

# Automatically calibrating the viewing direction of optic-flow sensors

Adrien Briod, Jean-Christophe Zufferey and Dario Floreano

**Abstract**—Because of their low weight, cost and energy consumption, optic-flow sensors attract growing interest in robotics for tasks such as self-motion estimation or depth measurement. Most applications require a large number of these sensors, which involves a fair amount of calibration work for each setup. In particular, the viewing direction of each sensor has to be measured for proper operation. This task is often cumbersome and prone to errors, and has to be carried out every time the setup is slightly modified.

This paper proposes an algorithm for viewing direction calibration relying on rate gyroscope readings and a recursive weighted linear least square estimation of the rotation matrix elements. The method only requires the user to realize random rotational motions of its setup by hand. The algorithm provides hints about the current precision of the estimation and what motions should be performed to improve it. To assess the validity of the method, tests were performed on an experimental setup and the results compared to a precise manual calibration. The repeatability of the gyroscope-based calibration process reached  $\pm 1.7^\circ$  per axis.

## I. INTRODUCTION

Optic-flow sensors such as the ones typically found in computer mice are increasingly used in robotics because they are light-weight and require very little processing power. Off-the-shelf mouse sensors with an adapted optics are often used in the literature [1], [2], [3], [4], [5], [6], along with other types of small vision chips [7], [8], [9]. These sensors, generally lighter than 1g, directly provide the optic-flow information in pixel counts per time unit and several of them can be interfaced by a single microcontroller, making them very attractive for low-weight/low-power applications.

Several applications in robotics require many optic-flow sensors to cover a wide field of view, such as optic-flow-based self-motion estimation or obstacle avoidance. In [10], [11], [12], several solutions for self-motion estimation are presented, which depend on wide field of view for good results. Experiments in [8] showed a flying robot estimate its own speed and yaw angle thanks to 6 optic-flow sensors looking in all lateral directions (see fig. 1(a)). In [1], 16 optic-flow sensors (see fig. 1(b)) allowed to estimate the precise rotations and direction of motion of a platform moving in 3D. In the case of depth or nearness measurement for obstacle avoidance, the field of view must be large enough so that the obstacles all around the robot can be detected. In the experiments from [2], a plane is able to avoid obstacles

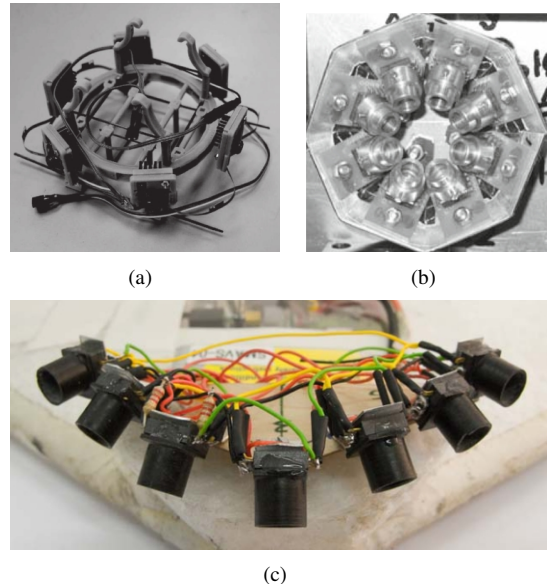


Fig. 1. Typical setups including several optic-flow mouse sensors used for self-motion estimation or obstacle avoidance. (a) shows the ring from [8] that comprises 6 sensors, (b) shows half of the setup from [1] that uses a total of 16 optic-flow sensors and (c) shows the front-end from [2] that contains 7 sensors (Image courtesy of the Autonomous Vehicle Laboratory at the University of Maryland, the Cognitive Neurosciences group at the University of Tübingen and the Laboratory of Intelligent Systems at EPFL, respectively).

thanks to the nearness measurements provided by 7 optic-flow sensors arranged around a  $90^\circ$  forward-facing view cone (see fig. 1(c)). Other experiments showed flying or ground robots avoid obstacles thanks to a centering behavior [8], [4] or a regulation of the optic-flow divergence [7].

All the applications presented before require to know in what direction the optic-flow sensors are looking. In addition, some applications are based on translational optic-flow [2], [3], [4], [5], [6], [7], [9], which means that the raw optic-flow is processed to cancel the rotational component using angular speed measurements (usually provided by on-board rate gyroscopes). Such processing (called de-rotation) requires the precise viewing direction of the sensor to be known. One solution to this problem is to align each sensor so that it faces a target in the surroundings and to compute the viewing directions from the feature's and sensor's positions and orientations. This method was used in [2] but proved to be imprecise and cumbersome.

As large numbers of optic-flow sensors are often used to cover the wide field of view needed in the applications described before, there is a demand for an easy procedure for viewing direction calibration. As described in [1], it is in

All authors are with the Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. Contact e-mail: adrien.briod@epfl.ch.

