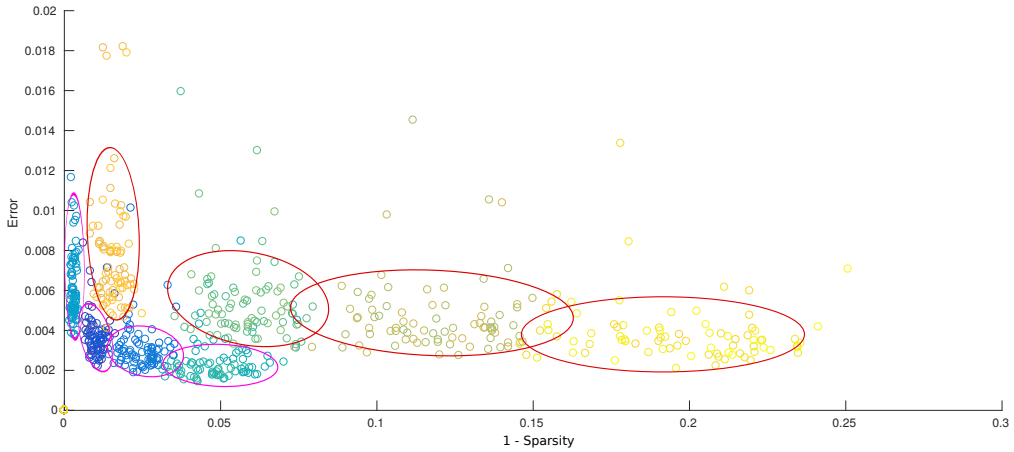

Figure 9: Error versus sparsity plot of the training data after learning. The sparsity quantifies how many neurons were involved in the reconstruction of the image. The error specifies the difference between original and reconstructed image. One circle approximately encloses the results of one hyperparameter setup of the algorithm on a smaller dataset with 6500 images. Dark red circles denote runs with $1/2$ times overcompleteness, magenta circles runs with 4 times overcompleteness. The sparsity factor λ decreases from left to right with values of 0.8, 0.4, 0.2 and 0.1 in both cases. Compare with Schultz et al. (2014). The sparse components seem to be over-fitting if they are not sparse enough. If they are generalizing better, the error starts to rise.

E.2. Simulation

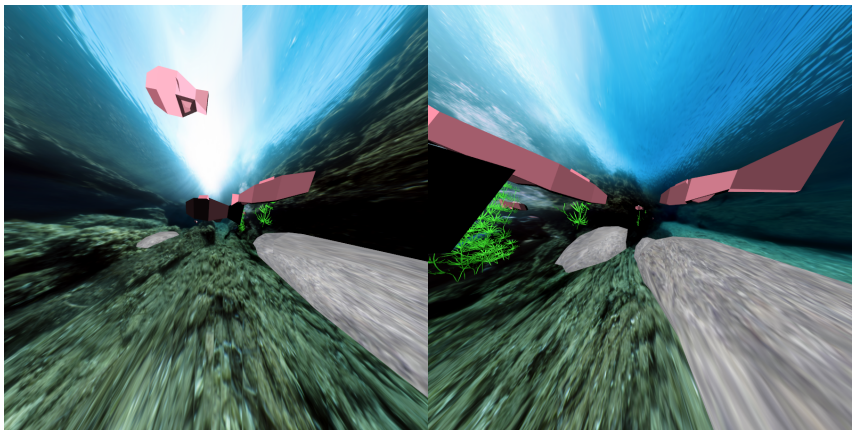

Figure 10: Image rendered in blender with two pinhole cameras and a field of view of 160°.

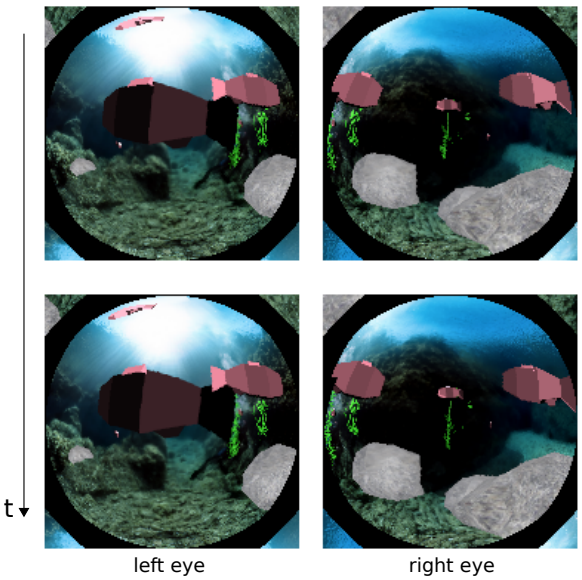

Figure 11: Image sequence created in the virtual reality simulation. This is the transformed image of figure 10. Time advances from top to bottom.

