RESEARCH ARTICLE

Insect-Inspired Self-Motion Estimation with Dense Flow Fields—An Adaptive Matched Filter Approach

Simon Strübbe[1]*, **Wolfgang Stürzl**[2], **Martin Egelhaaf**[1]

1 Department of Neurobiology and CITEC, Bielefeld University, Bielefeld, Germany, **2** Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany

* simon.struebbe@uni-bielefeld.de

Citation: Strübbe S, Stürzl W, Egelhaaf M (2015) Insect-Inspired Self-Motion Estimation with Dense Flow Fields—An Adaptive Matched Filter Approach. PLoS ONE 10(8): e0128413. doi:10.1371/journal.pone.0128413

Editor: Holger G Krapp, Imperial College London, UNITED KINGDOM

Received: September 10, 2014

Accepted: April 28, 2015

Published: August 26, 2015

Data Availability Statement: All relevant data are available from the Publications Server of Bielefeld University (PUB) (http://pub.uni-bielefeld.de/data/2736885).

Funding: S.S. was financed from the budget of the department of neurobiology, Bielefeld University. W.S. was financed by the Deutsche Forschungsgemeinschaft (DFG) while working at Bielefeld University. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

The control of self-motion is a basic, but complex task for both technical and biological systems. Various algorithms have been proposed that allow the estimation of self-motion from the optic flow on the eyes. We show that two apparently very different approaches to solve this task, one technically and one biologically inspired, can be transformed into each other under certain conditions. One estimator of self-motion is based on a matched filter approach; it has been developed to describe the function of motion sensitive cells in the fly brain. The other estimator, the Koenderink and van Doorn (KvD) algorithm, was derived analytically with a technical background. If the distances to the objects in the environment can be assumed to be known, the two estimators are linear and equivalent, but are expressed in different mathematical forms. However, for most situations it is unrealistic to assume that the distances are known. Therefore, the depth structure of the environment needs to be determined in parallel to the self-motion parameters and leads to a non-linear problem. It is shown that the standard least mean square approach that is used by the KvD algorithm leads to a biased estimator. We derive a modification of this algorithm in order to remove the bias and demonstrate its improved performance by means of numerical simulations. For self-motion estimation it is beneficial to have a spherical visual field, similar to many flying insects. We show that in this case the representation of the depth structure of the environment derived from the optic flow can be simplified. Based on this result, we develop an adaptive matched filter approach for systems with a nearly spherical visual field. Then only eight parameters about the environment have to be memorized and updated during self-motion.

1 Introduction

Knowing one’s self-motion is crucial for navigation, course control and attitude stabilization. Although GPS can provide information about the position and thus about the self-motion of an agent, this information depends on the reliability of the contact to satellites. GPS is not

Competing Interests: The authors have declared that no competing interests exist.

available to animals which have to rely on other means to gain information about their position and self-motion. A direct method to measure self-motion for a walking artificial or biological agent is counting the steps or, in the case of a wheeled vehicle, to monitor the turns of the wheels. In contrast, most flying agents rely on their visual system to solve this task.

The visual system of an artificial or biological agent obtains information about self-motion from pixel shifts in the retinal image over time. These pixel shifts can be described by vectors, the optic flow vectors. The flow vectors depend on both the rotational and translational components of self-motion as well as on the viewing direction. Moreover, for the translational component it also depends on the distance to objects in the environment.

For small translations and rotations, the flow vector for viewing direction $\vec{d}_i$ is given by (see [1] for derivation)

$$\begin{aligned} \vec{p}_i(\vec{t}, \vec{r}, \mu_i) &= -\mu_i(\vec{t} - (\vec{t} \cdot \vec{d}_i)\vec{d}_i) - \vec{r} \times \vec{d}_i \\ &= -\mu_i(\vec{d}_i \times \vec{t} \times \vec{d}_i) - \vec{r} \times \vec{d}_i, \end{aligned} \tag{1}$$

where μ_i is the inverse distance ("nearness") to the object seen in direction $\vec{d}_i$, $\vec{t}$ is the translation vector, and $\vec{r}$ is the rotation vector (defining a rotation of angle $r = |\vec{r}|$ around the axis given by $\vec{r}/|\vec{r}|$). According to Eq (1), the flow vector $\vec{p}_i$ is perpendicular to the corresponding viewing direction $\vec{d}_i$. (We use 3d vectors to represent optic flow vectors. Otherwise one would have to define a tangential plane for every viewing direction.)

There are two principally different ways to use optic flow information for self-motion estimation. One way is to identify features in the retinal image at one point in time and find the same features at the next time point in order to compute their displacement. Several technical estimation methods for self-motion are based on these feature correspondences [2, 3]. They rely on a small number of corresponding image points and have to concern about outliers. Such estimation methods are widely used in the technical literature to determine the movement and/or the calibration of a camera [4]. When the self-motion steps are small or the frame rates are high an alternative way to extract self-motion information is possible. Instead of extracting features in the image the local pixel shifts on the retina, called optical flow, produced by the self-motion of the agent is determined through spatiotemporal intensity correspondences in the pattern. This can be done by a gradient-based detector like the Lucas-Kanade detector [5], which compares spatial and temporal derivatives, or by a biologically inspired detector, like the elementary movement detector of the correlation type [6, 7], which uses spatiotemporal auto-correlation signals.

Here we propose a new adaptive approach which combines the advantages of two methods for self-motion estimation based on optical flow, the matched filter approach (MFA) proposed by Franz and Krapp [8] and an algorithm proposed by Koenderink and van Doorn (KvD algorithm) [1]. The MFA estimates self-motion by using linear filters, so called matched filters. Matched filters have the structural form of the pattern they are meant to detect [9] and are the optimal detectors for patterns, which are disturbed by Gaussian errors. In this case the linear filters of the MFA resemble ideal flow fields. Franz and Krapp [8] introduced six filters of this type for the six self-motion components, three for translation and three for rotation. Each of these six filters was tuned to one of the flow fields generated by the six self-motion components, although in general the filters react also to flow generated by the other self-motion components. There is one exception: For a flow field which covers the whole viewing sphere and for isotropic distances, i.e. in the center of a sphere, Borst and Weber [10] showed that model neurons acting as such linear filters are not influenced by other flow fields. To eliminate the influence of other flow fields in the case of an arbitrary field of view Franz et al. [11] introduced a coupling matrix and used its inverse to uncouple the output of the model neurons.

The KvD algorithm is iterative and tries to determine not only the self-motion components but also the distances of the moving agent to objects and surfaces in the environment. These distances influence the translational optic flow and therefore the self-motion estimate. The KvD algorithm starts with a simple distance estimate and determines in the same iteration preliminary self-motion components. In the next iteration these preliminary self-motion components are used as the basis for determining a better distance estimate, which is then used for improving the motion estimate. By the MFA the distances are taken into account statistically and are integrated without further changes in the filters. Dahmen et al. [12] have shown that one iteration step of the KvD algorithm corresponds to the MFA of Franz and Krapp [8] by assuming that some terms in the KvD algorithm are negligibly small. As an important step for the the development of an adaptive approach we show that the MFA with a specific coupling matrix is fully equivalent to one iteration step of the KvD algorithm and not just an approximation. The Gauss-Markov-Theorem [13] gives an explanation of this equivalence. This theorem guarantees the existence of a unique optimal estimator for a linear estimation problem. Both mentioned methods find this optimal solution, although the two approaches seem to be totally different.

The MFA was proposed to explain the motion sensitivity structure of the tangential cells in the fly visual system [8, 14–23]. These cells are directionally selective for motion within their large receptive fields [16, 17, 24–26]. The spatial pattern of their motion sensitivity resembles flow fields on the retina generated by self-motion. Therefore, these cells were proposed to act as matched filters for self-motion estimation and to help the fly to solve visual orientation tasks [8]. However, since the MFA makes a priori assumptions about the 3D structure of the environment, self-motion estimation deteriorates in an environment with variable distance distribution.

It is known that the fly's nervous system can adapt to sensory inputs [27–32]. With this in mind, we propose a biologically inspired adaptive MFA, which adapts to the depth structure of the environment. This new model avoids the multiple iteration steps used by the KvD algorithm, on the one hand, and the hard-wired distance dependence of the MFA, on the other hand. The adaptive MFA extracts the depth structure from the optic flow field similar to the KvD algorithm. When the distances are not known the self-motion estimation problem becomes non-linear. Although the KvD algorithm is an optimal estimator in the linear case, it is, as we will show, a biased estimator in the non-linear case. The error in the quantities which are estimated does not converge to zero with increasing number of flow vectors. Therefore, we propose a modified version of the KvD algorithm. Numerical simulations indicate that the modified version has no bias.

On the basis of this modified KvD algorithm an adaptive MFA is developed that is inspired by a property of the visual system of insects: Insects have a field of view which nearly covers the whole sphere. It will be discussed that this property is beneficial for self-motion estimation and hence is desirable also for artificial agents which navigate by means of their visual system. The insect or agent should only adapt to the global properties of the depth structure and ignore irrelevant details. To achieve this, the inverse distances are expanded in a complete set of orthonormal functions, the spherical harmonics. It is desirable that the first-orders of this function set contribute most to the solution of the self-motion problem. We show that in the case of spherically distributed flow vectors all orders beyond the second-order of this function set do not contribute to self-motion estimation and can, thus, be neglected without losing information. Hence, if insects or artificial agents adapt to the depth structure they have to be sensitive only to low order depths functions, which are the dipole and quadrupole moments of the depth structure.

2 Results

A major objective of this study is to show that two well-established self-motion estimators are mathematically equivalent: The MFA equals one iteration step of the KvD algorithm when the inverse distances to objects in the environment are assumed to be known. To achieve this, we derive the MFA in an alternative way.

We then show that the KvD approach leads to an biased estimator in the general case when distances are unknown and have to be estimated together with the self-motion parameters. We present a modification of the KVD iteration equation that removes the bias and derive an adaptive MFA from this corrected version, which includes a simple but, with respect to self-motion estimation, complete depth model.

Before dealing with these topics, both the basic equations underlying the MFA approach and the KvD algorithm need to be introduced.

2.1 The matched filter approach

In the original MFA [8, 11, 33, 34] the depth structure of the environment is not determined from the current flow field but described statistically with a fixed distribution that is assumed to be known. The first statistical parameter considered in [11] is the average inverse distance $\bar{\mu}_i$, which is measured in every viewing direction over a number of learning flights in different environments. The variability of the distances is given by the covariance matrix C_μ. Secondly, the noise in the flow measurement is determined for each viewing direction n_i where a zero mean is assumed. The noise values are combined in the covariance matrix C_n. The third statistical parameter is the distribution of the translations t. It is assumed that the agent does not translate in every possible direction with the same probability. The corresponding statistical parameter is the covariance matrix C_t.

An optic flow vector $\vec{p}_i$ has only two degrees of freedom because it is the projection of object motion on the retina and thus orthogonal to the corresponding viewing direction $\vec{d}_i$. To consider only these degrees of freedom Franz et al. [11] introduce a local two-dimensional vector space for each viewing direction $\vec{d}_i$ which is orthogonal to the direction $\vec{d}_i$:

$$x_i = \vec{p}_i \cdot \vec{u}_i = \vec{p}_i^0 \cdot \vec{u}_i + n_{x,i} \quad (2)$$

$$y_i = \vec{p}_i \cdot \vec{v}_i = \vec{p}_i^0 \cdot \vec{v}_i + n_{y,i} \quad (3)$$

where $\vec{u}$ and $\vec{v}$ are the basis vectors of the new vector space. The values x_i and y_i represent the two degrees of freedom of $\vec{p}_i$. The measured vector $\vec{p}_i$ consists of the true optic flow vector $\vec{p}_i^0$ and an additive noise n_i.

In [11] the weights W for the matched filters which are multiplied with the optic flow components $\vec{x}$ (where $\vec{x}$ is a 2N dimensional vector containing all flow components x_i and y_i, $i = 1, 2, \ldots, N$) to estimate the six self-motion components $\vec{\theta}^{\mathrm{est}}$,

$$\vec{\theta}^{\mathrm{est}} = W \cdot \vec{x}, \quad (4)$$

are derived by a least-square principle:

$$e = E\left(\|\vec{\theta} - \vec{\theta}^{\mathrm{est}}\|^2\right), \quad (5)$$

where $\vec{\theta}$ are the true self-motion components. The weight matrix that minimizes the error e is:

$$W = (F^T C^{-1} F)^{-1} \cdot F^T \cdot C^{-1}. \tag{6}$$

The covariance matrix C combines the covariance matrices C_μ, C_n and C_t. The matrix F is given by

$$F = \begin{pmatrix} \frac{-\bar{\mu}_1 \vec{u}_1}{\vec{u}_1 \times \vec{d}_1}, & \frac{-\bar{\mu}_1 \vec{v}_1}{\vec{v}_1 \times \vec{d}_1}, & \frac{-\bar{\mu}_2 \vec{u}_2}{\vec{u}_2 \times \vec{d}_2}, & \frac{-\bar{\mu}_2 \vec{v}_2}{\vec{v}_2 \times \vec{d}_2}, & \dots \end{pmatrix}^T, \tag{7}$$

where $\bar{\mu}_i$ is the average or expected inverse distance for direction $\vec{d}_i$. The introduction of the matched filter approach is kept short because an alternative derivation of this approach is introduced in section 2.3, which is more suitable for the comparison with the KvD algorithm.

2.2 The Koenderink-van-Doorn (KvD) algorithm

As described by Koenderink and van Doorn [1], a straightforward approach for estimating the self-motion parameters is to find, in accordance with Eq (1), a translation vector $\vec{t}$, a rotation vector $\vec{r}$ and inverse distances $\{\mu_i\}_{i\,=\,1,\,2,\,\dots,\,N}$ that minimize the mean squared error between the theoretical optical flow vectors according to Eq (1), $\{\vec{p}_i(\vec{t}, \vec{r}, \mu_i)\}_{i=1,2,\dots,N}$ and the measured optical flow vectors $\{\vec{p}_i\}_{i=1,2,\dots,N}$:

$$e(\vec{t}, \vec{r}, \{\mu_i\}) = \frac{1}{N} \sum_{i=1}^{N} \|\vec{p}_i(\vec{t}, \vec{r}, \mu_i) - \vec{p}_i\|^2. \tag{8}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left\| -\mu_i \left(\vec{t} - \left(\vec{t} \cdot \vec{d}_i\right) \vec{d}_i\right) - \vec{r} \times \vec{d}_i - \vec{p}_i \right\|^2 \tag{9}$$

Since the optic flow vector (see Eq [1]) depends on the product of $\vec{t}$ and μ_i, the same flow vector is obtained by multiplying $\vec{t}$ and dividing μ_i by the same factor. Thus, an additional constraint is imposed to ensure convergence of the minimization procedure. The algorithm described in [1] uses the constraint $\|\vec{t}\| = 1$ and, starting from an initial guess, solves for the motion parameters by iterating the following equations derived from Eq (8) until convergence:

$$\mu_i = -\frac{\vec{t} \cdot \left(\vec{p}_i - \vec{d}_i \times \vec{r}\right)}{1 - \left(\vec{t} \cdot \vec{d}_i\right)^2}, \tag{10}$$

$$\vec{t} = -\xi \left\{ \langle \mu \vec{p} \rangle + \vec{r} \times \left\langle \mu \vec{d} \right\rangle - \left\langle \mu^2 \left(\vec{t} \cdot \vec{d}\right) \vec{d} \right\rangle \right\}, \tag{11}$$

$$\vec{r} = \left\langle \vec{p} \times \vec{d} \right\rangle + \vec{t} \times \left\langle \mu \vec{d} \right\rangle + \left\langle \left(\vec{r} \cdot \vec{d}\right) \vec{d} \right\rangle. \tag{12}$$

where ξ is a Lagrange multiplier ensuring the constraint $\|\vec{t}\| = 1$. The brackets $\langle\rangle$ stand for the average over all viewing directions, i.e. the summation over all directions $i = 1, 2, \dots, N$ divided by the number of directions N, e.g. $\langle \mu \vec{p} \rangle = \frac{1}{N} \sum_{i=1}^{N} \mu_i \vec{p}_i$.