This work was supported by Armasuisse, competence sector Science + Technology for the Swiss Federal Department of Defense, Civil Protection and Sports.

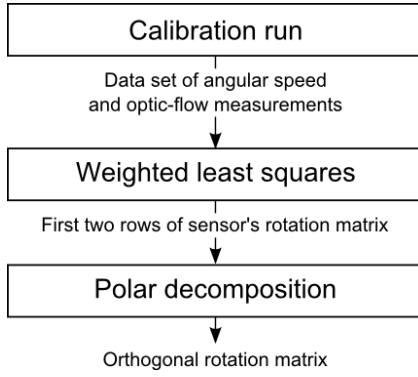


Fig. 2. Flow chart of the calibration process, that includes the following parts: 1) a calibration run where the setup is randomly rotated by hand, 2) a linear least squares algorithm that extracts some rotation matrix elements and 3) a polar decomposition that allows to find the closest rotation matrix that describes the sensor's orientation.

principle possible to recover elements of the sensor's rotation matrix by correlating sensor readings and known rotations. The approach in [1] proposes to strap the setup to a motor-controlled rotating shaft and make 3 measurements for each of the 3 rotation axes. The parameters are then obtained by comparing the integrated optic-flow to the motor steps. This method has the advantage that it provides an automatic calibration of several sensors at the same time. On the other hand, it requires a relatively complicated installation and a precise alignment of the setup with the rotating shaft for each of the 3 measurements.

We therefore propose a simpler calibration procedure that relies on a 3-axis angular rate sensor and random rotational motions realized by hand. The outcome of the algorithm is the sensors' orientations with respect to the gyroscope axes. This method does not require any special environment, the ability to access the sensor's captured image or any complicated installation. Finally, a rate gyroscope is a very simple and cheap sensor to integrate to a calibration setup, when it is not already part of the robotic platform.

This paper is organized as follows: section II presents the algorithm, section III and IV present the experimental setup and the manual calibration used to validate the concept, and finally the results are shown in section V.

## II. CALIBRATION ALGORITHM

We present in this section an algorithm that estimates the orientation of an optic-flow sensor with respect to the gyroscope reference frame. The three sub-chapters of this section correspond of the three main steps of the algorithm, shown in the flow-chart of fig. 2.

### A. Calibration run

The optic-flow vector  $\mathbf{p}$  is given by the following equation [13]:

$$\mathbf{p} = -\boldsymbol{\omega} \times \mathbf{d} - \frac{\mathbf{v} - (\mathbf{v} \cdot \mathbf{d})\mathbf{d}}{D} \quad (1)$$

Where  $\mathbf{d}$  is a unit vector pointing toward the viewing direction of the sensor,  $\boldsymbol{\omega}$  the rotation vector,  $\mathbf{v}$  the translational

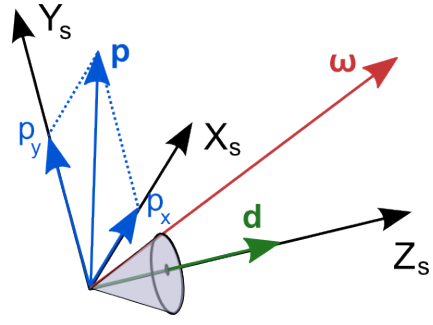


Fig. 3. The three vectors involved in the optic-flow equation  $\mathbf{p} = -\boldsymbol{\omega} \times \mathbf{d}$  when no translational motion is realized ( $\mathbf{v} = \mathbf{0}$ ). Expressing these vectors in the sensor frame  $X_s, Y_s, Z_s$  (with the sensor pointing in the direction along  $Z_s$ ) allows to greatly simplify this equation.

speed and  $D$  the distance to the object seen by the sensor.  $\mathbf{p}$  can be expressed in two parts, namely the 'rotational' optic-flow  $\mathbf{p}_{\text{rot}}$  and 'translational' optic-flow  $\mathbf{p}_{\text{trans}}$ :

$$\mathbf{p} = \mathbf{p}_{\text{rot}} + \mathbf{p}_{\text{trans}} \quad (2)$$

where:

$$\mathbf{p}_{\text{rot}} = -\boldsymbol{\omega} \times \mathbf{d} \quad (3)$$

$$\mathbf{p}_{\text{trans}} = -\frac{\mathbf{v} - (\mathbf{v} \cdot \mathbf{d})\mathbf{d}}{D} \quad (4)$$