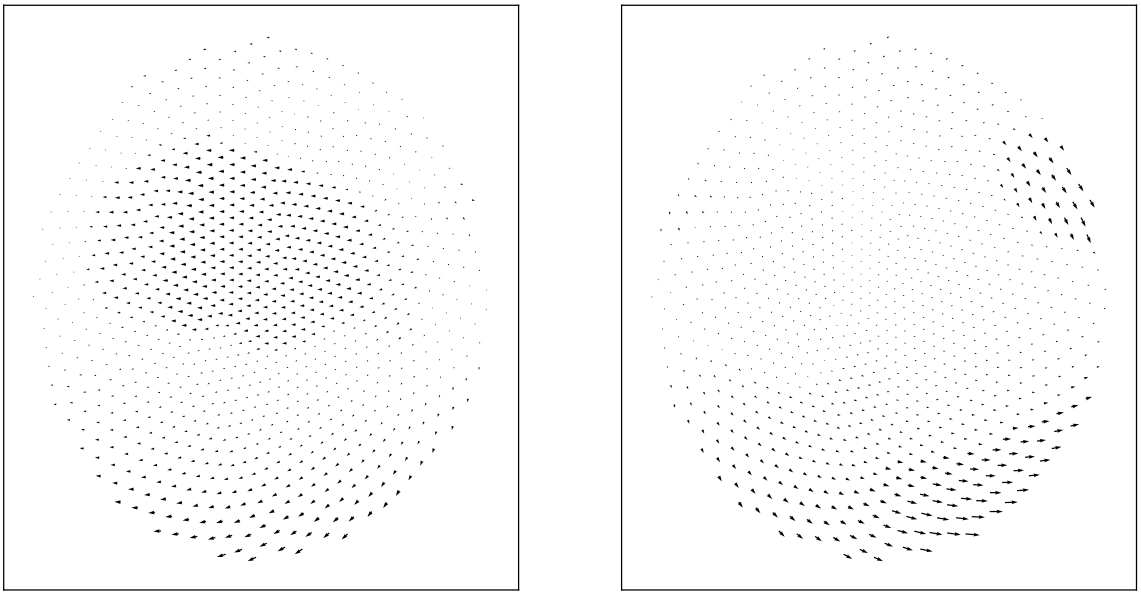

Figure 12: Optic flow elicited by the image sequence in figure 11. The fast moving floor and fish overshadow the optic flow resulting from slower motions. According to this optic flow field the motion of the fish is a forward translation close to the ground.

E.3. Testing

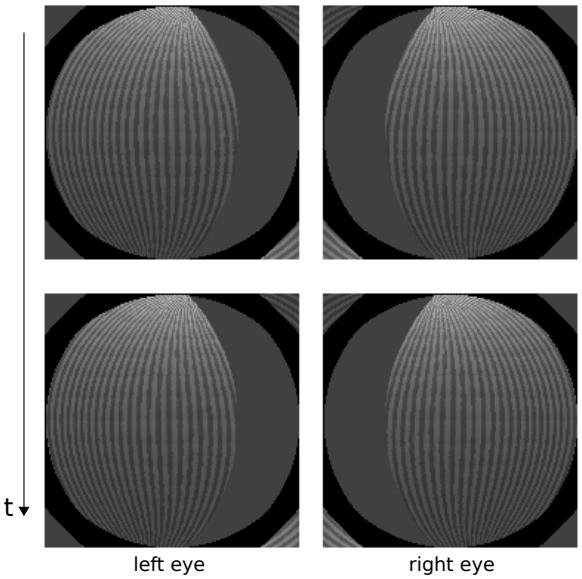

Figure 13: Image sequence created in the test environment. Depicted here is the backward translation (FW) condition.