2.3 Alternative derivation and properties of the coupling matrix in the MFA

To compare the two self-motion estimators one needs another mathematical form of the coupling matrix and the filters as given by the MFA (see 2.1). This can be achieved in two different ways. One can either transform the original Eqs (4), (6) and (7) of the MFA or derive the MFA in an alternative manner and show the equivalence to the original MFA afterward. Here, the second way is chosen. The matched filters will be derived according to the theory of optimal filters. The coupling of the estimated self-motion parameters will then be determined by inserting the filters in the flow Eq (1).

The optimal weights in the MFA depend on the statistics of the distances, of the noise and of the preferred translations. Here we assume that nothing is known about these distributions and thus consider the simplest case: The noise values are set to the same value independent of the viewing direction. We assume no preference for specific translation directions. The average inverse distances $\bar{\mu}_i$ are regarded as known.

The theory of optimal filters states that for a uniform Gaussian noise the best linear filter is a matched filter which has the same form as the pattern it has to detect [9]. Therefore, the templates for estimating translational flow fields have the form of a translational flow field.

$$\vec{T}^{t}_{ia} = -\mu_i\left(\vec{d}_i \times \vec{e}_a \times \vec{d}_i\right). \tag{13}$$

We have three translational templates one for each possible direction of translation represented by the three basis vectors $\vec{e}_a$ (a = 1,2,3).

Similarly, we have three rotational templates

$$\vec{T}^{r}_{ia} = -\vec{e}_a \times \vec{d}_i. \tag{14}$$

The three translational and three rotational templates $\vec{T}^{t}_{a}$ and $\vec{T}^{r}_{a}$ will be called “standard templates”.

The scalar product $\left\langle \vec{T}\cdot\vec{p} \right\rangle$ of a flow field $\vec{p}$ and a template $\vec{T}$, where the brackets stand for the mean over all viewing directions, can be interpreted as the output of a specific model neuron$a = \left\langle \vec{T}\cdot\vec{p} \right\rangle$. In general, the model neurons do not only react to the flow fields they are tuned to, but also to other flow fields. To solve this problem Franz et al. [11] introduced a matrix $(F^T C^{-1} F)^{-1}$ (see Eq (6)) which compensates for the coupling to other flow fields. The coupling between the self-motion estimates and therefore the coupling matrix are determined be inserting the templates $\vec{T}^{t}_{a}$ and $\vec{T}^{r}_{a}$ in the flow Eq (1). For this, the translation $\vec{t}$ and the rotation $\vec{r}$ must be separated into their components, $\vec{t} = t_1\vec{e}_1 + t_2\vec{e}_2 + t_3\vec{e}_3$ and $\vec{r} = r_1\vec{e}_1 + r_2\vec{e}_2 + r_3\vec{e}_3$, with the six self-motion parameters t_1, t_2, t_3 and r_1, r_2, r_3.

Since the cross product is linear $\left(\alpha\,\vec{a} + \beta\,\vec{b}\right) \times \vec{c} = \alpha(\vec{a} \times \vec{c}) + \beta\left(\vec{b} \times \vec{c}\right)$, the overall flow field is the sum of the six standard templates $\vec{T}_A$ (A = 1, 2, . . ., 6) weighted by the six self-motion components θ_A:

$$\begin{aligned} \vec{p}_i &= t_1\vec{T}^{t}_{i1} + t_2\vec{T}^{t}_{i2} + t_3\vec{T}^{t}_{i3} + r_1\vec{T}^{r}_{i1} + r_2\vec{T}^{r}_{i2} + r_3\vec{T}^{r}_{i3}\ , \\ &= \theta_1\vec{T}_{i1} + \theta_2\vec{T}_{i2} + \theta_3\vec{T}_{i3} + \theta_4\vec{T}_{i4} + \theta_5\vec{T}_{i5} + \theta_6\vec{T}_{i6}. \end{aligned} \tag{15}$$

Following our notation, the response of model neuron a_A with corresponding template $\vec{T}_A$ to the flow field $\vec{p}$ (Eq (15)) is

$$a_A = \left\langle \vec{T}_A \cdot \vec{p} \right\rangle = \left\langle \vec{T}_A \cdot \sum_B \vec{T}_B \theta_B \right\rangle = \sum \left\langle \vec{T}_A \cdot \vec{T}_B \right\rangle \theta_B = \sum_B \hat{M}_{AB} \theta_B. \tag{16}$$

Combining the responses of all six model neurons in one equation gives

$$\vec{a} = \hat{M} \cdot \vec{\theta}, \tag{17}$$

where the vectors $\vec{a}$ and $\vec{\theta}$ are considered as six dimensional vectors. Each entry of the 6 × 6 dimensional matrix $\hat{M}$, the coupling matrix, can be seen as the generalized scalar product of two of the six standard templates:

$$\hat{M}_{AB} = \left\langle \vec{T}_A \cdot \vec{T}_B \right\rangle \tag{18}$$

We can also write the coupling matrix as

$$M = \begin{pmatrix} M^{tt} & M^{tr} \\ M^{rt} & M^{rr} \end{pmatrix}, \tag{19}$$

where the indices t and r of the 3 × 3 sub-matrices indicate which templates are multiplied.

Using the inverse of the coupling matrix we can estimate the motion parameters $\vec{\theta}$ from the responses of the model neurons,

$$\vec{\theta}^{\text{est}} = \hat{M}^{-1}\vec{a} = \hat{M}^{-1} \begin{pmatrix} \left\langle \vec{T}^t_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_3 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_3 \cdot \vec{p} \right\rangle \end{pmatrix}. \tag{20}$$

The product $\left\langle \vec{T} \cdot \vec{p} \right\rangle$ and the multiplication with $\hat{M}^{-1}$ are linear transformations of the optical flow. One can therefore define new templates T' which include the linear transformation given by the matrix $\hat{M}^{-1}$.

Dahmen et al. [12] tested previously the self-motion estimation performance for different fields of view. The self-motion estimation performance even for error prone flow vectors is high, if the flow fields corresponding to different self-motion components differ essentially over the field of view. For a restricted field of view, for example a small region in front of the agent, upward translation cannot be distinguished from a pitch rotation of the agent. In this case the coupling matrix with constant distances becomes nearly singular and cannot be properly inverted.

In section 5.4 of the appendix it is shown by means of a coordinate transformation of two dimensional flow vectors (in tangent planes) into three dimensional ones (on the sphere) that Eq (20) is equivalent to Eq (4) with the weights given by Eq (6).

2.3.1 Properties of the coupling matrix. The equivalence between the MFA and one iteration of the KvD algorithm is shown in three steps. The first step was the alternative derivation of the MFA described above. The following second step is a simplification of the four submatrices of the coupling matrix.

Three cases have to be considered when calculating the entries of the matrix: the scalar product between two translational templates, the scalar product between two rotational templates and the scalar product between a translational and a rotational template.

The scalar product between two translational templates leads to the following expression:

$$\begin{aligned}(M^{tt})_{ab} &= \left\langle \vec{T}^{t}_{a} \cdot \vec{T}^{t}_{b} \right\rangle \\ &= \langle \mu^2 \rangle \vec{e}_a \cdot \vec{e}_b - \left\langle \mu^2 \left(\vec{e}_a \cdot \vec{d} \right) \cdot \left(\vec{e}_b \cdot \vec{d} \right) \right\rangle . \end{aligned} \tag{21}$$

The scalar product between two rotational templates results in

$$\begin{aligned}(M^{rr})_{ab} &= \left\langle \vec{T}^{r}_{a} \cdot \vec{T}^{r}_{b} \right\rangle \\ &= \vec{e}_a \cdot \vec{e}_b - \left\langle \left(\vec{e}_a \cdot \vec{d} \right) \cdot \left(\vec{e}_b \cdot \vec{d} \right) \right\rangle . \end{aligned} \tag{22}$$

Finally, the scalar product between a translational and a rotational template leads to

$$\begin{aligned}(M^{tr})_{ab} &= \quad (-M^{rt})_{ab} = \left\langle \vec{T}^{t}_{a} \cdot \vec{T}^{r}_{b} \right\rangle \\ &= \quad (\vec{e}_b \times \vec{e}_a) \cdot \left\langle \mu \vec{d} \right\rangle . \end{aligned} \tag{23}$$

Eq (21) informs us that the estimates of the three translation parameters t_1, t_2 and t_3 are coupled unless the term $\left\langle \left(\vec{e}_a \cdot \vec{d} \right) \cdot \left(\vec{e}_b \cdot \vec{d} \right) \right\rangle$ in M^{tt} is proportional to the identity matrix. The same holds for the coupling between the three rotation parameters r_1, r_2 and r_3 described by M^{rr}, see Eq (22). Similarly, the term $\left\langle \mu \vec{d} \right\rangle$ in M^{tr}, Eq (23), must be zero for the translation and rotation estimates to be uncoupled.

2.3.2 The case of constant distances and a spherical field of view. Borst and Weber [10] showed that for viewing directions homogeneously covering the whole sphere, and for identical distances (μ_i = constant for all i) the model neurons respond only to the components of the flow field they are tuned to. This result can be easily verified within the conceptual framework provided here by replacing the sums in Eqs (21), (22) and (23) with integrals over the unit sphere and by introducing spherical coordinates. The direction vectors $\vec{d}_i$ are then replaced by the vectors $\vec{d}_{\vartheta\varphi}$ that depend on the elevation angle ϑ and azimuth angle φ. In appendix 5.3 it is shown that the direction vectors $\vec{d}_{\vartheta\varphi}$ in the spherical coordinate system have the same form as the three real-valued dipole functions of the spherical harmonics. Due to the orthogonality of the spherical harmonics the integral $\int \left(\vec{e}_a \cdot \vec{d}_{\vartheta\varphi} \right) \cdot \left(\vec{e}_b \cdot \vec{d}_{\vartheta\varphi} \right) \sin \vartheta d\vartheta d\varphi$ becomes zero in the case of spherically distributed flow vectors, if $\vec{e}_a$ and $\vec{e}_b$ denote different basis vectors. The scalar product between a translational and a rotational template Eq (23) leads to the integral $\int (\vec{e}_b \times \vec{e}_a) \cdot \vec{d}_{\vartheta\varphi} \sin \vartheta d\vartheta d\varphi$ which can be regarded as the product between a first-order dipole function and the zeroth order spherical harmonic function (which is a constant). Due to the orthogonality of the spherical harmonics this integral is zero.

2.4 The relationship between the MFA and the KvD algorithm

In a final step we will show that the MFA with the coupling matrix is equivalent to one iteration of the KvD algorithm [1] for known distances. Hence, the two approaches do not represent principally different methods, but rather one method with two different ways of dealing with the depth structure. An additional way to take the depth structure into account is given in section 2.6 where an adaptive MFA is proposed.

Dahmen et al. [12] have already demonstrated that a MFA can be derived from the KvD algorithm, if certain terms are small and can be neglected. We will show that these terms are identical to entries of the coupling matrix, which was derived in the previous chapter.

The equivalence of the MFA and one iteration step of the KvD algorithm. Not only in the MFA, but also in the KvD approach coupling terms can be identified. In the translation Eq (11), the second term on the right side comprises the rotation. The rotation Eq (12) in turn contains a translational term. These terms were called "apparent rotation" and "apparent translation" by Dahmen et al. [12], and they couple translation and rotation. The third terms in both equations couple different components of the translation or the rotation whenever $\left\langle \mu^2 \vec{d} \otimes \vec{d} \right\rangle$ in $\left\langle \mu^2 \left(\vec{t} \cdot \vec{d}\right) \vec{d} \right\rangle = \left\langle \mu^2 \vec{d} \otimes \vec{d} \right\rangle \cdot \vec{t}$ or $\left\langle \vec{d} \otimes \vec{d} \right\rangle$ in $\left\langle \left(\vec{r} \cdot \vec{d}\right) \vec{d} \right\rangle = \left\langle \vec{d} \otimes \vec{d} \right\rangle \cdot \vec{r}$ contain off-diagonal components. The following derivation will show that these coupling terms are equivalent to the terms of the coupling matrix in the MFA.

The derivation starts with the MFA including the coupling matrix. From Eq (17) one obtains together with the equations for the coupling matrix Eqs (21), (22) and (23):

$$\vec{a} \; = \; \hat{M} \cdot \vec{\theta} = \begin{pmatrix} M^{tt} & M^{tr} \\ M^{rt} & M^{rr} \end{pmatrix} \cdot \begin{pmatrix} \vec{t} \\ \vec{r} \end{pmatrix} \; = \; \begin{pmatrix} \left\langle \vec{T}^t_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_3 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_3 \cdot \vec{p} \right\rangle \end{pmatrix} \tag{24}$$

$$(M^{tt})_{ab} = \langle \mu^2 \rangle \vec{e}_a \cdot \vec{e}_b - \left\langle \mu^2 \left(\vec{e}_a \cdot \vec{d}\right) \cdot \left(\vec{e}_b \cdot \vec{d}\right) \right\rangle \tag{25}$$

$$(M^{tr})_{ab} = \left\langle \mu (\vec{e}_a \times \vec{e}_b) \cdot \vec{d} \right\rangle \tag{26}$$

$$(M^{rt})_{ab} = -\left\langle \mu (\vec{e}_a \times \vec{e}_b) \cdot \vec{d} \right\rangle \tag{27}$$

$$(M^{rr})_{ab} = \vec{e}_a \cdot \vec{e}_b - \left\langle \left(\vec{e}_a \cdot \vec{d}\right) \cdot \left(\vec{e}_b \cdot \vec{d}\right) \right\rangle \tag{28}$$

On the left side of Eq (24), the vectors $\vec{t}$ and $\vec{r}$ are multiplied with the entries of the coupling matrix. On the right side the optical flow vectors $\vec{p}$ are multiplied with the templates. After some algebraic simplifications, which are given in appendix 5.1, and a rearrangement of the terms one obtains the known equations for translation Eq (11) and rotation Eq (12) of the KvD algorithm:

$$\vec{t} = -\frac{1}{\langle \mu^2 \rangle} \left\{ \langle \mu \vec{p} \rangle + \vec{r} \times \left\langle \mu \vec{d} \right\rangle - \left\langle \mu^2 \left(\vec{t} \cdot \vec{d} \right) \vec{d} \right\rangle \right\},$$

$$\vec{r} = \left\langle \vec{p} \times \vec{d} \right\rangle + \vec{t} \times \left\langle \mu \vec{d} \right\rangle + \left\langle \left(\vec{r} \cdot \vec{d} \right) \vec{d} \right\rangle,$$

where $\frac{1}{\langle \mu^2 \rangle}$ represents the Lagrange multiplier ξ.

Hence, the MFA and the KvD algorithm are identical for one iteration step.

2.5 The bias of the KvD algorithm

The equivalence of the KvD algorithm and the MFA for known distances follows directly from the Gauss-Markov theorem [13], which states that an ordinary least-squares estimator is the best unbiased estimator for an estimation problem which is linear and has uncorrelated errors with equal variances. Both methods start with such a least-squares approach. The MFA minimizes the quadratic error between the six true self-motion values and the estimated values as can be seen in Eq (5), whereas the KvD algorithm minimizes the quadratic error between the measured optic flow and the theoretical optic flow as can be seen in Eq (8). For known distances, the optic flow and the self-motion values are connected through a linear transformation. Thus, the two least-square approaches lead to the same self-motion estimator. This estimator is the unique optimal estimator as stated by the Gauss-Markov theorem.