We assume that all the sensors to be calibrated are on a single vantage point, and that only pure rotations around this point are realized during the calibration run. This assumption remains valid if the sensor positions are not exactly centered on the rotation point but the distance to the surroundings is relatively large while performing the rotations. We can therefore set  $\mathbf{v} = \mathbf{0}$ , implying  $\mathbf{p}_{\text{trans}} = \mathbf{0}$ , reducing equation 1 to:

$$\mathbf{p} = \mathbf{p}_{\text{rot}} = -\boldsymbol{\omega} \times \mathbf{d}, \quad (5)$$

which can be expressed in the sensor coordinate system ( $X_s, Y_s, Z_s$ ) shown in fig. 3:

$$\mathbf{p}_s = -\boldsymbol{\omega}_s \times \mathbf{d}_s \quad (6)$$

where :

$$\mathbf{p}_s = \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}, \quad \mathbf{d}_s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

$$\text{and } \boldsymbol{\omega}_s = \mathbf{R}_s \boldsymbol{\omega} = \begin{bmatrix} r_{s,11} & r_{s,12} & r_{s,13} \\ r_{s,21} & r_{s,22} & r_{s,23} \\ r_{s,31} & r_{s,32} & r_{s,33} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

with  $\mathbf{R}_s$  being the rotation matrix from the reference frame (or gyroscope frame) to the sensor frame. Note that  $\mathbf{d}_s$  is normal to the sensor image plane, therefore the optic-flow vector  $\mathbf{p}_s = -\boldsymbol{\omega}_s \times \mathbf{d}_s$  is parallel to the image plane and its third element is zero.  $p_x$  and  $p_y$  are the two components of the optic-flow vector that are typically given by optic-flow sensors. Note also that vector  $\mathbf{d}$  is collinear with  $Z_s$  and can thus be expressed as  $\mathbf{d} = [r_{s,31} \ r_{s,32} \ r_{s,33}]^T$ .

Equation 6 can be rewritten as:

$$\begin{aligned} \begin{bmatrix} p_x \\ p_y \end{bmatrix} &= \begin{bmatrix} -\omega_{s,2} \\ \omega_{s,1} \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} -p_x \\ p_y \end{bmatrix} &= \begin{bmatrix} r_{s,21} & r_{s,22} & r_{s,23} \\ r_{s,11} & r_{s,12} & r_{s,13} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \end{aligned} \quad (8)$$

Assuming a calibration run providing  $N$  measurements  $[p_x \ p_y \ \omega_1 \ \omega_2 \ \omega_3]_{1..N}$ , it is clear from equation 8 that the coefficients  $r_{s,1:2,1:3}$  can be estimated, and thus the orientation of the sensor too. The six coefficients  $r_{s,1:2,1:3}$  describe the orientation of the sensor in an over-constrained manner, as only three independent variables are necessary to express an orientation. Non-linear techniques can be used to fit quaternions or Euler angles to the dataset, but trials showed inconsistent results and recurrent failures to converge in the case of non-linear least squares. This is why we suggest to use two independent linear least square regressions to find the best fit for each row of  $r_{s,1:2,1:3}$  independently and to constrain the estimates to a rotation matrix at a later stage. Using linear least squares allows to guarantee a convergence toward a solution, and provides a useful feedback during the calibration run that tells which is the axis that should be rotated to decrease the remaining uncertainty of each rotation matrix component.

### B. Iterative linear weighted least squares

The aim is to solve the over determined observation equation  $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{u}$  where  $\mathbf{x}$  is the unknown parameter vector of length  $l$ ,  $\mathbf{z} = [z_1; \dots; z_N]$  the vector of all  $N$  measurements,  $\mathbf{H} = [\mathbf{h}_1; \dots; \mathbf{h}_N]$  the  $N \times l$  design matrix that describes the linear relation between measurements and parameters and  $\mathbf{u}$  the measurement noise that is assumed to follow a Gaussian distribution with zero mean:  $\mathbf{u} \sim N(\mathbf{0}, \mathbf{Q})$  (where  $\mathbf{Q}$  is a covariance matrix). The weighted least squares estimate is given by [14]:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{Q}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Q}^{-1} \mathbf{z} \quad (9)$$

However, we suggest to use the iterative form of the linear weighted least squares algorithm, solving for  $\hat{\mathbf{x}}_n$  iteratively as the measurements  $z_n$  are sampled. The iterative form of the algorithm has the advantage that it provides real-time feedback to the user about the on-going parameter estimation and is more adapted to microcontroller implementation.