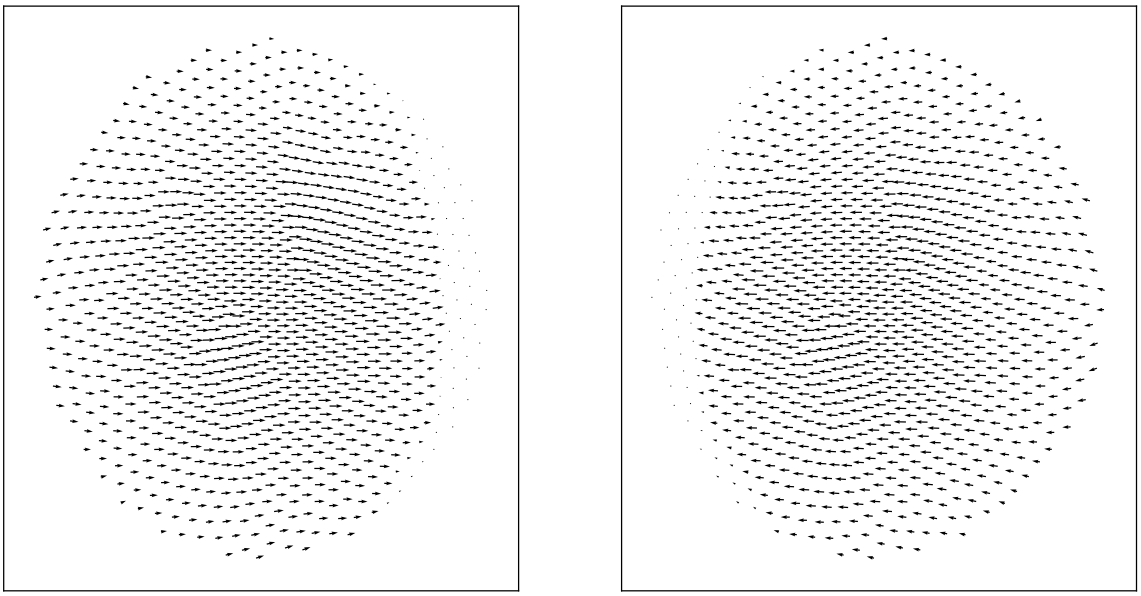

Figure 14: Optic flow in the backward translation condition (FW). The different lengths of the vectors are mainly an effect of the optic flow algorithm working better in the center of the images with stripes.

E.4. Results from Kubo et al.

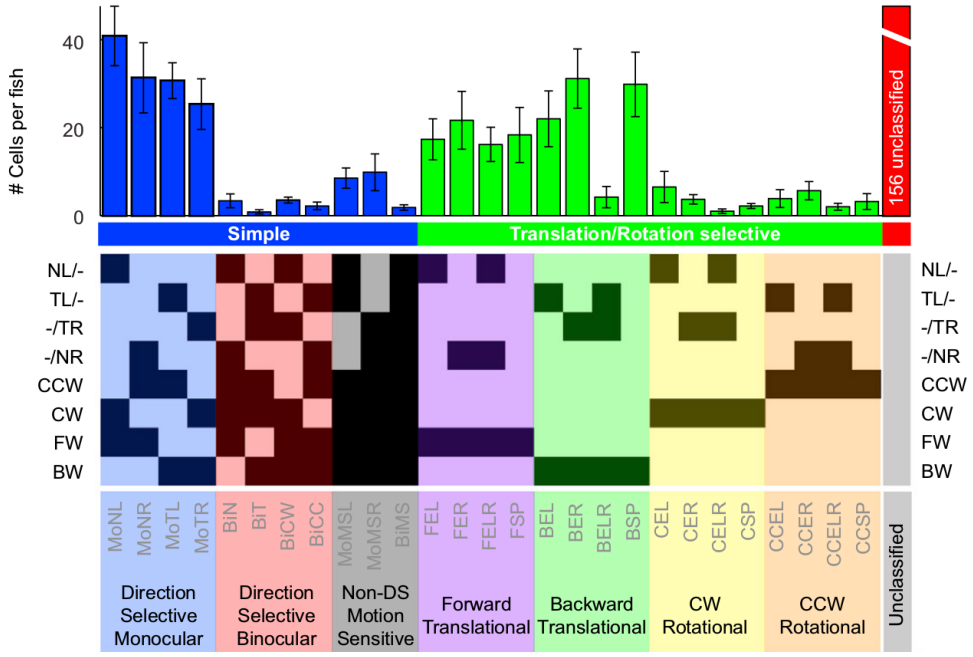

Figure 15: Figure 5 from Kubo et al. Please see there for more information, compare with figure 8.

