Bioinspired Direct Visual Estimation of Attitude Rates with Very Low Resolution Images using Deep Networks*

M. Mérida-Floriano[1], F. Caballero[1], D. Acedo[1], D. García-Morales [2], F. Casares [2] and L. Merino[1]

***Abstract*— In this work we present a bioinspired visual system sensor to estimate angular rates in unmanned aerial vehicles (UAV) using Neural Networks. We have conceived a hardware setup to emulate *Drosophila*'s ocellar system, three simple eyes related to stabilization. This device is composed of three low resolution cameras with a similar spatial configuration as the ocelli. There have been previous approaches based on this ocellar system, most of them considering assumptions such as known light source direction or a punctual light source. In contrast, here we present a learning approach using Artificial Neural Networks in order to recover the system's angular rates indoors and outdoors without previous knowledge. A classical computer vision based method is also derived to be used as a benchmark for the learning approach. The method is validated with a large dataset of images (more than half a million samples) including synthetic and real data. The source code of the algorithms and the datasets used in this paper have been released in an open repository.**

I. INTRODUCTION

The potential of Micro-Aerial Vehicles (MAVs, aerial vehicles between 0.1 and 0.5 meters and 0.1-0.5 kg. in mass) has been shown by different research results in the last years [1], [2], and even new commercial systems, like for instance the Skydio system[1].

Given the limited payload of such vehicles, vision systems are a preferred solution for micro-UAV perception, as cameras are low-power passive sensors and can be made small. Vision-based procedures have been proposed for odometry [3], [4], [5], localization [6], [7], [8], mapping [9] and navigation [10]. All these vision systems are typically founded on feature-based methods and are computationally demanding, requiring large onboard processing power, though.

On the other hand, it is very impressive the maneuverability that flying insects like flies can achieve with their very small payloads. This ability of flying insects is the reason why several authors have studied bio-inspired solutions for the development of new sensors and/or actuators for micro aerial vehicles, like [11], [12], [13]. However, most of these works have been devoted to the development of artificial compound-like eyes. The ocelli system found on the forehead of most insects is also an interesting biological inspiration for vision systems. The ocelli are small, structurally simple camera type eyes (see Fig. 1). Their large lens makes them extremely sensitive light sensors while, by focusing beyond the retina, blur the image. They are capable of very quick visual processing to trigger swift stabilization reflexes [14], [15]. And several approaches inspired on the ocellar system have been also proposed in the literature.

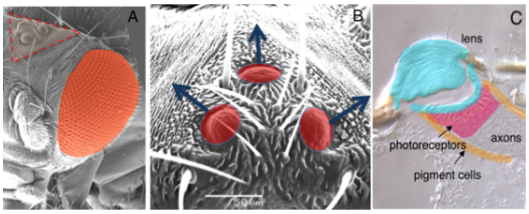

Fig. 1. The *Drosophila* ocelli system. A. Scanning electron microscope (SEM) micrograph of an adult *Drosophila* head. The three ocelli are located on the dorsal head (triangle). The lateral compound eyes are pseudo-colored (red). B: Higher magnification SEM of the dorsal head showing the ocellar lenses. Each ocellus (one anterior and two or lateral) surveys a different region of the space. C. Crossection through a lateral ocellus.

In [16], the authors present a device based on the ocellar system using 8 photodiode pairs. The outputs of the photodiodes are used to obtain a reference signal for the stabilization. In [17], the ocellar sensors are modeled as sensors providing an estimation of time derivatives of scalar luminance values. Then, they derive a linear relation between the ocelli inputs and robot states. They conclude that there is a relation between the ocellar input and roll and pitch angular rates, as well as heave rate. This is then used to develop an analog angular rate sensor based on photodiodes. A similar sensing approach is followed in [18], where an ocelli-inspired flight stabilization system has been implemented on a bee-sized flying robot. The addition of a torque controller based on a proportional feedback to the estimated angular velocity has sufficed to stabilize the upright orientation of the system.

The previous ocelli-inspired approaches disregard the spatial information of the ocelli simple eyes, and in some cases some assumptions are needed with respect to the light source direction [18]. On the contrary, in [19], linear receptive fields are optimized from data to obtain the relation between the sensorial input of simulated insect-like eyes and attitude angles. We have shown in our previous work [20] that Deep Neural Networks (DNNs), which have been used for applications of robot estimation control [21], [22], can be effective on estimating angular rates from low-resolution visual inputs in a camera system emulating the ocelli.