Assuming that all measurements are independent ( $\mathbf{Q}$  is diagonal), the iterative expression of the algorithm is expressed as [14]:

$$\mathbf{k}_n = \mathbf{P}_{n-1} \mathbf{h}_n^T (\mathbf{h}_n \mathbf{P}_{n-1} \mathbf{h}_n^T + q_n)^{-1} \quad (10)$$

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_{n-1} + \mathbf{k}_n (z_n - \mathbf{h}_n \hat{\mathbf{x}}_{n-1}) \quad (11)$$

$$\mathbf{P}_n = (\mathbf{I} - \mathbf{k}_n \mathbf{h}_n) \mathbf{P}_{n-1} \quad (12)$$

Where  $\mathbf{P}$  is the covariance matrix of  $\hat{\mathbf{x}}$ , and  $\mathbf{k}_n$  an internal gain that determines the relative influence of the previous estimate  $\hat{\mathbf{x}}_{n-1}$  and the new measurement  $z_n$ . Each new measurement  $z_n$  is weighted thanks to  $q_n$ , the diagonal elements of  $\mathbf{Q}$ , that describe the noise of each measurement

as a variance  $\sigma_{z_n}^2$ . The state is usually initialized at  $\hat{\mathbf{x}}_0 = \mathbf{0}$  with a covariance matrix  $\mathbf{P}_0 = \mathbf{I}$ .

To solve for the parameters of equation 8, two linear least square estimations have to be applied to the following variables:

$$\left\{ \begin{array}{l} \hat{\mathbf{x}} = \begin{bmatrix} \hat{r}_{s,21} \\ \hat{r}_{s,22} \\ \hat{r}_{s,23} \end{bmatrix}^T \\ \mathbf{h}_n = \begin{bmatrix} \omega_{1,n} \\ \omega_{2,n} \\ \omega_{3,n} \end{bmatrix} \\ z_n = -p_{x,n} \end{array} \right. \text{ and } \left\{ \begin{array}{l} \hat{\mathbf{x}} = \begin{bmatrix} \hat{r}_{s,11} \\ \hat{r}_{s,12} \\ \hat{r}_{s,13} \end{bmatrix}^T \\ \mathbf{h}_n = \begin{bmatrix} \omega_{1,n} \\ \omega_{2,n} \\ \omega_{3,n} \end{bmatrix} \\ z_n = p_{y,n} \end{array} \right. \quad (13)$$

The real-time feedback of the calibration is obtained by looking at  $\mathbf{P}$ . The diagonal elements  $P_{ii,n} = \sigma_{r_{1:2,i}}^2$  describe the remaining error on the parameter  $\hat{\mathbf{x}}$  at step  $n$  and their evolution gives an idea about the error remaining on the estimation of each  $r_{s,1:2,i}$ . If the error  $\sigma_{r_{1,i}}^2$  or  $\sigma_{r_{2,i}}^2$  remains high, it means that the algorithm needs more data with rotations around the  $i^{\text{th}}$  axis. The estimation can be stopped once all variances  $\sigma_{r_{1:2,1:3}}^2$  go below a threshold, or after a fixed number of iterations. Typically, the calibration run should contain data of rotations around all axes to produce good estimates. The measurement standard deviation  $\sigma_{z_n}$  corresponds here to the standard deviation of the optic-flow measurements  $\sigma_{p_n}$ .

### C. Polar decomposition

The vectors of a rotation matrix must, by definition, be perpendicular to each other and have unity norm. The estimated vectors  $\hat{\mathbf{r}}_{s,1}$  and  $\hat{\mathbf{r}}_{s,2}$  are affected by noise and thus don't fulfill these constraints.

In a first place, we suggest to reconstruct a non-orthogonal matrix  $\mathbf{A}$  such that:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{bmatrix}, \mathbf{a}_1 = \frac{\hat{\mathbf{r}}_{s,1}}{\|\hat{\mathbf{r}}_{s,1}\|}, \mathbf{a}_2 = \frac{\hat{\mathbf{r}}_{s,2}}{\|\hat{\mathbf{r}}_{s,2}\|} \text{ and } \mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2$$

Then, we use the polar decomposition to reconstruct the orthogonal matrix  $\hat{\mathbf{R}}_s$  closest to  $\mathbf{A}$ , as proposed in [15]:

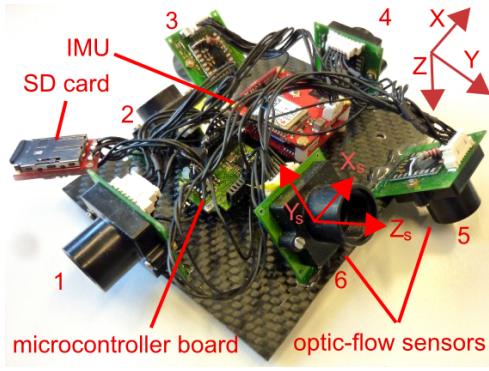
$$\hat{\mathbf{R}}_s = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-\frac{1}{2}} \quad (14)$$

$\hat{\mathbf{R}}_s$  is the calibrated rotation matrix describing the orientation of the sensor with respect to the gyroscope axes.