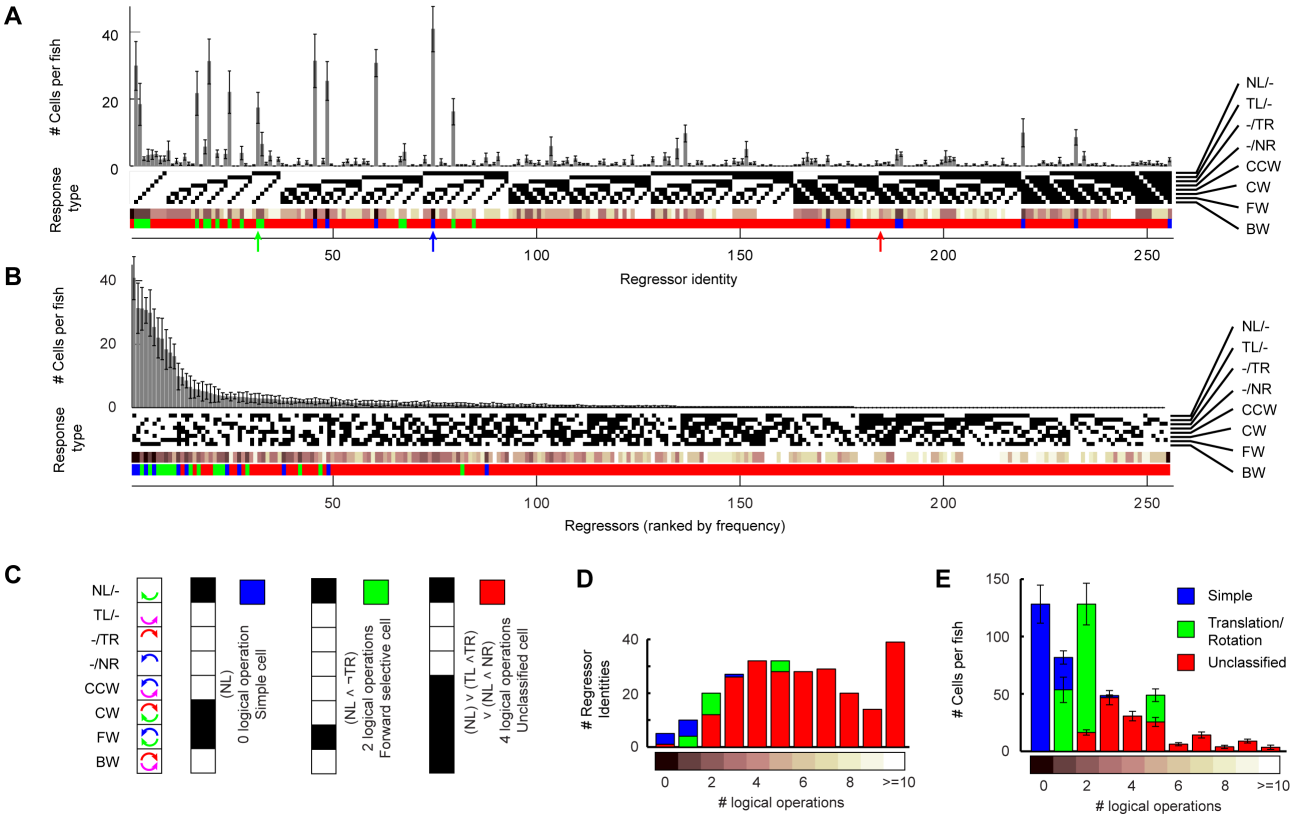

Figure 16: Figure 4 from Kubo et al. Please see there for more information, compare with figure 7.