The situation is different if the distances are not known and have to be estimated by the estimator together with the self-motion values. Then the problem is no longer linear and the Gauss-Markov theorem does not hold. Nonetheless, it seems likely that a least-square approach like in Eq (8) leads to an optimal estimator in the sense that the error in the estimated self-motion components approaches zero with increasing number of measured optical flow values. However, as will be shown in this section, the KvD algorithm in general does not converge to the true self-motion values for an infinite number of flow vectors. It is still an open question from which minimization principle an optimal estimator can be derived. The increasing number of flow vectors raises the problem that for each flow vector which gives two additional error-prone values one additional value has to be estimated: the inverse distance in the respective viewing direction. Although the number of measured values increases towards infinity, the ratio between the number of estimated and measured values does not decrease to zero. Hence, even for an infinite number of flow vectors the estimated inverse distances are still afflicted with errors. However, it should still be possible to correctly estimate the fixed number of self-motion values for an infinite number of flow vectors. In section 2.5.2 a modified KvD algorithm will be derived. The modification is tested numerically under two conditions where the original KvD algorithm turns out to be biased (section 2.5.3).

2.5.1 The non-vanishing error term. The KvD algorithm is an unbiased estimator only under certain conditions. To show this, the propagation of the error in the flow vectors $\vec{p}_i$ over the iterations will be analyzed. We model the measured flow vectors $\vec{p}_i$ as the sum of the true vector $\vec{p}_i^0$ and a random error vector $\Delta\vec{p}_i$, $\vec{p}_i = \vec{p}_i^0 + \Delta\vec{p}_i$. Similar to vector $\vec{p}_i$, the vectors $\vec{p}_i^0$ and $\Delta\vec{p}_i$ have only two degrees of freedom. It will be assumed that the random vectors $\Delta\vec{p}_i$ are unbiased, i.e. the expectation values for all directions i are zero, $E(\Delta\vec{p}_i) = 0$.

Two special conditions will be considered in the following:

1. The viewing directions $\vec{d}_i$ are equally distributed over the sphere.

2. The random vectors $\Delta\vec{p}_i$ are uncorrelated and their variances constant, independent of the directions i, $\mathrm{var}(\Delta\vec{p}_i) = E(\Delta\vec{p}_i \cdot \Delta\vec{p}_i) = \mathrm{constant}$.

The Gauss-Markov theorem assumes condition (2) to be fulfilled. We will also consider deviations from this condition, because the KvD algorithm and its modified version that will be derived in section 2.5.2 behave differently then. Condition (2) is violated if, for example, the error of the optic flow measurement depends on the length of the measured optic flow vectors.

In subsection 5.2 of the appendix it is shown that, in general, the translation estimated by the KvD algorithm contains errors that do not vanish even for an increasing number of flow vectors. There are two error terms which are additive to the real translation $\vec{t}^0$.

$$\vec{t} = \vec{t}^0 + \Delta\vec{t} = \vec{t}^0 + a + b$$

$$a \propto \left\langle \frac{\Delta\vec{p} \otimes \Delta\vec{p}}{1 - \left(\vec{t}^0 \cdot \vec{d}\right)^2} \right\rangle_\infty \cdot \vec{t}^0$$

$$b \propto \left\langle \frac{\vec{t}^0 \cdot (\Delta\vec{p} \otimes \Delta\vec{p}) \cdot \vec{t}^0}{(1 - (\vec{t}^0 \cdot \vec{d})^2)^2} (\vec{d} \otimes \vec{d}) \right\rangle_\infty \cdot \vec{t}^0$$

The index ∞ of the brackets $\langle\rangle$ stands for the limit of an infinite number of flow vectors. The discrete direction vectors $\vec{d}_i$ can be exchanged by their continuous counterparts $\vec{d}_{\vartheta\varphi}$, and the sum over i can be replaced by an integral over the field of view.

The iteration equation for $\vec{r}$, Eq (12), does not contain terms that lead to a bias in the estimated motion values. However, the estimated rotation will be affected indirectly by errors in the estimated translation.

First the integrals containing the denominators $D_1 = 1 - \left(\vec{t}^0 \cdot \vec{d}\right)^2$ and $D_2 = \left(1 - \left(\vec{t}^0 \cdot \vec{d}\right)^2\right)^2$ are analyzed, disregarding the numerators. The integrals over $\frac{1}{D_1}$ and $\frac{1}{D_2}$ are zero only if the first condition of equally distributed flow vectors is fulfilled. To avoid the singularity for $\vec{d} = \vec{t}^0$ a small constant ε was added to the denominators.

If conditions (1) and (2) are fulfilled, the terms a and b are zero and the KvD algorithm is an unbiased estimator.

If condition (1) holds but condition (2) does not as, for instance, in the case of realistic EMDs or gradient-based detectors, we have to integrate over a direction dependent function resulting from the direction dependent flow errors $\Delta\vec{p}$. Hence the terms a and b converge to finite values.

The error terms a and b do not play a role if they are proportional to the identity matrix, because of the rescaling of the translation vector, which ensures $\|\vec{t}\| = 1$. The matrices $E_1 = \langle\Delta\vec{p} \otimes \Delta\vec{p}\rangle$ and $E_2 = \left\langle(\Delta\vec{p} \otimes \Delta\vec{p}) \cdot (\vec{d} \otimes \vec{d})\right\rangle$ are proportional to the unit matrix, if and only if both conditions (1) and (2) are satisfied. This can be shown by taking into account the symmetry of viewing directions and the constant variances of the flow errors.

Most interestingly, if condition (2) is fulfilled (a pre-condition of the Gauss-Markov theorem) but condition (1) is not, the terms a and b converge to finite values. In this case the integrals over the denominators D_1, D_2 and the integrals over the numerators E_1, E_2 have finite values. This means that the ordinary least-squares approach from Eq (8) leads to a biased self-motion estimator.

2.5.2 Modification of the KvD iteration equations. To improve the KvD algorithm a modified version of the iteration Eq (11) will be derived. The flow Eq (1) can be transformed:

$$\vec{p}_i = -\mu_i \left(\vec{t} - \left(\vec{t} \cdot \vec{d}_i \right) \vec{d}_i \right) - \vec{r} \times \vec{d}_i \tag{29}$$

$$\mu_i \vec{t} = -\vec{p}_i - \vec{r} \times \vec{d}_i + \mu_i \left(\vec{t} \cdot \vec{d}_i \right) \vec{d}_i \quad . \tag{30}$$

By taking the average and solving for $\vec{t}$ we obtain

$$\langle \mu \vec{t} \rangle = -\langle \vec{p} \rangle - \left\langle \vec{r} \times \vec{d} \right\rangle + \left\langle \mu \left(\vec{t} \cdot \vec{d} \right) \vec{d} \right\rangle,$$

$$\vec{t} = \langle \mu \rangle^{-1} \left\{ -\langle \vec{p} \rangle - \left\langle \vec{r} \times \vec{d} \right\rangle + \left\langle \mu \left(\vec{t} \cdot \vec{d} \right) \vec{d} \right\rangle \right\},$$

$$\vec{t} = -\xi \left\{ \langle \vec{p} \rangle + \vec{r} \times \left\langle \vec{d} \right\rangle - \left\langle \mu \left(\vec{t} \cdot \vec{d} \right) \vec{d} \right\rangle \right\}, \tag{31}$$

where ξ ensures that $\vec{t}$ is normalized. Compared with the original equation for the translation Eq (11) the additional factor μ_i is absent. Nonetheless, the above equation for the translation depends still on the distances. An analog derivation leads to the same iteration equation for the rotation as in the original KvD algorithm.

If the flow vectors have no errors the modified version converges, as does the original algorithm, to the true self-motion parameters. In contrast to the result for the original KvD algorithm in appendix 5.2, the iteration equations of the modified version do not contain $\mu\vec{p}$ or μ^2 and the true self-motion values are fix points of the iteration (only when the true values are fix points in the iteration the algorithm can converge to these values).

2.5.3 Numerical tests of the original and modified KvD algorithm. In Fig 1 the bias of the KvD algorithm is shown numerically (see section 'Material and Methods', 4.3 for a detailed description of the numerical test). The left part of the figure shows simulation results for flow vectors with added errors of equal variance. The field of view given by the viewing directions $\vec{d}_i$ is non-equally distributed: The flow vectors are equally distributed except for two regions of the sphere which do not contain any flow vectors. The two regions are quarters of the half-sphere which lie opposite to each other in the upper half-sphere. Thus, the simulation result provides an example where condition (2) of section 2.5.1 is fulfilled but condition (1) is not. The translation error of the original KvD algorithm is significantly larger and increasingly deviates from that of the modified version with increasing number of flow vectors. Due to the coupling of the iteration equations, the error of the rotation, in case of the original KvD algorithm, is affected by the translation error and, thus, also deviates from that of the modified version.

The right part of the figure shows results where the standard deviation of $\triangle\vec{p}_i$ is proportional to the length of the local flow vector $\vec{p}_i$. This time, the viewing directions cover the whole sphere homogeneously and, thus, condition (1) of section 2.5.1 holds while condition (2) does not. Again, the original KvD algorithm is biased (see also section 2.5.1). However, the error of the rotation in the original KvD algorithm is not influenced by the translation error as a consequence of the spherically distributed viewing directions.

The error of the modified KvD algorithm is in both analyzed cases inversely proportional to the square root of the number of flow vectors (see black line in Fig 1). The modified KvD algorithm shows therefore an error behavior as is characteristic of an unbiased linear estimator.

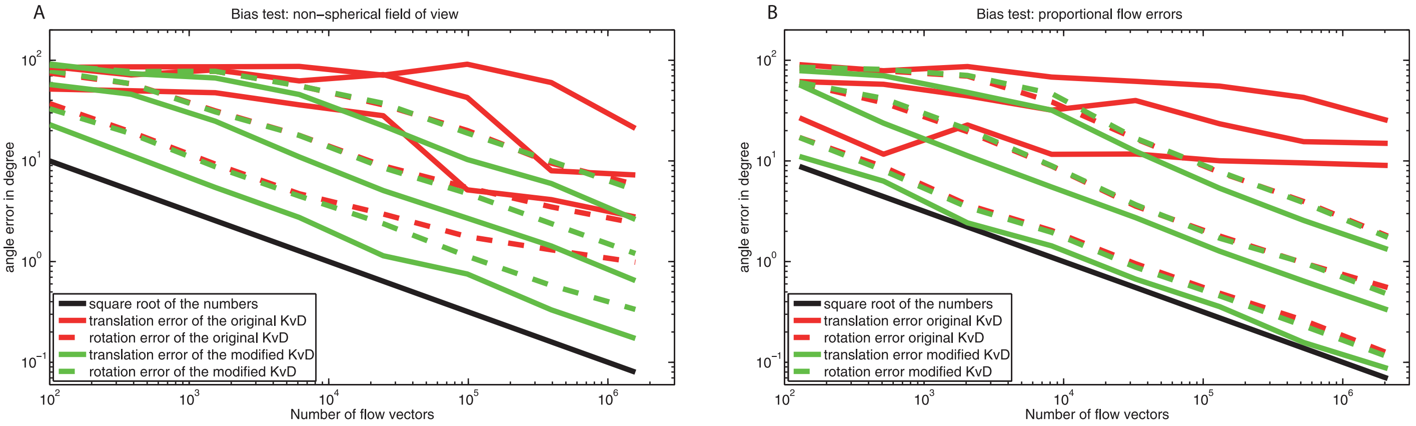

Fig 1. The averaged angle error, *arccos*$\left(\frac{\vec{\theta}_N^{\mathbf{est}}\cdot\vec{\theta}}{|\vec{\theta}_N^{\mathbf{est}}|\cdot|\vec{\theta}|}\right)$, between the estimation $\vec{\theta}_N^{\mathbf{est}}$, which depends on the number of flow vectors *N*, and the true values $\vec{\theta}$, is shown for translation (solid lines) and rotation (dashed lines). Red curves show the errors for original KvD algorithm, green curves for the modified KvD algorithm. The errors are averaged over 40 trials (see methods 4.3). For each trial the true self-motion parameters are chosen randomly, equally distributed over the sphere, in such a way, that the resulting translational flow equals the resulting rotational flow in magnitude. The distances, also determined randomly, lie with equal probability between one and three in arbitrary units. The results for three different variances $(\triangle\vec{p})^2$ are shown (from bottom to top): 1, 3, and 9 times the flow vector length, where the factor is interpreted differently in the two graphs. **A)** Results for non-equally distributed flow vectors with equal variance of the flow vector errors $\triangle\vec{p}_i$ (the variance is matched to the mean flow vector). **B)** Results for equally distributed flow vectors, where the variance of the errors $\triangle\vec{p}_i$ depends linearly on the length of the $\vec{p}_i$ (the variance is matched to the local flow vector).

doi:10.1371/journal.pone.0128413.g001

2.6 An adaptive MFA

The approach of Franz and Krapp [8] is based on the assumption that the statistics of the depth structure of the environment is fixed as well as the preferred translation directions of the agent are known. For the simplest statistical model, which assumes that the distance variability, the noise in the flow measurements and the preferred translation of the agent are independent from the viewing direction, the dedicated covariance matrices are proportional to the identity matrix. Most important, one has to specify the depth structure of the environment by defining the average inverse distances $\langle\mu\rangle_i$ or, as in [8], the average distances $\langle D\rangle_i$. Franz and Krapp [8] modeled the distances $\langle D\rangle_i$ by

$$\langle D\rangle_i = \begin{cases} D_0 & \varepsilon_i \geq 0\ , \\ \dfrac{\beta D_0}{\sqrt{1+(\beta^2-1)\cos^2(\varepsilon_i)}} & \varepsilon_i < 0, \end{cases} \tag{32}$$

where D_0 denotes a typical distance in the upper hemisphere, ε_i the elevation angle of viewing direction i and $\beta = \frac{h}{D_0} < 1$ the ratio of the average flight altitude h and D_0. This model takes into account that the distances are usually smaller for viewing directions below the horizon.

It is obvious that the performance of self-motion estimation is poor when the fixed depth model Eq (39) is not a good description of the depth structure at the current position of the agent. Therefore, we propose an adaptive MFA, which allows the agent to adapt its depth model to the current environment. This is only possible, if we can assume that the depth structure properties of the environment do not change abruptly from one time step to the next. A time constant describes the intervals in which the depth model will be updated. Between updates the depth model remains fixed and has to be memorized by the agent. The depth information is obtained from the optic flow, as well as the self-motion values. Since the original KvD algorithm is biased in this case, we use our modified KvD version as the starting point for deriving the adaptive model.

As will be shown in the following, it is not necessary to represent the full depth information, because it is sufficient to memorize just eight distance dependent parameters. Three parameters are contained in the distance-dependent term of the rotation Eq (12), $\vec{t} \times \left\langle \mu\vec{d} \right\rangle$ or $(t_1, t_2, t_3)^T \times (\mu \sin(\vartheta)\cos(\varphi), \mu \sin(\vartheta)\sin(\varphi), \mu \cos(\vartheta))^T$. In the translation equation of the modified KvD algorithm Eq (31) only $\left\langle \mu\vec{d} \otimes \vec{d} \right\rangle \cdot \vec{t}$ depends on μ. As the matrix $\left\langle \mu\vec{d} \otimes \vec{d} \right\rangle$ is symmetric it contains only six different μ-dependent elements.

The first three orders of spherical harmonic functions, the zeroth, first and second order, comprise nine parameters, but only eight parameters can be determined from the optic flow. The zeroth order remains undefined, because one cannot distinguish between the situation, where all objects have half the distance to the agent, and the situation, where the agent translates with double speed: The optical flow will be the same. Hence, from optic flow fields only the direction of the translation can be identified.

Between updates of the depth model, rotations, in particular, impair its validity, but can be easily compensated by counter-rotating the depth model. This is achieved by multiplying the depth dependent coupling matrix with a rotation matrix obtained from the rotation parameters of the current self-motion estimate.