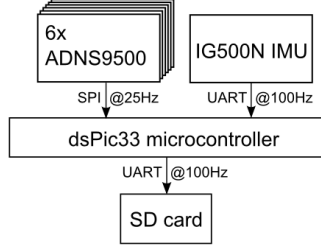
## III. SETUP

To test the proposed calibration procedure, the setup shown in fig. 4 is used. We will automatically calibrate all six ADNS9500<sup>1</sup> mouse chip sensors that are fixed to the platform and point in various directions. The setup is designed for experiments on self-motion extraction, for which the viewing direction of each sensor has to be known. The wide field of view and numerous sensors make it a perfect setup to gather experimental data for the research presented in this paper. The optic-flow measurements are

<sup>1</sup>from Avago: [www.avagotech.com](http://www.avagotech.com)



(a)



(b)

Fig. 4. Sensor head that will be used in this paper to validate the calibration algorithm. It is equipped with six ADNS9500 sensors, an IG500-N 9-DOF IMU, and a microcontroller board for SD data logging. Fig. (a) shows the six optic-flow sensors mounted on the platform and fig. (b) shows a schematics of the electrical wiring.

recorded at 25Hz, while rate gyroscope data from a 9-DOF IG-500N IMU<sup>2</sup> is recorded at 100Hz on a microSD card.

Each ADNS9500 sensor samples a 30x30 pixel image up to 11,750 times/s and increments the image displacement counts in a 16 bits register (which is reset to zero when it is read). The sensors are equipped with Philips CAY046 lenses that have a 4.6mm focal length, providing a 20° field of view, chosen to maximize the amount of light the sensor receives to enable indoor use in low-light conditions (tested down to 25 lumen). The exposure time is automatically adjusted by the chip to adapt to the external light. The ADNS9500 provides the 2D optic-flow vector  $\mathbf{p}_{\text{raw}}$  corresponding to the image displacement on the sensor in counts per time unit. Some other information is provided, such as the squal value which gives the number of valid features that are seen by the sensor and is therefore a good indicator of the quality of the optic-flow measurement. To convert the optic-flow counts to  $\text{rad/s}$ , the following formula can be used:

$$\mathbf{p} = \frac{\mathbf{p}_{\text{raw}}}{K \cdot f \cdot \Delta t \cdot \text{Res}} \quad (15)$$

where  $K$  is a chip-specific constant ( $0.694\text{rad}^{-1}$  in the case of the ADNS9500, see section V for a method to measure this constant),  $f$  is the focal length,  $\Delta t$  is the sampling period ( $40\mu\text{s}$  in our case) and  $\text{Res}$  is the sensor's resolution ( $1.6 \cdot 10^5$  counts/m in our case). The resolution of the optic-flow

<sup>2</sup>from SBG: [www.sbg-systems.com](http://www.sbg-systems.com). IMU includes a 3-axis accelerometer, 3-axis rate gyroscope and 3-axis magnetometer

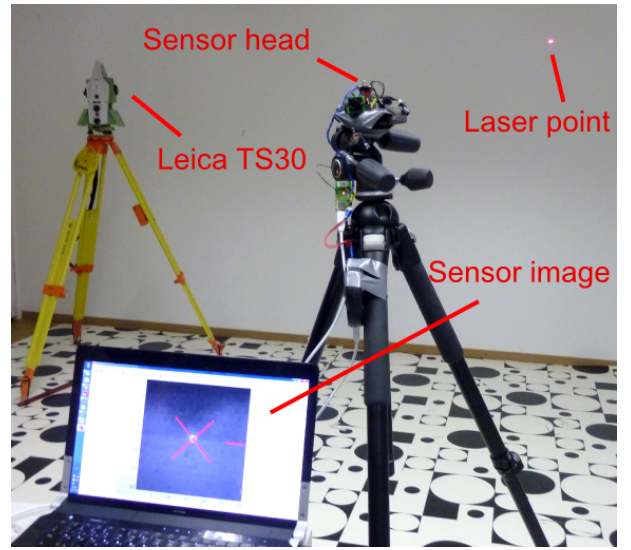


Fig. 5. Setup used for the manual calibration of the sensor head: The laser of a Leica TS30 precision tracking system is pointed at a white wall. It is then moved so as to align the laser with two spots on the sensor image. The two positions of the laser pointer are then recorded in order to geometrically reconstruct the sensor orientation (the position and orientation of the sensor head are recorded as well).

measurements is therefore of  $0.049\text{rad/s}$ . The resolution can be improved by decreasing the sampling rate (at the cost of slower response) or increasing the focal length (at the cost of brighter environment requirements).

The sensor can also be used in camera mode, and send each pixel value to make possible an image reconstruction. This is useful to adjust the focus or for other calibration tasks.