E.5. Alternative results

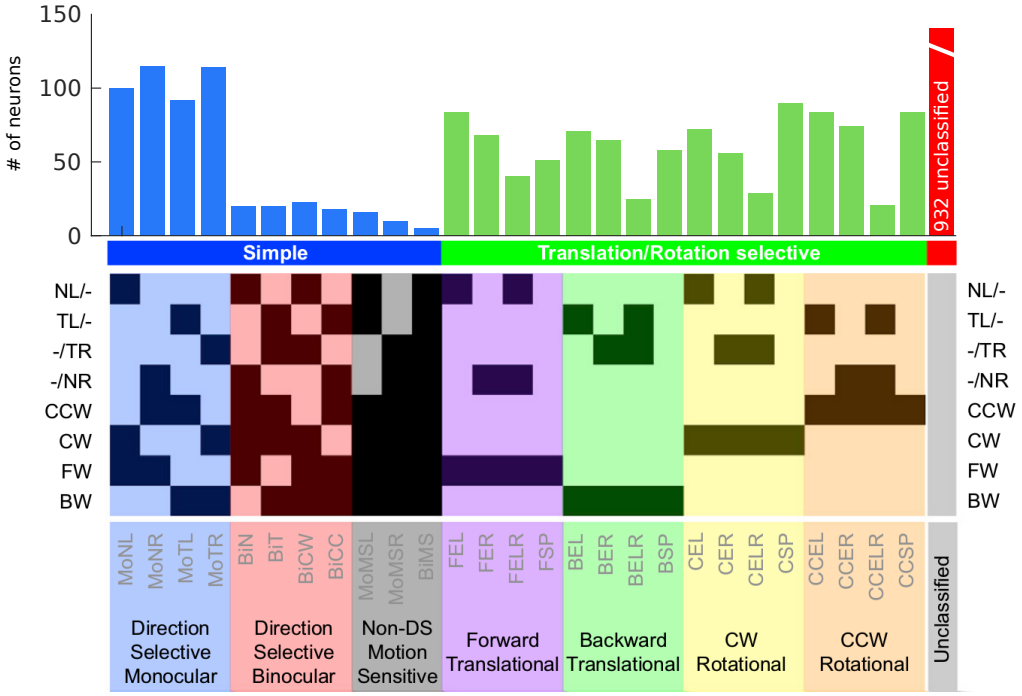

Figure 17: Alternative to figure 8 with 1/2 overcompleteness and $\lambda = 0.02$.