The adaptive model will be derived first for the general case with an arbitrary field of view. Then a more biologically plausible adaptive model with a spherical field of view will be presented. A spherical field of view facilitates an intuitive interpretation of the depth model. In this specific case, the nine depth-dependent parameters are exactly the coefficients of the first three orders of the spherical harmonics expansion, i.e the dipole and quadrupole moments of the depth structure $\mu_{\vartheta\varphi}$.

2.6.1 Motivation for an adaptive MFA. In Fig 2 the initialization phase of the adaptive model is shown. The agent flies inside a sphere in such a way, that the depth model is the same for every trajectory step (see Figs 2 and 3B and methods 4.4 for details). The error in the estimated self-motion parameters decreases exponentially. Hence, the error is corrected to a large extent in the first few iteration steps.

In natural environments one can detect subspaces where the distances to a moving agent do not change over a certain time [35]. Hence, one can assume that the overall depth model changes only slightly from one step to the next in a given environment and can expect good self-motion estimates even for only a single iteration step of the KvD algorithm based on the old depth model. Furthermore, the depth model can be used even for a longer time interval. In this case the depth model is not updated instantaneously after receiving new optic flow information, but less frequently after several optic flow processing steps. Because the self-motion parameters and the depth model are formulated in the body coordinate system of the agent, the depth model has to be rotated together with the agent.

2.6.2 Matched filters and depth-dependent coupling matrix for the adaptive MFA. The equations of the modified KvD algorithm form the basis of the adaptive MFA (a small constant ε is inserted in Eq (33), the equation of the inverse distance, to avoid the singularity in case $\vec{d}_i = \vec{t}$):

$$\mu_i = -\frac{\vec{t} \cdot \left(\vec{p}_i - \vec{d}_i \times \vec{r}\right)}{1 - \left(\vec{t} \cdot \vec{d}_i\right)^2 + \varepsilon}, \tag{33}$$

$$\vec{t} = \frac{-1}{\langle \mu \rangle}\left(\langle \vec{p} \rangle + \vec{r} \times \left\langle \vec{d} \right\rangle - \left\langle \mu\left(\vec{t} \cdot \vec{d}\right)\vec{d} \right\rangle\right), \tag{34}$$

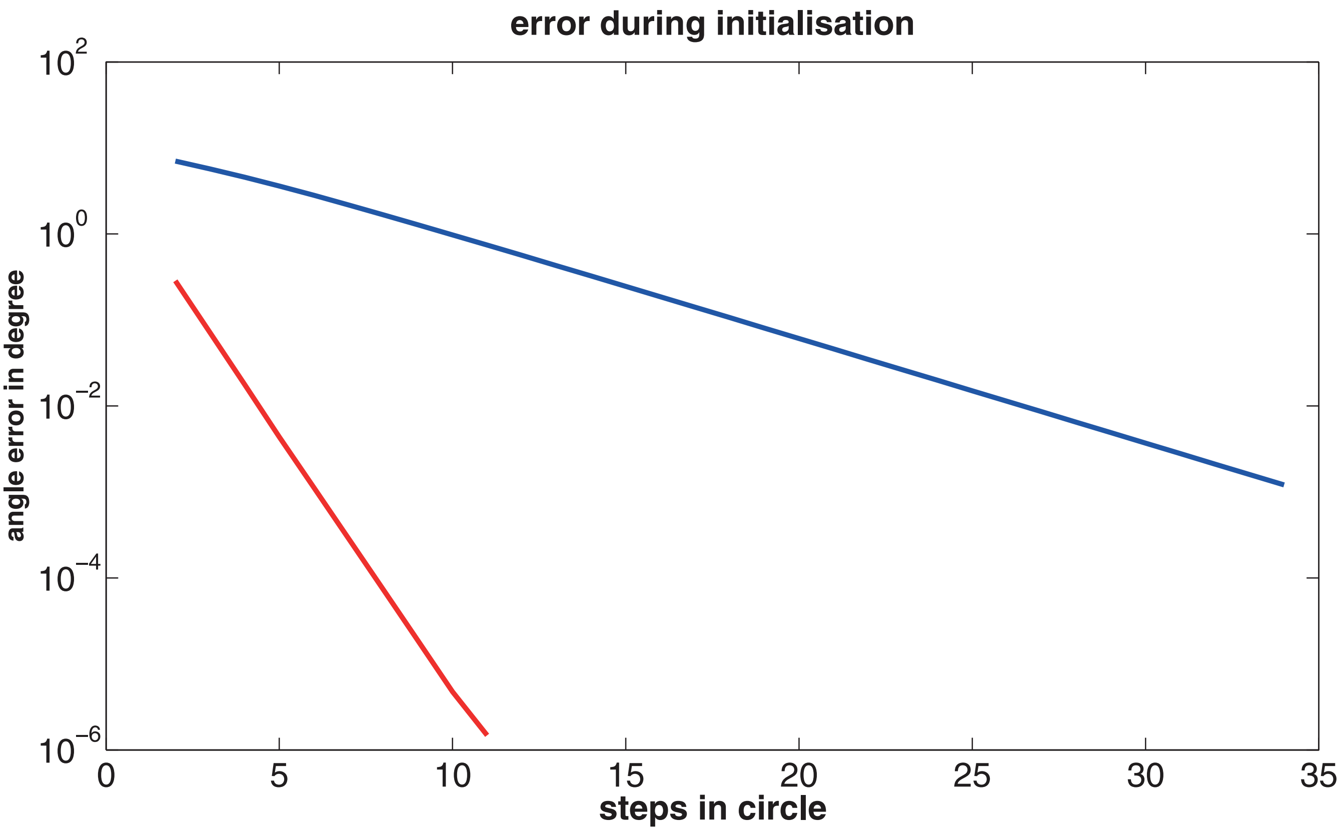

Fig 2. An agent flies inside the sphere on a circular trajectory (see Fig 3A). The center of the circle does not coincide with the center of the sphere to avoid symmetries in the depth model. The correct depth model is constant in this configuration, only the initialization of the depth model is tested. The y-axis shows the angle error $\arccos\left(\frac{\vec{\theta}^{\text{est}}\cdot\vec{\theta}}{|\vec{\theta}^{\text{est}}|\cdot|\vec{\theta}|}\right)$, between the estimated self-motion axis of $\vec{\theta}^{\text{est}}$ and the axis of the true self-motion values $\vec{\theta}$. The depth model is initialized with constant distances. With every update step of the depth model the error decreases exponentially for the translation (blue) as well as for the rotation (red).

doi:10.1371/journal.pone.0128413.g002

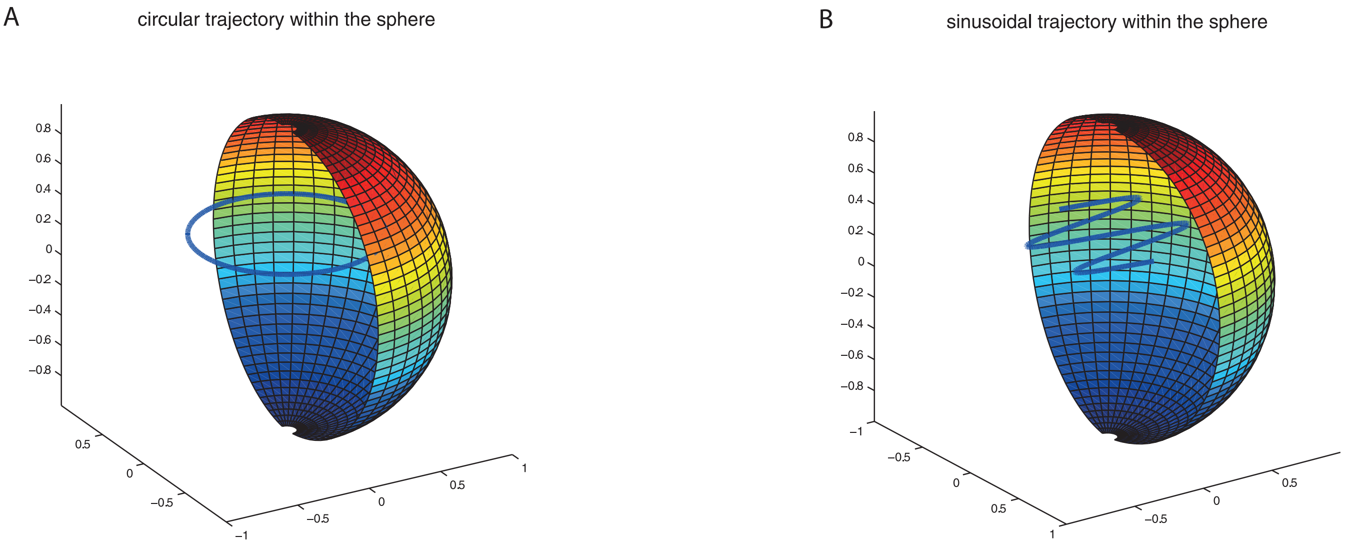

Fig 3. A) shows a circular trajectory to analyze the initialization phase of the adaptive MFA. The height of the trajectory lies above the middle point of the sphere to avoid trivial depth models. Due to symmetry the depth model for this configuration is the same at every trajectory point. **B)** shows a sinusoidal trajectory. It is used to analyze the self-motion estimation error during adaptation. Again the height of the trajectory is lifted up against the middle point of the sphere to make the depth model more complex in relation to an agent, which flies along the trajectory.

doi:10.1371/journal.pone.0128413.g003

$$\vec{r} = \left\langle \vec{p} \times \vec{d} \right\rangle + \vec{t} \times \left\langle \mu\vec{d} \right\rangle + \left\langle \left(\vec{r} \cdot \vec{d} \right) \vec{d} \right\rangle. \tag{35}$$

As shown in section 2.4 these iteration equations can be decomposed into the product of the current optical flow and a standard template, on the one side, and a coupling matrix, on the other side (see following equation). The coupling matrix is the part of the adaptive model that contains the depth values μ_i:

$$\begin{pmatrix} M^{tt} & M^{tr} \\ M^{rt} & M^{rr} \end{pmatrix} \cdot \begin{pmatrix} \vec{t} \\ \vec{r} \end{pmatrix} = \begin{pmatrix} \langle -\vec{p} \rangle \\ \left\langle \vec{p} \times \vec{d} \right\rangle \end{pmatrix}, \tag{36}$$

$$M^{tt} = \langle \mu \rangle I - \left\langle \mu\vec{d} \otimes \vec{d} \right\rangle, \tag{37}$$

$$M^{tr} = -\left\langle [\vec{d}\times] \right\rangle, \tag{38}$$

$$M^{rt} = \left\langle \mu [\vec{d}\times] \right\rangle, \tag{39}$$

$$M^{rr} = I - \left\langle \vec{d} \otimes \vec{d} \right\rangle, \tag{40}$$

where the matrix $[\vec{d}\times]$ is defined by

$$[\vec{d}\times]\vec{v} = \vec{d} \times \vec{v},$$

$$[\vec{d}\times] = \begin{pmatrix} 0 & -d_3 & d_2 \\ d_3 & 0 & -d_1 \\ -d_2 & d_1 & 0 \end{pmatrix},$$

i.e. the cross product of the two vectors $\vec{d}$ and $\vec{v}$ can be expressed as multiplication of matrix $[\vec{d}\times]$ and vector $\vec{v}$.

2.6.3 The case of spherically distributed flow vectors. For the special case of spherically distributed flow vectors the depth-dependent coupling matrix (see subsection (2.3.1) is given explicitly. In the simplest case of an agent being in the center of a sphere, i.e. if all inverse distances have the same value, the depth-dependent coupling matrix is proportional to the identity matrix (see end of subsection (2.3.1). In the general case, i.e. when the inverse distances can have arbitrary values, the entries of the depth-dependent coupling matrix are the expansion coefficients of the first three orders of the real valued spherical harmonic functions.

Then the environmental depth structure μ_i can be described by a real-valued function $\mu_{\vartheta\varphi}$ parameterized by the azimuth angle φ and the elevation angle ϑ (in a spherical coordinate system). Such functions can be described by an expansion of real-valued spherical harmonic functions [36] (see appendix 5.3 and Fig 4). Lower orders of these functions contain less details than higher order functions.

Given a function $\mu(\vartheta, \varphi)$ depending on azimuth angle φ and elevation angle ϑ the expansion coefficients a_{ln} are

$$a_{ln} = \int_{\varphi=0}^{2\pi} \int_{\vartheta=0}^{\pi} R_{ln}(\vartheta, \varphi) \cdot \mu(\vartheta, \varphi) \cdot \sin \vartheta d\vartheta d\varphi. \tag{41}$$

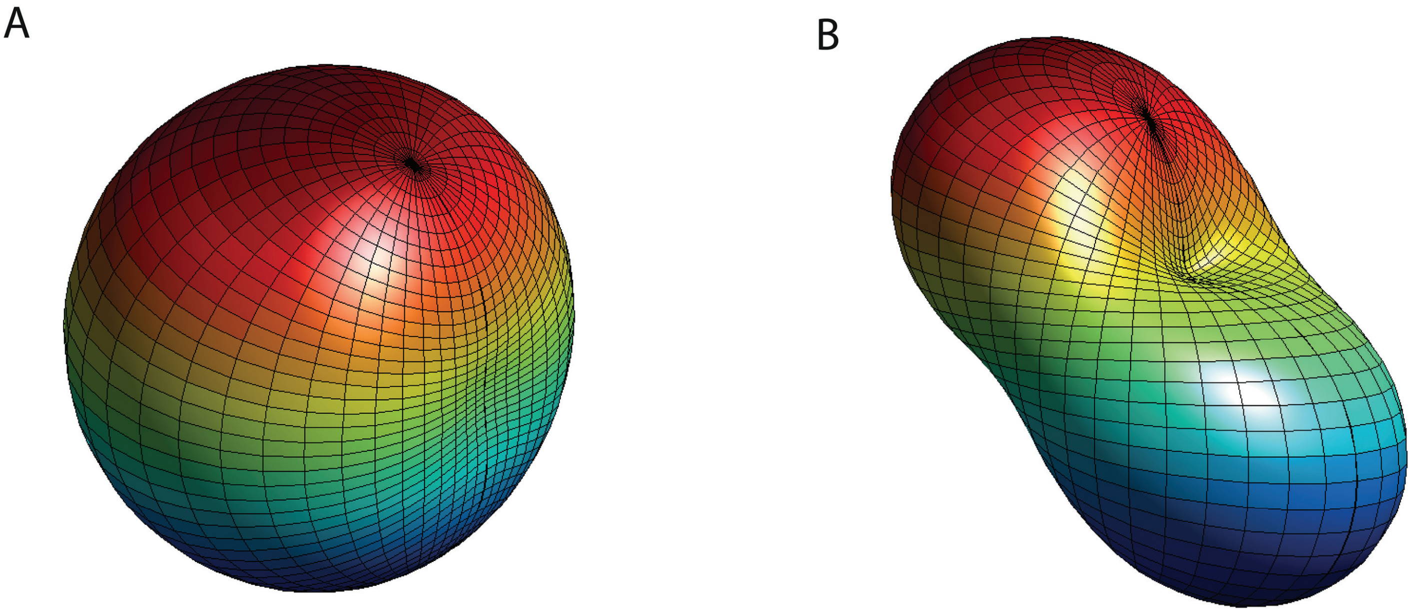

Fig 4. Spherical harmonic functions from the expansion of the inverse distances μ_i. A) The sum of the zeroth order function and a first-order dipole-function. **B)** The sum of the zeroth order function and a second-order quadrupole-function.

doi:10.1371/journal.pone.0128413.g004

$R_{ln}(\vartheta, \varphi)$ represent, for example, a dipole function for specific l and n. The corresponding coefficient a_{ln} provides information about how pronounced the specific dipole part is in the expanded function $\mu(\vartheta, \varphi)$.