The paper contributes by presenting a method for the

*This work was supported by MINECO (Spain) grant OCELLIMAV (TEC-61708-EXP)

[1]M. Mérida-Floriano, F. Caballero, D. Acedo and L. Merino are with School of Engineering, Universidad Pablo de Olavide, Seville, Spain {mmerflo,fcaballero,dacegom,lmercab} at upo.es

[2]D. García-Morales and F. Casares are with Department of Gene Regulation and Morphogenesis, CABD, CSIC and Universidad Pablo de Olavide, Seville, Spain {dgarmor,fcasfer} at upo.es

[1]https://www.skydio.com/

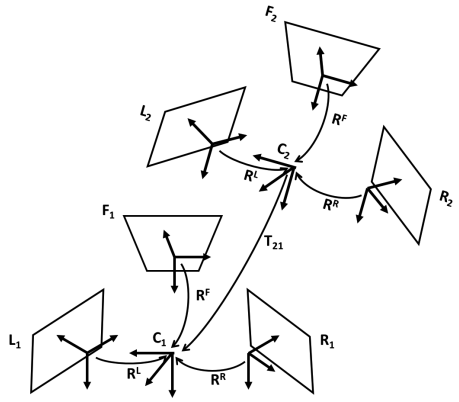

Fig. 2. Representation of the ocelli as a multi-camera system where L, R, and F stand for left, right and front photoreceptors respectively.

estimation of angular rates from visual inputs with application to UAVs. A preliminary study of the feasibility of such approach was presented in [20], where different architectures and input structures were tested in a limited scenario. The paper departs from that work by considering a completely network structure (CNNBiGRU), that includes temporal information and that clearly improves the accuracy of its predecessor. This method allows direct estimation of the angle rates using very low resolution images (10x8 pixels). Furthermore, in order to evaluate the generalization of the method, a much larger dataset, including simulated and real-images under different scenarios and lighting conditions is considered. Additionally, a second direct method based on perspective geometry and non-linear optimization is also presented to benchmark the proposed DNN approach in similar circumstances. The paper also includes experiments to validate the use of the proposed method to estimate the random BIAS present in gyroscopes without the use of accelerometers and magnetometers. Finally, the source code and the datasets used for validation have been released to the public.

The paper is organized as follows. Next section describes a hardware setup based on cameras inspired by the ocelli morphology in Drosophila. Next, in Section III-B we describe the CNNBiGRU proposed architecture. A new algorithm based on projective geometry is presented in Section IV, which will be used for benchmarking the proposed learning approach. After that, in Section V data recorded from simulation and real environments is explained. Section VI describe the learning results using both indoor and outdoor scenarios. There is also a comparative with the geometry-based model proposed on [18] and with the geometry-based approach explained in Section IV. The paper ends with a discussion and lines for future work.

II. A COMPUTER VISION PERSPECTIVE ON OCELLI

From a computer vision perspective, the ocelli structure can be seen as a multi-camera system in which the visual information of the photoreceptors (image pixels) is used to estimate the rotation the rigid-body undergoes. Figure 2 shows a simplified scheme of the ocellar system. The mission of this system is to compute the transformation $\mathbf{T}_{21}$ based on the visual information gathered by the photoreceptors, emulated using three cameras in our case. We can see how such transformation can be easily computed up to a scale factor from the optical flow computed between the images captured by each camera. While this is a conventional procedure in computer vision, the limited resolution of the sensors (in this research work will be assumed as 10×8 pixels) reduces the solutions to be applied.

This paper focuses on the estimation of the angular rates based on the information provided by the visual sensors and the use of Artificial Neural Networks to process such information. Notice that the ocelli system might be rotated and translated, but we focus in the rotation estimation and translation rejection. Future work will analyze the possibility of also estimating ocelli translation. Additionally, a second method based on non-linear optimization will be also derived and used to benchmark the output of the proposed network together with one state of the art method.

In order to get real data to test the proposed approaches, a hardware setup to emulate *Drosophila*'s ocellar system as a computer vision sensor has been conceived. The setup consists of three small fisheye cameras with 320×240 resolution spatially distributed according to the geometry of *Drosophila*'s ocelli. To emulate the biological system, the optics were chosen to have a 20% overlap between the images and a wide field of view (approximately $110°$), in contrast with the narrower FoV ($60°$) used in previous work [20]. To reproduce *Drosophila*'s vision through ocelli, images recorded are downsampled and blurred to 10×8 pixel images. It is worth to mention that cameras' frame rate is fixed at 30Hz. Although this is not relevant for smooth rotations, this rate limits the maximum rotation rate the cameras can notice.