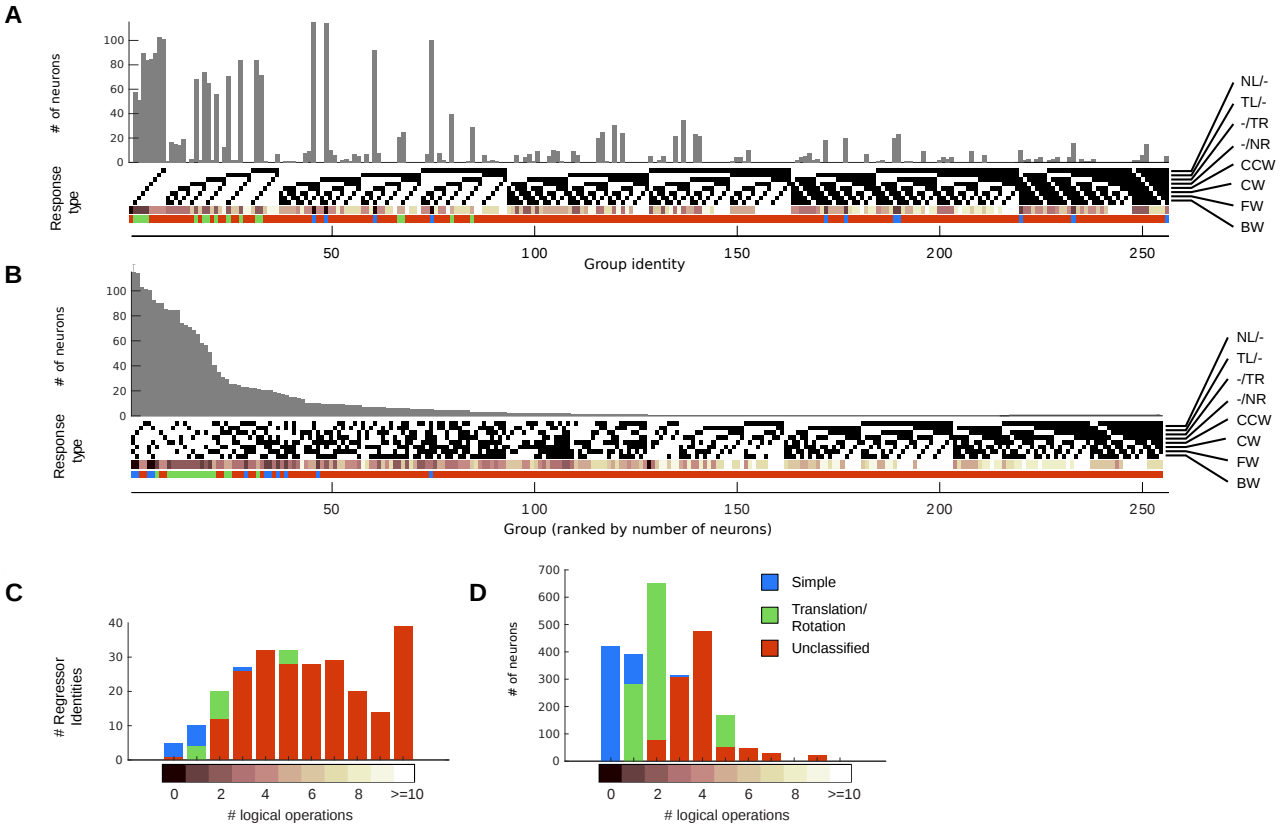

Figure 18: Alternative to figure 7 with 1/2 overcompleteness and $\lambda = 0.02$.

E.6. Resulting weights

Depicted here are some of the weights that have been calculated by the LCA with an overcompleteness of 2 and λ of 0.04. For more exemplary weights please refer to the supplementary disk.