With the help of all coefficients a_{ln} the expanded function $\mu(\vartheta, \varphi)$ is given by the reverse transformation

$$\mu(\vartheta, \varphi) = \sum_{l=0}^{N} \sum_{n=-l}^{+l} a_{ln} R_{ln}(\vartheta, \varphi), \tag{42}$$

with $N = 0, 1, 2, \ldots$.

The terms $\langle \mu \rangle$ and $\left\langle \mu \vec{d} \otimes \vec{d} \right\rangle$ in M^{tt} and the term $\left\langle \mu [\vec{d} \times] \right\rangle$ in M^{rt} are the sole terms which contain the inverse distances μ_i. In appendix (5.3) it is shown, that $\vec{d} \otimes \vec{d}$ and $\vec{d}$ can be expressed through real valued spherical harmonic functions when these term are given by their continuous counterparts $\vec{d}_{\vartheta\varphi}$. It can be directly seen that the continuous counterparts of $\vec{d}$ are the first-order real valued harmonic functions (except for a constant factor), whereas some transformations are needed to see that the continuous counterparts of $\vec{d} \otimes \vec{d}$ are linear combinations of the zeroth and second order real valued harmonic functions.

Due to the linearity of an integral expression and the orthogonality of the spherical harmonic functions [36],

$$\int R_{ln}(\vartheta, \varphi) R_{l'n'}(\vartheta, \varphi) \, d\Omega = \delta_{ll'} \delta_{nn'}, \tag{43}$$

the terms $\langle \mu \rangle$, $\left\langle \mu \vec{d} \otimes \vec{d} \right\rangle$ and $\left\langle \mu [\vec{d} \times] \right\rangle$ can be interpreted as the definition equations for specific coefficients of a spherical harmonic expansion of $\mu(\vartheta, \varphi)$ [see Eq (41)]. Hence, in the spherical case $\langle \mu \rangle$, $\left\langle \mu \vec{d} \otimes \vec{d} \right\rangle$ and $\left\langle \mu [\vec{d} \times] \right\rangle$ in the coupling matrix (Eqs (38) until (40)) can be replaced by the coefficients of spherical harmonic functions:

$$M^{tt} = \sqrt{4\pi} \cdot a \cdot \left(I - \frac{1}{3}I\right) - \sqrt{\frac{4\pi}{15}} \cdot C$$

$$C = \begin{pmatrix} -c_4 - \sqrt{\frac{1}{3}} \cdot c_1 & c_5 & c_2 \\ c_5 & c_4 - \sqrt{\frac{1}{3}} \cdot c_1 & c_3 \\ c_2 & c_3 & \sqrt{\frac{4}{3}} \cdot c_1 \end{pmatrix}$$

$$M^{tr} = 0$$

$$M^{rt} = \sqrt{\frac{4\pi}{3}} \cdot B$$

$$B = \begin{pmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{pmatrix}$$

$$M^{rr} = \frac{2}{3}I$$

where a is the coefficient of the zeroth spherical harmonic function, which cannot be determined by the algorithm as explained earlier. The three b's are the three coefficients of the first-order spherical harmonic functions, the dipole functions. The five c's are the five coefficients of the second-order spherical harmonic functions, the quadrupole functions.

The only non-constant parameters in the depth-dependent matrix are the nine coefficients of the expansion by spherical harmonics. The nine parameters the agent has to memorize during flight have, in the case regarded here, a physical interpretation: They are the dipole and quadrupole parts of the depth distribution of the environment. The non-existence of higher-order coefficients in the depth-dependent matrix indicates that these orders contain no information for solving the self-motion problem. If the distances are constant, the matrix M is the identity matrix (except for a normalization constant) as mentioned before.

The computation of the nine coefficients might not seem to be biologically plausible at first sight. However, the computation of one coefficient a_{ln} of the expansion corresponds to a weighted wide-field integration and is reminiscent of the function of LPTCs in flies [16, 17, 25, 26]. One could imagine a neuron for each of the nine parameters a_{ln}, which represents a specific global property of the depth structure.

With regard to the computational effort a full inversion of a 6 × 6 matrix is not required. The submatrix M^{tr} is zero in the spherical case, hence only an inversion of the submatrix M^{tt} is required. If the quadrupoles in this submatrix are sufficiently small, the inverse matrix can be linearly approximated by a Neumann series [37], $(I–A)^{-1} = I+A+A^2+A^3+\ldots$.

2.6.4 Test of the adaptive MFA. In this section we compare quantitatively the adaptive and the original MFA. We present a simulation in a very simple environment where the agent moves inside a sphere (see Fig 3 and methods 4.4). Nothing is known in advance about the flight directions and whereabouts of the agent inside the sphere. Hence, the covariance matrices of the original MFA are set proportional to the unit matrix. The chosen trajectory in this setting is a sinusoidal curve. On this trajectory the current depth distribution differs essentially from that in the center of the sphere (Fig 3A).

The amount of translation and rotation in each step varies along the trajectory. The steps are chosen so that the maximum rotation angle is about 8 degrees per step ensuring that the approximation of the KvD algorithm, which is valid for small rotation angles, still holds. The maximum rotation angles between the processed images is in the same range as the maximum rotation angles during saccades of flying insects [38, 39].

On the whole, the adaptive MFA performs better than the original non-adaptive MFA (Fig 5). The second y-axes on the right side of the figure show the averaged ratio between the rotational and translational optic flow at every time step. Due to the sinusoidal trajectory of the agent this ratio varies between a factor of hundred in favor of the rotational or translational optic flow. When an optic flow component is overlayed hundredfold by the other flow component the coupling terms $\vec{r} \times \left\langle \vec{d} \right\rangle$ and $\vec{t} \times \left\langle \mu\vec{d} \right\rangle$ between the flow components are the dominant parts in the related estimation Eqs (34) and (35). For a spherical field of view the term $\vec{r} \times \left\langle \vec{d} \right\rangle$ is zero. But the term $\vec{t} \times \left\langle \mu\vec{d} \right\rangle$ depends on the dipole components of the inverse distance μ (see appendix 5.3.1). Hence, only the rotation estimate is affected by errors in the estimated dipoles. Because the original MFA does not determine the values for the dipole components of the inverse distance for the current optical flow, the rotation estimates become totally useless, whenever the translational flow component is dominant. The angle error between the estimated rotation axis and the true rotation axis increases to a value of about hundred degrees. Whereas the angle error of the adaptive MFA does not exceed few degrees.

If an Gaussian error is added to the optical flow with a standard deviation of ten percent of the averaged overall flow value (Fig 5), one could expect that a translational or rotational flow component, which is overlayed hundredfold by the other flow component disappears totally in this flow error, because the error is tenfold higher as the flow component in this situation. But the estimators use hundreds or a few thousands of flow vectors to estimate self-motion. Due to the large number of flow vectors (insects usually have between a few hundred (e.g. fruit fly) and a few thousand ommatidia (bee, dragonfly) per eye), the self-motion can be still estimated in this case within a useful error range. The results are shown for about 5000 flow vectors (Fig 5). The error increases for both the translational and the rotational self-motion estimate to a value of about 10 degrees. This error is additive to the error described in the upper panels of Fig 5 and affects the adaptive MFA in the same way as the original MFA.

In the bottom panels of Fig 5 different update rates are tested. Even for an update at only every twentieth optical flow processing step the errors remain in a useful range.

Albeit the simplicity of the simulation it shows some basic features of the compared algorithms. The simulation does not generate any outliers due to moving objects or depth discontinuities. A small number of outliers will not affect the MFA as a consequence of the linear summation over thousands of optic flow vectors. More complex simulations in virtual environments with rendered images and EMDs is left to future work (Strübbe et al., in prep.).

3 Discussion

The aim of this study is to develop an adaptive matched filter approach to self-motion estimation which could be in principle the underlying concept of self-motion estimation in flying insects. As a novel characteristic, this approach assumes an adaptation to the depth structure in the insect visual motion pathway, an assumption that is supported by recent experimental evidence [31, 40, 41]. Our approach starts from a theoretical point of view by analysing and unifying the

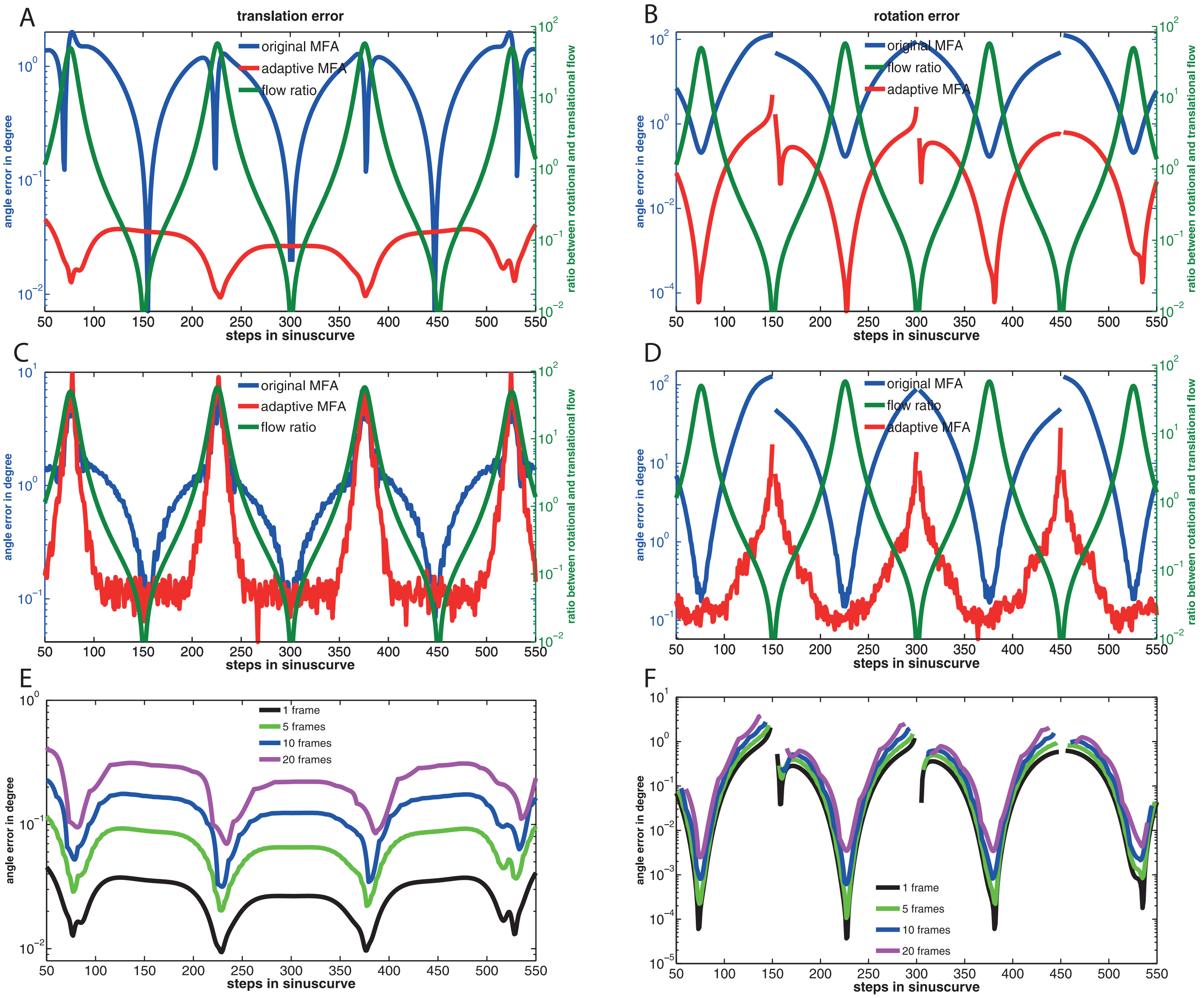

Fig 5. As in Fig 2 an agent flies inside a sphere (see methods 4.4). The trajectory is a sinus curve with two full periods. The amplitude of the sinus curve is 0.5 of the radius of the sphere. The sinus curve is lifted in the sphere by 0.3 units in z-direction to avoid symmetries in the depth model. The agent performs 600 steps which result in rotation angles of up to 8 degrees per frame. The initialization phase is not shown. The left figures shows the error of the translation estimates and the right figures shows the error of the rotation estimates. The y-axes show the angle error $\arccos\left(\left(\frac{\vec{\theta}^{\text{est}}\cdot\vec{\theta}}{|\vec{\theta}^{\text{est}}|\cdot|\vec{\theta}|}\right)\right)$, between the estimated self-motion axis of $\vec{\theta}^{\text{est}}$ and the axis of the true self-motion values $\vec{\theta}$. The estimated angle errors have a pole, when the rotation gets zero at the inflection points of the sinusoidal curve (see methods 4.4). The small region around the poles are cut out in the figures. **A, B)** The two figures show the error of the adaptive MFA (red curve) and the original MFA (blue curve) with a constant inverse distant assumption for the original MFA. The adaptive MFA is updated every time step. The right y-axes of the figures show the averaged ratio between the rotational and translational optic flow. **C, D)** shows the adaptive MFA and the original MFA as in figures A and B, but with an error of 10% added to the optical flow. **E, F)** show different update frequencies of the depth model. All models rotate with the agent. The update frequencies are: black = 1 frame, green = 5 frames, blue = 10 frames and violet = 20 frames.

doi:10.1371/journal.pone.0128413.g005

non-adaptive matched filter approach (MFA) to self-motion estimation together with the Koenderink van Doorn (KVD) algorithm which incorporates an estimation of the depth values.

To take advantage of both algorithms, some mathematical problems had to be solved. First, it was shown that the two algorithms are equivalent in case the distances to objects in the environment are assumed to be known. Secondly, a bias in the KvD algorithm was removed by a small correction of the iteration equations. And last but not least, an analysis of the specific case of a spherical field of view, reminiscent of that of flying insects, shows that the depth structure can be represented by only eight parameters without losing relevant information for self-motion estimation and that these eight parameters are the dipole and quadrupole moments of the spherical harmonics.

Technical and biological systems have different origins and often operate under different conditions. Biological systems arise through evolutionary adaptation. They usually have to operate in a great variety of environments. Hence, the neural computations underlying the animal's behavior need to be particularly robust. In addition, the animal has restrictions with respect to its computational hardware. Neuronal circuits can perform linear transformations in parallel by a dendritic weighted summation of the inputs of a neuron and non-linear operations through the non-linear response behavior of a neuron to its overall input. Nonetheless, a non-linear operation, such as computing the inverse of a matrix, with changing entries, is not easy to implement by neuronal hardware. The bio-inspired computational model analyzed here is the MFA of self-motion estimation. It is a linear model for a fixed depth assumption, derived under the side condition of maximal robustness against errors in the measured optical flow field (see Eq (5) from Franz et al. [11]).

The KvD algorithm of self-motion estimation which is compared with the MFA model was derived analytically in a technical framework on the basis of a minimization principle. The resulting iteration equations represent a gradient descent where the current self-motion parameters are used to determine a better depth model and the new depth model is used to determine a better estimate of self-motion in the next iteration step. If one considers only one iteration step, where the depth model is seen as fixed, the self-motion estimation is linear. It is not only linear, but also equivalent to the biologically inspired MFA which uses a fixed depth distribution.

The equivalence of both models becomes evident within the framework of linear estimator theory. There exists an unique optimal estimator for a linear estimation problem with error-prone inputs. The Gauss-Markov theorem describes this estimator. Both compared methods represent this optimal solution.

However, some differences exists. In the MFA Franz et al. [11] weighted the filters by matrices that represent additional assumptions about the situation under which the self-motion is estimated. If these assumptions are correct, the weighting of the filters improves self-motion estimation; however, if these assumptions are incorrect in the current situation, the estimator gets worse. Hence, the additional matrices make the estimator more specific. These matrices can also be implemented in the KvD algorithm by a modification of the minimization principle. When, for example, it is known, that the optical flow can be measured more accurately below the horizon, because the objects are generally closer there, this knowledge can be taken into account by introducing weights in the initial equation. We argue that it is not always useful to take knowledge about the preferred self-motion directions into account. Even when the moving agent solely translates in the forward direction a disturbance can lead to a passive translation also in other directions.