To complete the basic system, an Inertial Measurement Unit (IMU) has been attached to the device's base in order to have a ground-truth of the angle rate on three axis. This sensor integrates a three-axis gyroscope, accelerometer and magnetometer. The three images and the three ground-truth angle rates constitute the pair inputs-labels used to train and test the approaches proposed in this paper.

III. LEARNING TO ESTIMATE ANGULAR RATES USING ARTIFICIAL NEURAL NETWORK

We present a learning, model-free approach, in which we estimate a mapping between image inputs and angular rate outputs by using a neural network.

A. The temporal nature of the problem

In order to recover angular rates from sequenced images, it is necessary to supply the network with temporal information about input data. The images from the three cameras are first stacked into a single 30×8 image, as follows: first left camera image, then frontal camera image and finally right camera

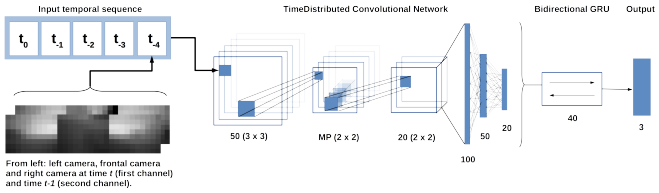

Fig. 3. Scheme of the CNNBiGRU proposed.

image. Then, the input to the network is this stacked gray-scale image with two channels: the first channel is the image at time *t* and the second channel is the image at time *t-1*.

Besides this temporal information, the network must be able to learn the natural evolution of a sequence of angular rates in a rotational movement. In order to avoid abrupt changes in the prediction of our ANN, the network incorporates a layer that works as a sequence processor. Thus, we organize the input data as temporal sequences of five elements, each one containing a 30×8 image with two channels. This layer works as a temporal filter over the sequence, being able to smooth the predicted angular rates.

Henceforth, in order to clarify the data format, the notation `[samples, seq_length, channels, height, width]` is used, where the first dimension is the number of samples, the second refers to the length of the temporal sequence, and the other three refers to channels and image's dimensions.

B. Convolutional-BiGRU Neural Network

Determining the best architecture and internal parameters of an ANN (number of layers, number of units per layer, activation function, etc) is a great challenge. In our previous work [20], we studied three different internal structures to analyze the influence of network's depth and width. Here we present a network with few layers and simple architecture in concordance with this previous study (see Fig. 3).

In this work we use a Convolutional Neural Network with a Bidirectional GRU layer (CNNBiGRU). The goal of the network's convolutional layers is to extract spatial and temporal features (recall that the input contains images at two time instants). The BiGRU layer processes the temporal evolution of these features, acting like a smoothing filter. We make use of Gated Recurrent Units (GRU) as they have shown to perform similar to LSTM networks [23] with less internal parameters and avoiding vanishing gradient problem.

At the top, the network is composed of two convolutional layers with 3×3 and 2×2 kernel size respectively. The first layer has 40 filters that are applied with zero padding and a stride of 1 unit in the horizontal direction and 2 in the vertical direction. Second layer has 20 filters without zero padding and a stride of 2 in both directions. A MaxPooling layer separates both layers, with a pool size 2×2. All layers, except the output layer, use the ReLu function [24] as activation function. After the convolutional part it follows the fully-connected block, composed of three layers with 100, 50 and 20 units respectively. The network is fed with a temporal sequence of length 5 (recall that each element of the sequence contains images of two consecutive time instants as two channels). First layers apply independently to each of the elements of the temporal sequence. Once the spatio-temporal information has been extracted and processed, a Bidirectional GRU layer with 40 units works on the full sequence. To finish, the output is a fully-connected layer with three units and linear activation function. A 20% Dropout layer [25] is used after every other layer, except the output, to avoid overfitting.

IV. PERSPECTIVE GEOMETRY APPROXIMATION FOR OCELLI

As baseline, in this section we summarize a perspective geometry approximation of the ocelli rotations purely based on the whole visual information provided by the image sensors. This section proposes a direct method to estimate the rotation of the ocelli system based on the minimization of the photometric error in all the cameras at once.