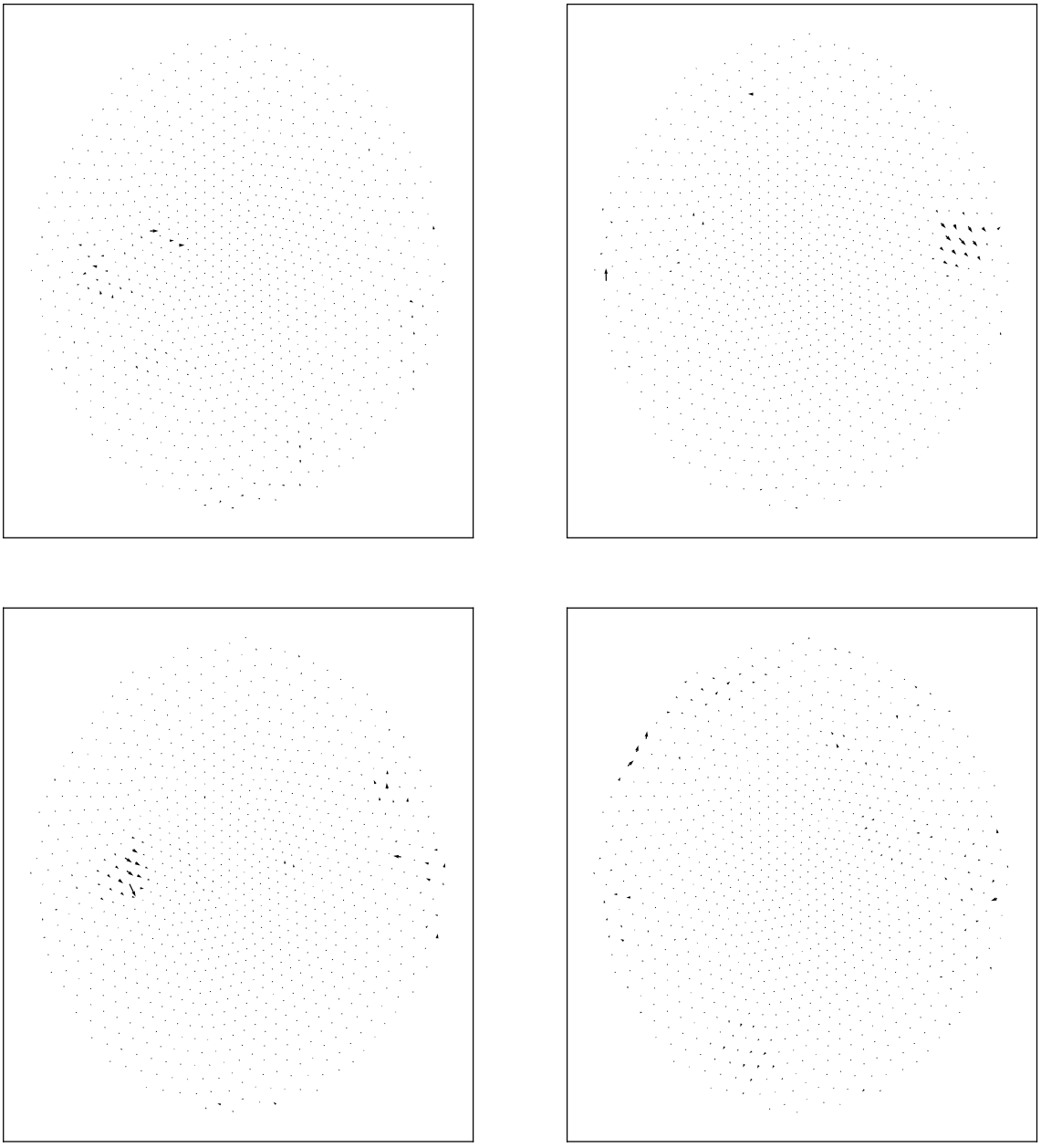

Figure 19: Examples showing monocular small localized optic flow components. These components are thought to be specialized for objects in the world, moving independently from the fish, though some larger vectors randomly occur outside the patches.

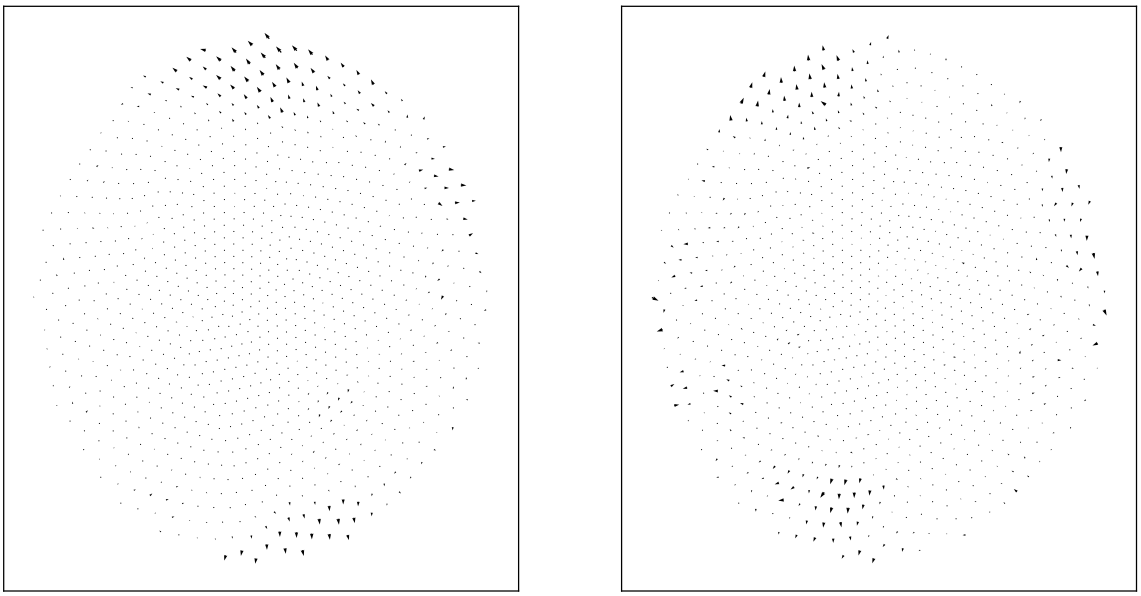

Figure 20: Optic flow component that could be specific for translational movement. The eyes of the fish are rotated nasalward, so this would be a component of a forward translation of the fish. We believe these components occur as a result of the simulation condition where top and bottom boundaries are flat. In an entirely natural habitat the components may be localized to only one side of the fish as here the existence of structures above and below the fish should not be correlated.

Figure 21: Component that could be specific for an upward pitch rotation of the fish. Again there are some vectors that do not quite fit the idealized optic flow (right).