From a mathematical point of view the bias of the KvD algorithm is remarkable. When the depth distribution of the environment has to be determined together with the self-motion parameters, the estimation problem is no longer linear. The standard procedure for estimating parameters from inputs, which are disturbed by Gaussian errors, is the minimization of the mean squared error. It might be counter-intuitive that the true self-motion values are not even a local minimum. The standard approach fails, because the standard condition assumes that an increasing number of measured values, here the flow vectors, are accompanied by a fixed number of estimation parameters. However, for the non-linear estimation problem the number of distance values increases together with the number of flow vectors. Only the number of self-motion parameters remains constant. With every additional flow vector additional information about the self-motion parameters is obtained, because one gets two additional independent values from the flow vector, but only one additional parameter has to be estimated (the distance corresponding to the flow vector). However, the standard approach does not use this additional information in an optimal way.

Here we derived a modified version of the KvD algorithm that is not derived from a minimization principle. Hence it is not clear whether it leads to the best estimate for a given finite number of flow vectors. Rather, the numerical simulation indicates that the modified version has the desired property that the algorithm converges to the real self-motion values for an infinite number of flow vectors. It is left to further mathematical work to analyze optimal criteria for the non-linear estimation problem in case of a finite number of flow vectors.

Based on this modified version of the KvD algorithm an adaptive MFA was derived. It was shown that it is a critical issue to correctly determine the dipole components. If a small rotation is superimposed by a large translation the non-adaptive MFA cannot provide useful rotation estimates, whereas the adaptive MFA is accurate up to a few degrees. A situation where a relatively large translation encounters a relatively small rotation is given in the inter-saccadic phases of insect flight [38, 39, 42]. In these phases the insect tries to avoid any rotation. If the insect stabilizes its flight with the help of the visual system, the non-adaptive MFA cannot be the underlying concept to detect small rotations in these phases in environments it is not tuned to. To estimate rotations, which are superimposed by a large translation, one has to determine the current dipole components, as is done by the adaptive MFA.

The adaptive MFA was inspired by the finding that for a spherical field of view the depth structure of the environment can be represented by only eight parameters without losing relevant information for self-motion estimation and by the fact that the visual system of insects has an almost spherical field of view. The spherical field of view is also a desirable property of technical systems which are designed to estimate their self-motion on the basis of optical flow fields. Such systems can be realized by panoramic cameras [43, 44].

Adaptation to the depth structure of the environment means that the adaptation takes place on another time scale than the image processing itself. Hence, some information about the depth structure has to be memorized by the system. The result that exact self-motion can only be estimated for a spherical field of view, if eight parameters about the depth structure of the environment are known, is therefore in accordance with the limited computational resources of insects.

Motion adaptation was analyzed in the insect visual pathway and found to depend on the overall velocity in the visual field [27, 31, 32, 40, 41, 45]. Since, at least during translational motion, the overall retinal velocity depends on the depth distribution of the environment. The experimentally characterized processes of motion adaptation may well play a role in an adaptive mechanism of self-motion estimation as proposed in the present study. Here we give a short analysis from a theoretical point of view which components are needed for the adaptive MFA. Minimalistically, one needs eight model neurons for the eight depth parameters. The weighted summation over the inputs of one of these model neurons corresponds to one of the eight integrals over the depth distribution, where the spherical harmonic functions play the role of the weighting parameters. Examples of neurons performing such an integration are the LPTC neurons of flies, the neuronal candidates for the six model neurons, which represent the matched filters for the six self-motion components. Given the properties of LPTCs [16, 17, 24–26], it is likely that one hypothetical model neuron for depth representation does not cover the whole sphere. Due to the linearity of self-motion estimation LPTCs can be combined to represent one self-motion component. On this basis, it might be possible that one LPTC codes information for both translation and rotation when the corresponding flow fields resemble each other within the receptive field of the neuron. Hence, the hypothetical depth neurons could be realized by a network of neurons with each neuron receiving input from only part of the visual field.

The hypothetical neurons representing the depth structure need some pre- and post-processing. In the adaptive MFA, only the pre-processing contains non-linear operations, namely the transformation of the optical flow into local depth values that are integrated afterwards by the hypothetical model neurons. One can assume that the determination of the depth values is simplified during nearly pure translation as is characteristic of the insect saccadic flight strategy [19]. In these phases the depth structure can be determined more easily, because the optic flow is not superimposed by a rotational component [46, 47].

The post-processing concerns the determination of the depth-dependent matrix which corrects the outputs of the six model neurons, corresponding to the motion sensitive LPTC cells. From a mathematical point of view, two subsequent linear transformations can be combined in a single linear transformation. In the adaptive MFA we have two subsequent linear transformations: the fixed linear transformation by the six model neurons that receive direct optical flow input and the adaptive linear transformation by the depth-dependent matrix, the entries of which are the responses of the eight depth neurons. There are two options where the adaptation could take place: The two linear transformations could be merged into one linear transformation, which means that the adaptation takes place at an early stage of optic flow processing. Alternatively, one could assume that the two linear transformations are spatially separated, and the depth-dependent matrix is realized by an adapting linear circuitry, which wires the early stage neurons.

The linear transformation given by the depth dependent matrix can be obtained without a matrix inversion by applying an appropriate linearization of the inverse depth dependent matrix (see section 2.6.3). With this simplification and the above simplification of depth capturing the adaptive MFA can be realized by relatively simple neuronal circuits.

4 Materials and Methods

4.1 Numerical test and simulation

We used a numerical test (Fig 1) and a simulation (Fig 5) to show the performance of the considered self-motion estimators based on optical flow fields. The optical flow fields used here are computed in all cases from the flow Eq (1) where the distances, the viewing directions and the self-motion parameters are given. In some cases a Gaussian error was added to the optical flow values. The task of the self-motion estimators is to determine the self-motion parameters from these flow fields.

Whereas the distances in the numerical test are obtained by a random process, the distances in the simulation are defined by the environment. In the numerical test the individual estimates are independent of each other. In the simulation a trajectory through the environment is constructed to enable an agent to use the depth model of the environment for a series of estimates at subsequent trajectory points.

For programming and testing the algorithm the programming language Matlab was used.

4.2 Construction of a spherical field of view

For the simulation and the numerical test a spherical visual field of view is needed. It is not a trivial task to arrange a number of viewing directions on a sphere in a way that the density is equally distributed. Here we used an iterative solution. The iteration starts with a Platonic solid, an octahedron. An octahedron has eight faces: Eight equilateral triangles with the same side lengths, which surround a symmetric solid.

In each iteration step every triangle is replaced by four new triangles with one triangle placed in the middle of the old triangle and the other three are placed in corners. The mid-points of the sides of the old triangle coincide with the corners of the new triangles. These mid-points are projected radially on the surface of the sphere which surround the object so that all corners of the new triangles lie on this sphere.

In each iteration step the sphere is covered with nearly identical triangles. After the last iteration step the mid-points of each triangle give a viewing directions. With an arbitrary number of iterations n the number of viewing directions are $8 \cdot 4^n$.

4.3 Numerical test of the bias of the KvD algorithm

In the numerical test of the bias of the KvD algorithm two configurations are tested, one with a spherical field of view and one with a non-spherical field. The non-spherical field of view is realized by the same procedure as in the spherical case (see methods 4.2), but two starting triangles are left out. The omitted triangles are opposite triangles both placed in the upper part of the sphere. This configuration has some obvious symmetry, but avoids the case that each omitted viewing direction has an omitted counterpart on the other side of the sphere.

The number of viewing directions is increased by increasing the number of iterations starting from an octahedron. Each step increases the number of viewing directions by four. For each number of viewing directions 40 self-motion estimation trials are tested.

4.4 Simulation of the adaptive MFA

The environment used for the simulation is a unit sphere in which the agent flies on a sinusoidal trajectory. The effect of adaptation should be larger if, for example, the agent flies outdoors and then enters a tunnel which is a common experimental set-up for navigation studies in flying insects [48–51]. However, the test of the adaptive MFA in this kind of environment needs a 3-d engine for rendering images, on which the flow vectors are estimated with motion detectors like the Reichardt detector or the Lucas-Kanade detector. This issue will be addressed in another study (Strübbe et al., in prep.).

For the trajectory a sinusoidal curve was chosen (see Fig 3 right). Although it is typical that flying insects use a saccadic flight strategy to separate translation and rotation, the self-motion estimators in this study were analyzed under the condition of a combined translation and rotation. In each run the agent flies along the trajectory and the angle error between the true self-motion axes and the estimated axes of the tested estimators are shown in Fig 5.

5 Appendix

5.1 Derivation of the equivalence of the MFA and KvD algorithm

From Eq (17) one obtains together with the equations for the coupling matrix Eqs (21), (22) and (23) and the definitions of the templates Eqs (13) and (14):

$$\vec{a} = \hat{M} \cdot \vec{\theta} = \begin{pmatrix} M^{tt} & M^{tr} \\ M^{rt} & M^{rr} \end{pmatrix} \cdot \begin{pmatrix} \vec{t} \\ \vec{r} \end{pmatrix} = \begin{pmatrix} \left\langle \vec{T}^t_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^t_3 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_1 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_2 \cdot \vec{p} \right\rangle \\ \left\langle \vec{T}^r_3 \cdot \vec{p} \right\rangle \end{pmatrix} \tag{44}$$

$$(M^{tt})_{ab} = \langle\mu^2\rangle\vec{e}_a\cdot\vec{e}_b - \left\langle \mu^2\left(\vec{e}_a\cdot\vec{d}\right)\cdot\left(\vec{e}_b\cdot\vec{d}\right)\right\rangle \tag{45}$$

$$(M^{tr})_{ab} = \left\langle \mu(\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right\rangle \tag{46}$$

$$(M^{rt})_{ab} = -\left\langle \mu(\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right\rangle \tag{47}$$

$$(M^{rr})_{ab} = \vec{e}_a\cdot\vec{e}_b - \left\langle\left(\vec{e}_a\cdot\vec{d}\right)\cdot\left(\vec{e}_b\cdot\vec{d}\right)\right\rangle \tag{48}$$

$$\vec{T}^t_{ia} = -\mu_i\left(\vec{d}_i\times\vec{e}_a\times\vec{d}_i\right) \tag{49}$$

$$\vec{T}^r_{ia} = -\vec{e}_a\times\vec{d}_i \tag{50}$$

Multiplying $\vec{t}$ and $\vec{r}$ with the matrices and $\vec{p}$ with the templates leads to the two equations:

$$\begin{aligned}\left(M^{tt}\cdot\vec{t} + M^{tr}\cdot\vec{r}\right)_a = \quad & \langle\mu^2\rangle t_a - \left\langle \mu^2\sum_b\left(\vec{e}_a\cdot\vec{d}\right)\cdot\left(\vec{e}_b\cdot\vec{d}\right)\cdot t_b\right\rangle \\ & + \left\langle \mu\sum_b\left((\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right)\cdot r_b\right\rangle\end{aligned} \tag{51}$$

$$= \left\langle \vec{T}^t_a\cdot\vec{p}\right\rangle = \left\langle \mu\vec{p}\cdot\left(-\vec{d}\times\vec{e}_a\times\vec{d}\right)\right\rangle \tag{52}$$

$$\begin{aligned}\left(M^{rt}\cdot\vec{t} + M^{rr}\cdot\vec{r}\right)_a = \quad & -\left\langle \mu\sum_b\left((\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right)\cdot r_b\right\rangle \\ & + t_a - \left\langle \sum_b\left(\vec{e}_a\cdot\vec{d}\right)\cdot\left(\vec{e}_b\cdot\vec{d}\right)\cdot t_b\right\rangle\end{aligned} \tag{53}$$

$$= \left\langle \vec{T}^r_a\cdot\vec{p}\right\rangle = \left\langle \vec{p}\cdot\left(\vec{d}\times\vec{e}_a\right)\right\rangle \tag{54}$$

The term $\sum_b\left[\left(\vec{e}_a\cdot\vec{d}\right)\cdot\left(\vec{e}_b\cdot\vec{d}\right)\right]_{ab}\cdot v_b$ with an arbitrary vector $\vec{v}$ can be expressed through the dyadic product $\sum_b\left[\vec{d}\otimes\vec{d}\right]_{ab}\cdot v_b = \left(\vec{v}\cdot\vec{d}\right)d_a$, and the term $\sum_b\left((\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right)\cdot v_b$ can be transformed with the triple product rule into

$$\begin{aligned}\sum_b\left((\vec{e}_a\times\vec{e}_b)\cdot\vec{d}\right)\cdot v_b \quad &= \quad \left(\vec{d}\times\vec{e}_a\right)\cdot\textstyle\sum_b\vec{e}_b\cdot v_b = \left(\vec{d}\times\vec{e}_a\right)\cdot\vec{v} \\ &= \left(\vec{v}\times\vec{d}\right)\cdot\vec{e}_a = [\vec{v}\times\vec{d}]_a\end{aligned} \tag{55}$$

The right side of the Eq (52), $\left\langle \mu\vec{p}\cdot\left(-\vec{d}\times\vec{e}_a\times\vec{d}\right)\right\rangle$, is equal to $\left\langle \mu\vec{p}\cdot\left(-\vec{e}_a + \left(\vec{e}_a\cdot\vec{d}\right)\vec{d}\right)\right\rangle = \langle -\mu\vec{p}\cdot\vec{e}_a\rangle = -[\langle\mu\vec{p}\rangle]_a$, because $\vec{p}$ and $\vec{d}$ are orthogonal. With the triple product identity $\left\langle \vec{p}\cdot\left(\vec{d}\times\vec{e}_a\right)\right\rangle$ in Eq (54) is transformed into $\left\langle \vec{e}_a\cdot\left(\vec{p}\times\vec{d}\right)\right\rangle = \left[\left\langle \vec{p}\times\vec{d}\right\rangle\right]_a$.

Writing the a indexed values as vectors, we obtain

$$\langle\mu^2\rangle\vec{t} - \left\langle \mu^2\left(\vec{t}\cdot\vec{d}\right)\vec{d}\right\rangle + \vec{r}\times\left\langle \mu\vec{d}\right\rangle = -\langle\mu\vec{p}\rangle$$

$$-\vec{t}\times\left\langle \mu\vec{d}\right\rangle + \vec{r} - \left\langle\left(\vec{r}\cdot\vec{d}\right)\vec{d}\right\rangle = \left\langle \vec{p}\times\vec{d}\right\rangle,$$

which results in the iteration Eqs (11) and (12) of the KvD algorithm:

$$\vec{t} = -\frac{1}{\langle\mu^2\rangle}\{\langle\mu\vec{p}\rangle + \vec{r}\times\left\langle \mu\vec{d}\right\rangle - \left\langle \mu^2\left(\vec{t}\cdot\vec{d}\right)\vec{d}\right\rangle\},$$

$$\vec{r} = \left\langle \vec{p}\times\vec{d}\right\rangle + \vec{t}\times\left\langle \mu\vec{d}\right\rangle + \left\langle\left(\vec{r}\cdot\vec{d}\right)\vec{d}\right\rangle,$$

where $\frac{1}{\langle\mu^2\rangle}$ represents the Lagrange multiplier ξ.

5.2 Bias-term of the KvD algorithm

The iteration Eqs (10), (11) and (12) do not converge to the true parameters $\vec{t}^0$, $\vec{r}^0$ and μ_i^0 in general, even in the limit of an infinite number of flow vectors. We consider unbiased and independently distributed flow vector errors, $E\left(\Delta\vec{p}_i\right) = 0$, $\mathrm{cov}\left(\Delta\vec{p}_i, \Delta\vec{p}_j\right) = 0$ if $i \neq j$. Their variance $\mathrm{var}(\Delta\vec{p}_i)$ may depend on the direction i.