#### IV. MANUAL CALIBRATION

To validate the proposed automatic calibration method, its results are compared to a more conventional manual calibration. As explained in the introduction, manually calibrating the viewing direction of optic-flow sensors is cumbersome and often imprecise. It consists in looking at the image captured by each sensor to infer where it is pointing at from geometric calculations. This section presents how we performed the manual calibration to be the most precise possible in order to compare it to our automatic calibration procedure.

For this task, the Leica TS30<sup>3</sup> tracking system is used. This expensive piece of equipment is usually used for environment or construction monitoring and is able to measure precisely ( $< 1\text{mm}$  error) the 3D position of any feature thanks to a laser mounted on a pan-tilt head.

The sensor head of fig. 4 was installed in the middle of a room, its position and orientation measured respectively by the TS30 and the on-board IMU. The laser of the TS30 was then aligned with the center and the right border of the image of each optic-flow sensor and these positions recorded (see fig. 5). The orientation of each sensor with

<sup>3</sup>from Leica Geosystems: [www.leica-geosystems.com](http://www.leica-geosystems.com)



respect to the ENU (East-North-Up) frame was computed from these position measurements. These orientations were then expressed in the IMU frame.

The good precision of the Leica TS30 allows to minimize the geometry errors. However, the alignment of the optic-flow sensor image with the laser point remains difficult because of the low resolution of the camera. Considering all error sources, we estimate the precision of this manual calibration technique to be around  $\pm 2^\circ$ . Needless to say that this error would be much higher without the use of the Leica TS30.

## V. EXPERIMENTAL RESULTS

Ten calibration sequences were logged on the SD card for offline analysis on Matlab. The sensor head was rotated manually around many different axes several times during each log, each between 1 and 2 minutes long. The experiments were performed in a standard well-lit office space, without any modification to the surroundings.

Before starting the calibration process, it is important that the angular rate readings and optic-flow readings are correctly synchronized. To achieve that, the gyroscope readings are first integrated between each new optic-flow reading since they are not recorded at the same frequency. Then, to analyze if there is a delay between the angular speed and optic-flow, a cross-correlation was performed between all angular speeds and optic-flow readings. A 20ms delay was found between the angular rate and optic-flow readings, which is taken into account in the rest of the calibration procedure.

The optic-flow standard deviation  $\sigma_{p_n}$  (see section II-B) is not easy to determine for the ADNS9500 readings. Typically the precision of the optic-flow value depends on the environment, the luminosity, etc. We observed a correlation between the squal value (number of features tracked by the mouse chip, see chapter III) and the quality of the optic-flow readings. Hence we suggest to use  $\sigma_{p_n} = \frac{k}{\text{squal}}$  where  $k$  is a constant chosen so that the estimation error converges at the desired rate. This expression for  $\sigma_{p_n}$  implies that the measurements where the squal is higher will have more impact in the estimation process.  $k$  was set to 100 after a quick tuning procedure and the obtained convergence rate is reasonable (the standard deviation - or error - of the state estimation decreases by 90% in less than 60s for most logs). Finally, to avoid spurious data to influence the estimation, all measurements with  $\text{squal} < 50$  are discarded, which showed to be a good way of rejecting most outliers.

An example of a 2 minutes calibration procedure is shown in figure 6. It can be seen that the coefficient errors (or standard deviations) decrease as soon as a rotation is performed along the corresponding axis. For example, a rotation around the  $x$  axis will provide valuable information to estimate  $r_{11}$  and  $r_{21}$  so their corresponding errors will decrease (e.g. at  $t = 20s$  in fig. 6 a rotation around the  $x$  axis is started and provokes a drop in the error of  $r_{11}$  and  $r_{21}$ ). In the example of fig. 6, all coefficient errors are fairly low after 60s (standard deviation below 0.1). Note that if the constant  $K$  described in section III is correct, the norm of  $\hat{r}_{s,1}$  and