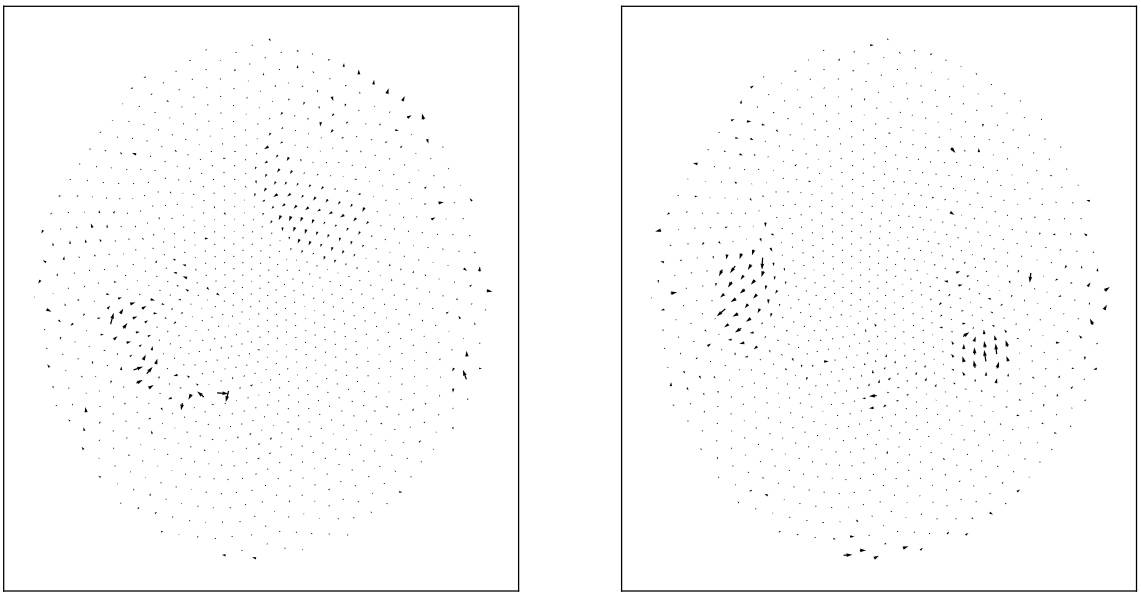

Figure 22: Combination of local, unidirectional patches of vectors with unknown specificity.

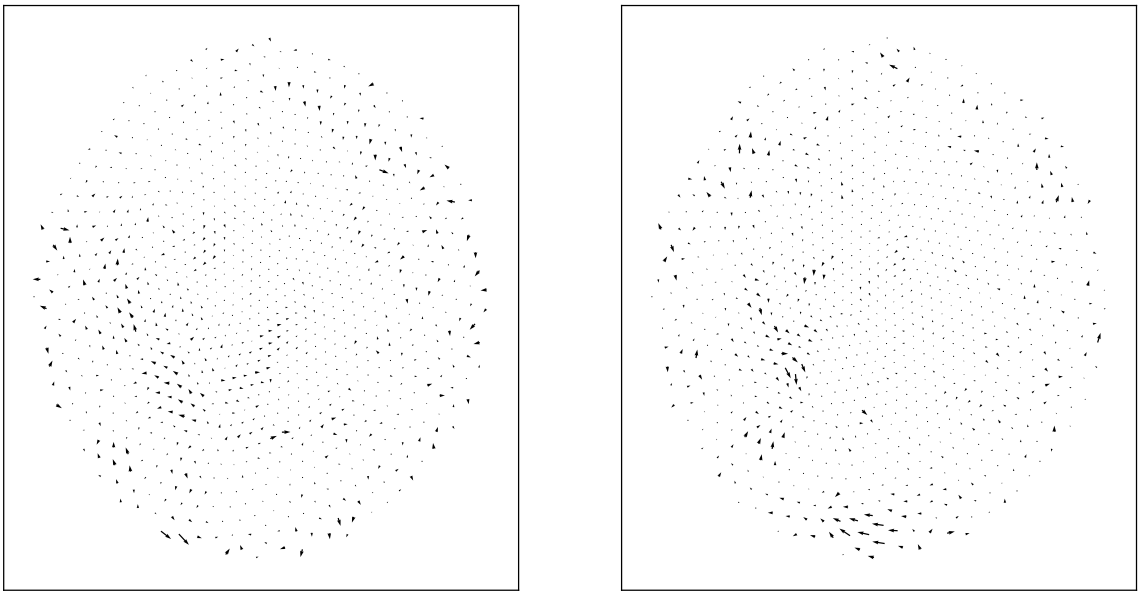

Figure 23: Typical example of a component that seems to be ill-formed.

Erklärung

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Datum: .....

(Unterschrift)