The mean of an infinite number of independent flow vector errors converges to the mean of the expectation value,

$$\langle\Delta\vec{p}_i\rangle_\infty = 0, \tag{56}$$

for an arbitrary integration area around viewing direction i. Because the error vectors are independent from each other, the mean of the product of the error vectors and an arbitrary direction dependent vector function $\vec{f}_i$ is zero,

$$\left\langle \vec{f}_i\cdot\Delta\vec{p}_i\right\rangle_\infty = 0. \tag{57}$$

To show that the KvD algorithm does not converge to the real values for an infinite number of flow vectors, we start by assuming the opposite. If the KvD algorithm had a minimum at the true values $\vec{t}^0$, $\vec{r}^0$ and μ_i^0 we could insert these values in the iteration Eqs (10), (11) and (12) and would get the same values back. Substituting the true values for translation and rotation in the equation for the nearness Eq (10), we get the nearness error $\triangle\mu_i$ which directly depends on the flow vector errors:

$$\Delta\mu_i = -\frac{\vec{t}^0\cdot\Delta\vec{p}_i}{1-\left(\vec{t}^0\cdot\vec{d}_i\right)^2}.$$

The estimated translation after an infinite number of iterations $\vec{t}(\infty) = \lim_{n\to\infty}\vec{t}(n)$ shows the following equation. Please note that we consider two limits: the limit of infinite iteration steps, $\lim_{n\to\infty}$, and the limit of an infinite number of flow vectors, indicated by an infinity symbol as index of the brackets which stands for the mean over all viewing directions.

$$\begin{aligned}\vec{t}(\infty) &= \vec{t}^0 + \Delta\vec{t} = -\xi\left\{\langle\mu\vec{p}\rangle_\infty + \vec{r}^0 \times \left\langle\mu\vec{d}\right\rangle_\infty - \left\langle\mu^2\left(\vec{t}^0\cdot\vec{d}\right)\vec{d}\right\rangle_\infty\right\} \\ &= -\xi\left\{\langle(\mu^0+\Delta\mu)(\vec{p}^0+\Delta\vec{p})\rangle_\infty + \vec{r}^0 \times \left\langle(\mu^0+\Delta\mu)\vec{d}\right\rangle_\infty - \left\langle(\mu^0+\Delta\mu)^2\left(\vec{t}^0\cdot\vec{d}\right)\vec{d}\right\rangle_\infty\right\}.\end{aligned}$$

If $\vec{t}(\infty)$ is a stable minimum $\triangle\vec{t}$ must vanish:

$$\begin{aligned}\triangle\vec{t} &= -\xi\left\{\langle\mu^0\Delta\vec{p} + \triangle\mu\vec{p}^0 + \triangle\mu\Delta\vec{p}\rangle_\infty + \vec{r}^0 \times \left\langle\triangle\mu\vec{d}\right\rangle_\infty - \left\langle(2\mu^0 \triangle \mu + \triangle\mu^2)\left(\vec{t}^0\cdot\vec{d}\right)\vec{d}\right\rangle_\infty\right\} \\ &= -\xi\left\{\langle\triangle\mu\Delta\vec{p}\rangle_\infty - \left\langle\triangle\mu^2\left(\vec{t}^0\cdot\vec{d}\right)\vec{d}\right\rangle_\infty\right\} \\ &= \xi\left\{\left\langle\frac{\vec{t}^0\cdot\Delta\vec{p}}{1-\left(\vec{t}^0\cdot\vec{d}\right)^2}\Delta\vec{p}\right\rangle_\infty + \left\langle\frac{(\vec{t}^0\cdot\Delta\vec{p})(\Delta\vec{p}\cdot\vec{t}^0)}{\left(1-\left(\vec{t}^0\cdot\vec{d}\right)^2\right)^2}\left(\vec{t}^0\cdot\vec{d}\right)\vec{d}\right\rangle_\infty\right\} \\ &= \xi\left\{\left\langle\frac{\Delta\vec{p}\otimes\Delta\vec{p}}{1-\left(\vec{t}^0\cdot\vec{d}\right)^2}\right\rangle_\infty \vec{t}^0 + \left\langle\frac{\vec{t}^0\cdot(\Delta\vec{p}\otimes\Delta\vec{p})\cdot\vec{t}^0}{\left(1-\left(\vec{t}^0\cdot\vec{d}\right)^2\right)^2}(\vec{d}\otimes\vec{d})\right\rangle_\infty \vec{t}^0\right\}\end{aligned}$$

For the analysis of the conditions under which the term $\triangle\vec{t}$ vanishes, see the text (subsection 2.5.1).

5.3 Expression of $\vec{d}$ and $\vec{d}\otimes\vec{d}$ by spherical harmonics

The vector $\vec{d}$ and the dyadic product $\vec{d}\otimes\vec{d}$ are expressed by linear combinations of real-valued spherical harmonics, which are itself linear combinations of complex spherical harmonicsY_{lm},

$$R_{lm} = \begin{cases}\frac{1}{\sqrt{2}}\left(Y_{lm} + (-1)^m Y_{l(-m)}\right) & m > 0 \\ Y_{l0} & m = 0 \\ \frac{1}{\sqrt{2}i}\left(Y_{l(-m)} - (-1)^m Y_{lm}\right) & m < 0\end{cases} \tag{58}$$

The real-valued spherical harmonics from zero order to second-order are:

Zeroth-Order

$$g_0 = \sqrt{\frac{1}{4\pi}} \tag{59}$$

First-order

$$f_1 = \sqrt{\frac{3}{4\pi}}\sin\vartheta\cos\varphi \propto \sin\vartheta\cos\varphi \tag{60}$$

$$f_2 = \sqrt{\frac{3}{4\pi}}\sin\vartheta\sin\varphi \propto \sin\vartheta\sin\varphi \tag{61}$$

$$f_3 = \sqrt{\frac{3}{4\pi}} \cos\vartheta \propto \cos\vartheta \tag{62}$$

Second-order

$$h_1 = \sqrt{\frac{5}{16\pi}} (3\cos^2\vartheta - 1) \propto 3\cos^2\vartheta - 1 \tag{63}$$

$$h_2 = \sqrt{\frac{15}{4\pi}} \sin\vartheta\cos\vartheta\cos\varphi \propto \sin\vartheta\cos\vartheta\cos\varphi \tag{64}$$

$$h_3 = \sqrt{\frac{15}{4\pi}} \sin\vartheta\cos\vartheta\sin\varphi \propto \sin\vartheta\cos\vartheta\sin\varphi \tag{65}$$

$$h_4 = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta\cos(2\varphi) \propto \sin^2\vartheta\cos(2\varphi) \tag{66}$$

$$h_5 = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta\sin(2\varphi) \propto \sin^2\vartheta\sin(2\varphi) \tag{67}$$

5.3.1 Expression of $\vec{d}$ through spherical harmonics. The components of $\vec{d}$ correspond to the first-order real-valued spherical harmonics f_i ($i = 1, 2, 3$). This can be seen when the components of $\vec{d}$ are written in a spherical coordinate system:

$$\vec{d} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{4\pi}{3}}\cdot f_1 \\ \sqrt{\frac{4\pi}{3}}\cdot f_2 \\ \sqrt{\frac{4\pi}{3}}\cdot f_3 \end{pmatrix}. \tag{68}$$

The components d_1, d_2 and d_3 are equal in this arrangement to the first-order functions f_1, f_2 and f_3 except of a normalization factor.

5.3.2 Expression of $\vec{d} \otimes \vec{d}$ through spherical harmonics. First the dyadic product $\vec{d} \otimes \vec{d}$ is formulated in a spherical coordinate system

$$\begin{aligned} \vec{d} \otimes \vec{d} &= \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \otimes \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} d_1^2 & d_1 d_2 & d_1 d_3 \\ d_2 d_1 & d_2^2 & d_2 d_3 \\ d_3 d_1 & d_3 d_2 & d_3^2 \end{pmatrix} \\ &= \begin{pmatrix} \sin^2\vartheta\cos^2\varphi & \sin^2\vartheta\sin\varphi\cos\varphi & \sin\vartheta\cos\vartheta\cos\varphi \\ \sin^2\vartheta\sin\varphi\cos\varphi & \sin^2\vartheta\sin^2\varphi & \sin\vartheta\cos\vartheta\sin\varphi \\ \sin\vartheta\cos\vartheta\cos\varphi & \sin\vartheta\cos\vartheta\sin\varphi & \cos^2\vartheta \end{pmatrix}. \end{aligned} \tag{69}$$

The following analysis shows that the off-diagonal elements of this matrix can each be expressed by a single second-order real-valued harmonic function, whereas the diagonal elements are linear combinations of the zeroth-order and several second-order real-valued spherical harmonic functions.

Off-diagonal elements. The element $d_1\,d_3$ is proportional to the function h_2 and the element $d_2\,d_3$ is proportional to the function h_3.

$$d_1 d_3 = \sqrt{\frac{4\pi}{15}} \cdot h_2$$

$$d_2 d_3 = \sqrt{\frac{4\pi}{15}} \cdot h_3$$

To see that the element $d_1\,d_2$ is proportional to the function h_5, h_5 must be rearranged,

$$h_5 = \sqrt{\frac{15}{16\pi}} \cdot \ \sin^2\vartheta \sin(2\varphi) = \sqrt{\frac{15}{4\pi}} \cdot \ \sin^2\vartheta \sin\varphi \cos\varphi, \tag{70}$$

$$d_1 d_2 = \sqrt{\frac{4\pi}{15}} \cdot h_5, \tag{71}$$

via the addition theorem: $\sin(2x) = 2 \cdot \sin(x)\cos(x)$.

Diagonal elements. The diagonal element $d_3^2 = \cos^2\vartheta$ can be expressed by a proper linear combination of $h_1 = \sqrt{\frac{5}{16\pi}} \cdot (3\cos^2\vartheta - 1) = \sqrt{\frac{45}{16\pi}} \cdot \cos^2\vartheta - \sqrt{\frac{5}{16\pi}}$ and $g_0 = \sqrt{\frac{1}{4\pi}}$, in such a way that the constant in h_1 compensate the constant in g_0.

$$h_1 + \sqrt{\frac{5}{4}} \cdot g_0 = \sqrt{\frac{45}{16\pi}} \cdot \cos^2\vartheta$$

$$d_3^2 = \sqrt{\frac{16\pi}{45}} \cdot \left(h_1 + \sqrt{\frac{5}{4}} \cdot g_0 \right) = \sqrt{\frac{16\pi}{45}} \cdot h_1 + \sqrt{\frac{4\pi}{9}} \cdot g_0$$

For the expression of d_1^2 and d_2^2 the function h_4 needs a closer examination. With $\cos(2x) = \cos^2 x - \sin^2 x$ the function h_4,

$$h_4 = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta \cos(2\varphi) = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta (\cos^2\varphi - \sin^2\varphi), \tag{72}$$

is rearranged in the two versions h_4^a and h_4^b,

$$h_4^a = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta (1 - 2 \cdot \ \sin^2\varphi) = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta - \sqrt{\frac{15}{4\pi}} \cdot \ \sin^2\vartheta \sin^2\varphi, \tag{73}$$

$$\sin^2\vartheta \sin^2\varphi = -\sqrt{\frac{4\pi}{15}} h_4^a + \sqrt{\frac{1}{4}} \sin^2\vartheta \tag{74}$$

$$h_4^b = \sqrt{\frac{15}{16\pi}} \sin^2\vartheta (-1 + 2 \cdot \ \cos^2\varphi) = -\sqrt{\frac{15}{16\pi}} \sin^2\vartheta + \sqrt{\frac{15}{4\pi}} \cdot \ \sin^2\vartheta \cos^2\varphi \tag{75}$$

$$\sin^2\vartheta \cos^2\varphi = \sqrt{\frac{4\pi}{15}} h_4^b + \sqrt{\frac{1}{4}} \sin^2\vartheta, \tag{76}$$

with the use of $\cos^2\varphi + \sin^2\varphi = 1$. The second term in h_4^a has the same form as the element d_1^2 and the second term in h_4^b equals d_2^2. It is left to show, that the term $\sin^2\vartheta$ can be expressed by the zero and the second-order real-valued spherical harmonic functions.

To see this the term $\sqrt{\frac{1}{4}}\sin^2\vartheta$ will be rearranged,

$$\sqrt{\frac{1}{4}}\sin^2\vartheta = \sqrt{\frac{1}{4}} - \sqrt{\frac{1}{4}}\cos^2\vartheta = -\sqrt{\frac{1}{36}}(3\cos^2\vartheta - 1) + \sqrt{\frac{1}{9}}, \tag{77}$$

$$= -\sqrt{\frac{4\pi}{45}}h_1 + \sqrt{\frac{4\pi}{9}}g_0 \tag{78}$$

which is a linear combination of h_1 and g_0.

Together d_1^2 and d_2^2 can be expressed through the following spherical harmonics:

$$d_1^2 = -\sqrt{\frac{4\pi}{15}}h_4 - \sqrt{\frac{4\pi}{45}}h_1 + \sqrt{\frac{4\pi}{9}}g_0$$

$$d_2^2 = \sqrt{\frac{4\pi}{15}}h_4 - \sqrt{\frac{4\pi}{45}}h_1 + \sqrt{\frac{4\pi}{9}}g_0$$

5.4 The weight matrix of the original MFA

To see that Eq (20) corresponds to Eq (4) as derived by Franz et al. in 2004 [11] for the original MFA, a change of the basis of the vector space is applied. In [11] the coordinates of each flow vector $\vec{p}_i$ were given with respect to different basis vectors $\vec{u}_i$, $\vec{v}_i$, spanning the tangential plane on the sphere for viewing direction $\vec{d}_i$. In the following we derive the transformation matrix to transform the coordinates of a vector given in the standard Euclidean basis vectors $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$ into the basis defined by $(\vec{u}_i, \vec{v}_i, \vec{d}_i)_{i=1,2,\ldots,N}$. With N optical flow vectors we define a $3 \times N$ dimensional vector space, which is the N-fold Cartesian product of the three basis vectors $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$.

All flow vectors $\vec{p}_i$ are represented now by a single stacked column vector $\vec{p}$,

$$\vec{p} = \begin{pmatrix} \vec{p}_1 \\ \vec{p}_2 \\ \vdots \\ \vec{p}_N \end{pmatrix}, \tag{79}$$

which has the dimension of $3 \times N$. In the same way the templates are written as

$$\vec{T}_A = \begin{pmatrix} \vec{T}_{1,A} \\ \vec{T}_{2,A} \\ \vdots \\ \vec{T}_{N,A} \end{pmatrix}, \tag{80}$$

where the index A stands for one of the six standard templates. The six templates $\vec{T}_A$ are combined to matrix $\hat{T}$ with six columns and $3 \times n$ rows:

$$\hat{T} = \begin{pmatrix} \vec{T}_{1,1} & \vec{T}_{1,2} & \cdots & \vec{T}_{1,6} \\ \vec{T}_{2,1} & \vec{T}_{2,2} & \cdots & \vec{T}_{2,6} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{T}_{N,1} & \vec{T}_{N,2} & \cdots & \vec{T}_{N,6} \end{pmatrix}. \tag{81}$$

The responses of the six model neurons $\vec{a}$ in this notation are

$$\vec{a} = \hat{T}^T \cdot \vec{p}, \tag{82}$$

where $\hat{T}^T$ stands for the transpose of $\hat{T}$. Together with the coupling matrix one obtains for the self-motion components (see Eq (20)):

$$\vec{\theta}^{\text{est}} = \hat{M}^{-1} \cdot \vec{a} = \hat{M}^{-1} \cdot \hat{T}^T \cdot \vec{p}. \tag{83}$$

The complete vector basis of the above templates and flow vectors is described by two indices $j = 1,2,3$ and $i = 1, 2, \ldots, N$ and has the form

$$B = \left(\vec{e}_{1,1}, \vec{e}_{2,1}, \vec{e}_{3,1}, \vec{e}_{1,2}, \vec{e}_{2,2}, \vec{e}_{3,2}, \ldots, \vec{e}_{1,N}, \vec{e}_{2,N}, \vec{e}_{3,N}\right), \tag{84}$$

where the $\vec{e}_{1,i}$, $\vec{e}_{2,i}$ and $\vec{e}_{3,i}$ are the same for each i. The basis vectors in the notation of Franz et al. [11] are

$$B' = \left(\vec{u}_1, \vec{v}_1, \vec{d}_1, \vec{u}_2, \vec{v}_2, \vec{d}_2, \ldots, \vec{u}_n, \vec{v}_n, \vec{d}_n\right), \tag{85}$$

where the viewing directions $\vec{d}_i$ supplemented the local vector spaces to a three dimensional space.