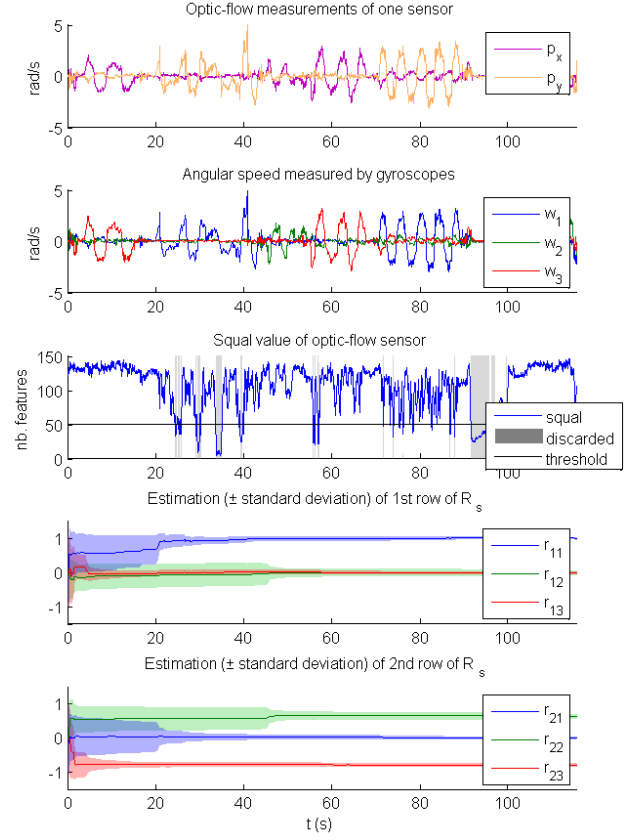


Fig. 6. Example of a 2 minutes calibration procedure where the sensor head is rotated by hand multiple times around all axes. The standard deviations of the matrix elements is plotted in transparency and decreases as long as sufficient measurements are provided for the estimation and the squal value is higher than the threshold. The standard deviations allow to know which types of motion are missing in order to achieve the estimation (see text for details) and can therefore provide a useful feedback to the user during the calibration process. For example, it can be seen that the error on  $\hat{r}_{11}$  and  $\hat{r}_{21}$  decreases when the sensor is rotated around the  $x$  axis (when  $\omega_1$  varies)

$\hat{r}_{s,2}$  should be 1. If  $K$  is unknown or wrong, it can be found by using first  $K = 1$  and then averaging the norms of  $\hat{r}_{s,1}$  and  $\hat{r}_{s,2}$ :  $K = 0.5(|\hat{r}_{s,1}| + |\hat{r}_{s,2}|)$ . The polar decomposition is then used to obtain the final rotation matrix. The elements of this matrix can directly be used to de-rotate an optic-flow signal thanks to equations 6 and 7.

To illustrate the results of the calibration procedure, the orientations of the 6 sensors estimated thanks to the 10 datasets are plotted in 3D in fig. 7. The calibrated orientations of the sensors are then statistically compared to the manual calibrations obtained in section IV. To this end, the triplet of Euler angles corresponding to the small rotation errors between the automatically and manually calibrated orientations are then analyzed. Fig. 8 shows the boxplot representations of these Euler angles. The average root mean square (RMSE) and the average standard deviation (STD) of each error angle are reported below:

$$\overline{RMSE} = 2.38^\circ \quad (16)$$

$$\overline{STD} = 1.79^\circ \quad (17)$$

Comparison between the automatically and manually calibrated orientations

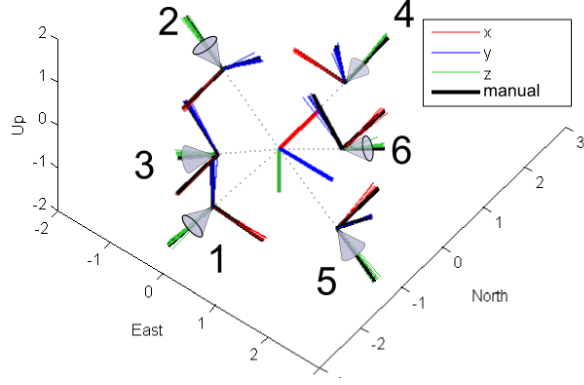


Fig. 7. The viewing direction of the six sensors of the setup from fig. 4 were calibrated using 10 different datasets. These orientations (in red, blue, green) are plotted next to the manually calibrated orientations (in black) for comparison. It can be seen that the automatically estimated sensor orientations are very close to the orientations obtained manually.

Error between the automatically and manually calibrated orientations as Euler triplets

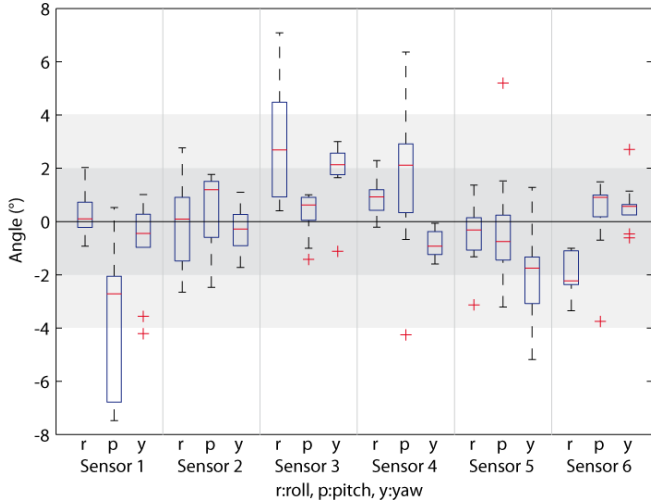


Fig. 8. Boxplot representation of the angular errors between the automatically and manually calibrated sensor viewing directions (expressed in Euler angle triplets). The boxplots show the results from the ten calibration runs. The baseline corresponds to the manual calibration. It can be seen that most error angles remain inside a  $\pm 2^\circ$  margin, which corresponds to the uncertainty of the manual calibration (see chapter IV).