The new basis vectors B' can be expressed by the old B,

$$\vec{u}_i = u_{1,i}\vec{e}_{1,i} + u_{2,i}\vec{e}_{2,i} + u_{3,i}\vec{e}_{3,i},$$

$$\vec{v}_i = v_{1,i}\vec{e}_{1,i} + v_{2,i}\vec{e}_{2,i} + v_{3,i}\vec{e}_{3,i},$$

$$\vec{d}_i = d_{1,i}\vec{e}_{1,i} + d_{2,i}\vec{e}_{2,i} + d_{3,i}\vec{e}_{3,i},$$

where $u_{1,\,i}$ etc. is one component of $\vec{u}_1$. The transformation matrix $\hat{V}$ between the two bases with the dimension $[3 \times N] \cdot [3 \times N]$ is

$$\hat{V} = \begin{pmatrix} u_{1,1} & u_{2,1} & u_{3,1} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ v_{1,1} & v_{2,1} & v_{3,1} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ d_{1,1} & d_{2,1} & d_{3,1} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & u_{1,2} & u_{2,2} & u_{3,2} & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & v_{1,2} & v_{2,2} & v_{3,2} & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{1,2} & d_{2,2} & d_{3,2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & u_{1,N} & u_{2,N} & u_{3,N} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & v_{1,N} & v_{2,N} & v_{3,N} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & d_{1,N} & d_{2,N} & d_{3,N} \end{pmatrix}.$$

The matrix $\hat{V}$ transforms the coordinates of a vector given in basis B, Eq (84), into basis B', Eq (85). Eq (83) becomes

$$\begin{aligned} \vec{\theta}^{\text{est}} &= \hat{M}^{-1} \cdot \hat{T}^T \cdot \vec{p} = \left(\hat{T}^T \cdot \hat{T}\right)^{-1} \cdot \hat{T}^T \cdot \vec{p} \\ &= \left(\hat{T}^T \cdot \hat{V}^T \cdot \hat{V} \cdot \hat{T}\right)^{-1} \cdot \hat{T}^T \cdot \hat{V}^T \cdot \hat{V} \cdot \vec{p} \\ &= \left(\left(\hat{V} \cdot \hat{T}\right)^T \cdot \hat{V} \cdot \hat{T}\right)^{-1} \cdot \left(\hat{V} \cdot \hat{T}\right)^T \cdot \hat{V} \cdot \vec{p}. \end{aligned}$$

The term $\hat{V} \cdot \vec{p}$ is equivalent to $\vec{x}$ in Eq (4) from the original MFA of Franz et al. [11]. The term $\vec{x}$ combines all optic flow components of the local two dimensional vector spaces as spanned by the two local vectors $\vec{u}$ and $\vec{v}$. The third component is zero because $\vec{p}$ and $\vec{d}$ are orthogonal.

To see that $\hat{V} \cdot \hat{T}$ is $F = \begin{pmatrix} \frac{-\bar{\mu}_1 \vec{u}_1}{\vec{u}_1 \times \vec{d}_1}, \frac{-\bar{\mu}_1 \vec{v}_1}{\vec{v}_1 \times \vec{d}_1}, \ldots \end{pmatrix}^T$ the first row of $\hat{V} \cdot \hat{T}$ is scrutinized. The six components are

$$\vec{u}_1 \cdot \vec{T}_{1,1}, \vec{u}_1 \cdot \vec{T}_{1,2}, \vec{u}_1 \cdot \vec{T}_{1,3}, \vec{u}_1 \cdot \vec{T}_{1,4}, \vec{u}_1 \cdot \vec{T}_{1,5}, \vec{u}_1 \cdot \vec{T}_{1,6}.$$

The first three components are ($j = 1, 2, 3$)

$$-\mu_1 \cdot \vec{u}_1 \cdot \left(\vec{d}_1 \times \vec{e}_j \times \vec{d}_1\right) = -\mu_1 \cdot \vec{u}_1 \cdot \left(\vec{e}_j - \left(\vec{e}_j \cdot \vec{d}_1\right)\vec{d}_1\right) = -\mu_1 \cdot \vec{u}_{j,1},$$

and the second three components are

$$-\vec{u}_1 \cdot \left(\vec{e}_j \times \vec{d}_1\right) = \vec{e}_j \cdot \left(\vec{u}_1 \times \vec{d}_1\right) = \left(\vec{u}_1 \times \vec{d}_1\right)_j.$$

Acknowledgments

We are grateful to Jens Peter Lindemann for programming support. This study was supported by the Deutsche Forschungsgemeinschaft (DFG).

Author Contributions

Analyzed the data: SS WS ME. Wrote the paper: SS WS ME. Conceived and designed the theoretical project: SS WS ME. Performed the theoretical analysis: SS.

References

1. Koenderink JJ, Van Doorn AJ (1987) Facts on optic flow. Biological Cybernetics 56: 247–254. doi: 10.1007/BF00365219 PMID: 3607100
2. Hartley RI (1997) In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 19: 580–593. doi: 10.1109/34.601246
3. Nistér D (2004) An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 26: 756–770. doi: 10.1109/TPAMI.2004.17 PMID: 18579936
4. Luong QT, Faugeras OD (1997) Self-calibration of a moving camera from point correspondences and fundamental matrices. International Journal of Computer Vision 22: 261–289. doi: 10.1023/A:1007982716991
5. Baker S, Matthews I (2004) Lucas-Kanade 20 years on: a unifying framework. International Journal of Computer Vision 56: 221–255. doi: 10.1023/B:VISI.0000011205.11775.fd
6. Reichardt W (1961) Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In: Rosenblith WA, editor, Sensory Communication, MIT Press. pp. 303–317.
7. Borst A, Egelhaaf M (1993) Detecting visual motion: theory and models. Reviews of Oculomotor Research 5: 3–27. PMID: 8420555
8. Franz MO, Krapp HG (2000) Wide-field, motion-sensitive neurons and matched filters for optic flow fields. Biological Cybernetics 83: 185–97. doi: 10.1007/s004220000163 PMID: 11007295
9. Turin G (1960) An introduction to matched filters. IRE Transactions on Information Theory 6: 311–329. doi: 10.1109/TIT.1960.1057571
10. Borst A, Weber F (2011) Neural action fields for optic flow based navigation: a simulation study of the fly lobula plate network. PLOS ONE 6: e16303. doi: 10.1371/journal.pone.0016303 PMID: 21305019
11. Franz MO, Chahl JS, Krapp HG (2004) Insect-inspired estimation of egomotion. Neural Computation 16: 2245–2260. doi: 10.1162/0899766041941899 PMID: 15476600
12. Dahmen HJ, Franz MO, Krapp HG (2001) Extracting egomotion from optic flow: limits of accuracy and neural matched filters. In: Zanker JM, Zeil J, editors, Motion Vision, Springer. pp. 143–168.
13. Chipman JS (2011) Gauss-Markov theorem. In: International Encyclopedia of Statistical Science, Springer. pp. 577–582.
14. Hausen K, Egelhaaf M (1989) Neural mechanisms of visual course control in insects. In: Stavenga DG, Hardie RC, editors, Facets of Vision, Springer. pp. 391–424.
15. Krapp HG, Hengstenberg R, Egelhaaf M (2001) Binocular contributions to optic flow processing in the fly visual system. Journal of Neurophysiology 85: 724–34. PMID: 11160507
16. Borst A, Haag J (2002) Neural networks in the cockpit of the fly. Journal of Comparative Physiology A 188: 419–37. doi: 10.1007/s00359-002-0316-8
17. Egelhaaf M (2006) The neural computation of visual motion information. In: Warrant E, Nielsson DE, editors, Invertebrate Vision, Cambridge University Press. pp. 399–461.
18. Taylor G, Krapp HG (2007) Sensory systems and flight stability: what do insects measure and why? Advances in Insect Physiology 34: 231–316. doi: 10.1016/S0065-2806(07)34005-8
19. Egelhaaf M, Boeddeker N, Kern R, Kurtz R, Lindemann JP (2012) Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action. Frontiers in Neural Circuits 6: 108. doi: 10.3389/fncir.2012.00108 PMID: 23269913
20. Britten KH (2008) Mechanisms of self-motion perception. Annual Review of Neuroscience 31: 389–410. doi: 10.1146/annurev.neuro.29.051605.112953 PMID: 18558861
21. Chen A, DeAngelis GC, Angelaki DE (2010) Macaque parieto-insular vestibular cortex: responses to self-motion and optic flow. Journal of Neuroscience 30: 3022–3042. doi: 10.1523/JNEUROSCI.4029-09.2010 PMID: 20181599
22. Frost BJ, Wylie DR, Wang YC (1990) The processing of object and self-motion in the tectofugal and accessory optic pathways of birds. Vision Research 30: 1677–1688. doi: 10.1016/0042-6989(90)90152-B PMID: 2288083
23. Wylie DR (2000) Binocular neurons in the nucleus lentiformis mesencephali in pigeons: responses to translational and rotational optic flowfields. Neuroscience Letters 291: 9–12. doi: 10.1016/S0304-3940(00)01367-7 PMID: 10962141

24. Hausen K (1984) The lobula-complex of the fly: structure, function and significance in visual behaviour. In: Photoreception and Vision in Invertebrates, Springer. pp. 523–559.

25. Krapp HG (1999) Neuronal matched filters for optic flow processing in flying insects. International Review of Neurobiology 44: 93–120. doi: 10.1016/S0074-7742(08)60739-4

26. Egelhaaf M, Kern R, Krapp HG, Kretzberg J, Kurtz R, Warzecha AK (2002) Neural encoding of behaviourally relevant visual-motion information in the fly. Trends in Neurosciences 25: 96–102. doi: 10.1016/S0166-2236(02)02063-5 PMID: 11814562

27. Maddess T, Laughlin S (1985) Adaptation of the motion-sensitive neuron H1 is generated locally and governed by contrast frequency. Proceedings of the Royal Society of London B 225: 251–275. doi: 10.1098/rspb.1985.0061

28. Harris RA, O'Carroll DC, Laughlin SB (2000) Contrast gain reduction in fly motion adaptation. Neuron 28: 595–606. doi: 10.1016/S0896-6273(00)00136-7 PMID: 11144367

29. Reisenman C, Haag J, Borst A (2003) Adaptation of response transients in fly motion vision. I: Experiments. Vision Research 43: 1293–1309.

30. Borst A, Reisenman C, Haag J (2003) Adaptation of response transients in fly motion vision. II: Model studies. Vision Research 43: 1311–1324.

31. Liang P, Kern R, Egelhaaf M (2008) Motion adaptation enhances object-induced neural activity in three-dimensional virtual environment. The Journal of Neuroscience 28: 11328–11332. doi: 10.1523/JNEUROSCI.0203-08.2008 PMID: 18971474

32. Kurtz R, Egelhaaf M, Meyer HG, Kern R (2009) Adaptation accentuates responses of fly motion-sensitive visual neurons to sudden stimulus changes. Proceedings of the Royal Society B 276: 3711–3719. doi: 10.1098/rspb.2009.0596 PMID: 19656791

33. Franz MO, Chahl JS (2002) Insect-inspired estimation of self-motion. In: Biologically Motivated Computer Vision, volume 2525 of *Lecture Notes in Computer Science*, pp. 171–180.

34. Franz MO, Chahl JS (2003) Linear combinations of optic flow vectors for estimating self-motion—a real-world test of a neural model. Advances in Neural Information Processing Systems 15: 1–8.

35. Roberts R, Potthast C, Dellaert F (2009) Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 57–64.

36. Müller C (1966) Spherical Harmonics, volume 17 of *Lecture Notes in Mathematics*. Springer.

37. Werner D (2005) Funktionalanalysis. Springer, 5 edition.

38. Schilstra C, Hateren Jv (1999) Blowfly flight and optic flow. I. Thorax kinematics and flight dynamics. Journal of Experimental Biology 202: 1481–1490. PMID: 10229694

39. Hateren Jv, Schilstra C (1999) Blowfly flight and optic flow. II. Head movements during flight. Journal of Experimental Biology 202: 1491–1500. PMID: 10229695

40. Liang P, Kern R, Kurtz R, Egelhaaf M (2011) Impact of visual motion adaptation on neural responses to objects and its dependence on the temporal characteristics of optic flow. Journal of Neurophysiology 105: 1825–1834. doi: 10.1152/jn.00359.2010 PMID: 21307322

41. Ullrich TW, Kern R, Egelhaaf M (2015) Influence of environmental information in natural scenes and the effects of motion adaptation on a fly motion-sensitive neuron during simulated flight. Biology Open 4: 13–21. doi: 10.1242/bio.20149449

42. Boeddeker N, Dittmar L, Stürzl W, Egelhaaf M (2010) The fine structure of honeybee head and body yaw movements in a homing task. Proceedings of the Royal Society B 277: 1899–1906. doi: 10.1098/rspb.2009.2326 PMID: 20147329

43. Stürzl W, Zeil J (2007) Depth, contrast and view-based homing in outdoor scenes. Biological Cybernetics 96: 519–531. doi: 10.1007/s00422-007-0147-3 PMID: 17443340

44. Stürzl W, Boeddeker N, Dittmar L, Egelhaaf M (2010) Mimicking honeybee eyes with a 280 degree field of view catadioptric imaging system. Bioinspiration & Biomimetics 5: 036002. doi: 10.1088/1748-3182/5/3/036002

45. Kurtz R (2012) Adaptive encoding of motion information in the fly visual system. In: Frontiers in Sensing, Springer. pp. 115–128.

46. Egelhaaf M, Kern R, Lindemann JP (2014) Motion as a source of environmental information: a fresh view on biological motion computation by insect brains. Frontiers in Neural Circuits 8: 127. doi: 10.3389/fncir.2014.00127 PMID: 25389392

47. Schwegmann A, Lindemann JP, Egelhaaf M (2014) Depth information in natural environments derived from optic flow by insect motion detection system: a model analysis. Frontiers in Computational Neuroscience 8: 83. doi: 10.3389/fncom.2014.00083 PMID: 25136314

48. Baird E, Srinivasan MV, Zhang S, Cowling A (2005) Visual control of flight speed in honeybees. Journal of Experimental Biology 208: 3895–3905. doi: 10.1242/jeb.01818 PMID: 16215217

49. Baird E, Kornfeldt T, Dacke M (2010) Minimum viewing angle for visually guided ground speed control in bumblebees. The Journal of Experimental Biology 213: 1625–1632. doi: 10.1242/jeb.038802 PMID: 20435812

50. Kern R, Boeddeker N, Dittmar L, Egelhaaf M (2012) Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information. The Journal of Experimental Biology 215: 2501–2514. doi: 10.1242/jeb.061713 PMID: 22723490

51. Srinivasan M, Lehrer M, Kirchner W, Zhang S (1991) Range perception through apparent image speed in freely flying honeybees. Visual Neuroscience 6: 519–535. PMID: 2069903