The fact that the RMSE is larger than the average STD shows that the error angles are not distributed around zero, which can also be seen on fig. 8. However, the baseline being the manual orientations, this error can be explained by the uncertainty of the manual calibration (estimated to be around  $\pm 2^\circ$ , as explained in chapter IV). On the other hand, the low standard deviation of the automatic calibration results ( $1.79^\circ$ ) shows that the calibration procedure is very repeatable. These results indicate that the automatic calibration procedure is at least as precise as the manual one. Without having the data to prove it (we would need an extremely precise ground-truth to do so), we suggest that this might even indicate that the automatic calibration is more reliable than the manual calibration performed in chapter IV. Overall, this shows

that the proposed calibration procedure gives very satisfying results while providing at the same time a much simpler and reliable solution to the viewing direction calibration problem (especially when high-tech equipment is not available for manual calibration).

## VI. CONCLUSION

The calibration procedure described in this paper is a very useful tool for all roboticists working with optic-flow sensors. Not only is it very simple, fast, does not require any special setup or to access the sensor image, but it also proved to be very reliable, with a repeatability of  $\pm 1.7^\circ$  per axis in our particular case.

## VII. ACKNOWLEDGMENTS

The authors thank the anonymous reviewers who provided very valuable input and helped improve the quality of the paper. Many thanks to Adam Klapotcz, who designed the microcontroller board used in the experiments. Finally, thanks go to Claudio Matussi who helped reviewing the paper.

## REFERENCES

- [1] H. Dahmen, A. Millers, and H. A. Mallot, "Insect-Inspired Odometry by Optic Flow Recorded with Optical Mouse Chips," in *Flying Insects and Robots*, D. Floreano, J.-C. Zufferey, M. V. Srinivasan, and C. Ellington, Eds. Berlin, Heidelberg: Springer, 2009.
- [2] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous Robots*, vol. 27, no. 3, pp. 201–219, 2009.
- [3] B. Barber, S. Griffiths, T. McLain, and R. Beard, "Autonomous landing of miniature aerial vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 5, pp. 770–784, 2007.
- [4] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, "Obstacle and terrain avoidance for miniature aerial vehicles," *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, pp. 213–244, 2007.
- [5] A. Rodriguez, E. Andersen, J. Bradley, and C. N. Taylor, "Wind estimation using an optical flow sensor on a miniature air vehicle," *AIAA Conference on Guidance, Navigation, and Control*, 2007.
- [6] J. Kim and G. Brambley, "Dual Optic-flow Integrated Navigation for Small-scale Flying Robots," in *Proc. of Australasian Conference on Robotics and Automation, Brisbane, Australia*, 2007.
- [7] J.-c. Zufferey, A. Klapotcz, A. Beyeler, J.-d. Nicoud, and D. Floreano, "A 10-gram Microflyer for Vision-based Indoor Navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Oct. 2006, pp. 3267–3272.
- [8] S. Humbert, J. K. Conroy, C. W. Neely, and G. L. Barrows, "Wide-Field Integration Methods for Visuomotor Control," in *Flying Insects and Robots*, D. Floreano, J.-C. Zufferey, M. V. Srinivasan, and C. Ellington, Eds. Springer Berlin Heidelberg, 2009, ch. 5.
- [9] G. L. Barrows, J. S. Chahl, and M. Srinivasan, "Biomimetic Visual Sensing and Flight Control," in *Proc. Bristol UAV Conf*, 2002, pp. 159–168.
- [10] J. Gluckman and S. Nayar, "Ego-motion and omnidirectional cameras," in *Sixth International Conference on Computer Vision*. IEEE, 1998, pp. 999–1005.
- [11] Y.-S. Chen, L.-G. Liou, Y.-P. Hung, and C.-S. Fuh, "Three-dimensional ego-motion estimation from motion fields observed with multiple cameras," *Pattern Recognition*, vol. 34, no. 8, pp. 1573–1583, Aug. 2001.
- [12] M. Franz, J. S. Chahl, and H. G. Krapp, "Insect-inspired estimation of egomotion," *Neural computation*, vol. 16, no. 11, pp. 2245–60, Nov. 2004.
- [13] J. Koenderink and A. Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, no. 4, pp. 247–254, 1987.
- [14] E. M. Mikhail and F. E. Ackermann, *Observations and least squares*. IEP (New York), 1976.
- [15] N. J. Higham, "Matrix nearness problems and applications," *Applications of Matrix Theory*, pp. 1–27, 1989.