Thus, if a camera undergoes pure rotations, a full homography can model the pixel transform between sequenced images in both 2D and 3D scenes [26]. In this case we can model the image transforms between previous and current image frames of the ocelli using three homographies, one for each visual sensor (left, right and front). Also under pure rotation assumption, a homography matrix $\mathbf{H}$ can be decomposed according to the following expression $\mathbf{H} = \mathbf{A}\mathbf{R}\mathbf{A}^{-1}$, where $\mathbf{A}$ stands for the intrinsic camera calibration matrix and $\mathbf{R}$ is the rotation that the camera suffered between images.

According to Fig. 2, the rotations that the different image sensors experiment are actually linked each other because they are fixed to the same rigid body. Thus, if the ocelli system undergoes a pure rotation $\mathbf{R}_{21}$, we can establish the following constraint for each camera:

$$\begin{aligned} \mathbf{H}_{21}^{L} &= \mathbf{A}^{L}(\mathbf{R}^{L})^{T}\mathbf{R}_{21}\mathbf{R}^{L}(\mathbf{A}^{L})^{-1} \\ \mathbf{H}_{21}^{R} &= \mathbf{A}^{R}(\mathbf{R}^{R})^{T}\mathbf{R}_{21}\mathbf{R}^{R}(\mathbf{A}^{R})^{-1} \\ \mathbf{H}_{21}^{F} &= \mathbf{A}^{L}(\mathbf{R}^{F})^{T}\mathbf{R}_{21}\mathbf{R}^{F}(\mathbf{A}^{F})^{-1} \end{aligned} \tag{1}$$

where $\mathbf{A}^{L}$, $\mathbf{A}^{R}$ and $\mathbf{A}^{F}$ are the intrinsic calibration matrix of each sensor and $\mathbf{R}^{L}$, $\mathbf{R}^{R}$ and $\mathbf{R}^{F}$ are the rotations that align the sensors to the ocelli system reference. Calibrations and rotations are known parameters that can be estimated beforehand.

Given the images of the sensor in the current time step $\mathbf{I}_2^L$, $\mathbf{I}_2^R$ and $\mathbf{I}_2^F$, and the images in the previous time step $\mathbf{I}_1^L$, $\mathbf{I}_1^R$ and $\mathbf{I}_1^F$, we propose minimizing the photometric error of the re-projected images according to the following non-linear optimization problem:

$$\begin{aligned} \underset{\{\mathbf{R}_{21},\mathbf{f}\}}{\arg\min} \Bigg[& \sum_{\mathbf{x}} (\mathbf{I}_1^L(\mathbf{x}) - b_L - s_L \mathbf{I}_2^L(\mathbf{H}_{21}^L(\mathbf{x})))^2 \\ & + \sum_{\mathbf{x}} (\mathbf{I}_1^R(\mathbf{x}) - b_R - s_R \mathbf{I}_2^R(\mathbf{H}_{21}^R(\mathbf{x})))^2 \\ & + \sum_{\mathbf{x}} (\mathbf{I}_1^F(\mathbf{x}) - b_F - s_F \mathbf{I}_2^F(\mathbf{H}_{21}^F(\mathbf{x})))^2 \Bigg] \end{aligned} \tag{2}$$

where we optimize the values of $\mathbf{R}_{21}$ and $\mathbf{f} = [b_L, s_L, b_R, s_R, b_F, s_F]^t$ subject to the re-projection error for all the pixels $\mathbf{x}$ in each image pair. The scalars b and s correspond to a bias and scaling factor in the image, they are computed for each image separately. We can see how the expression compares the every pixel $\mathbf{x}$ from image $\mathbf{I}_1$ with the corresponding pixel into image $\mathbf{I}_2$ according to the homography that relates both images $\mathbf{H}_{21}(\mathbf{x})$. The values of $\mathbf{H}_{21}^L$, $\mathbf{H}_{21}^R$ and $\mathbf{H}_{21}^F$ in (2) are computed from the current rotation solution according to (1).

This optimization problem is solved using a Levenberg-Marquardt solver. The rotation $\mathbf{R}_{21}$ is parametrized into the optimization problem as a quaternion $\mathbf{q}_{21}$. Together with the bias and scaling factors, the total number of parameters to be solved is small, just ten. Additionally, using all image sensors at once allows better constraining the problem; if a rotation is poorly observed by one of the cameras, it might be better observed by another sensor with a different orientation.

As mentioned, this method considers two main assumptions. A pure rotation of the system between two consecutive time instants, and the cameras sharing the optical centre. The first one is a good approximation for consecutive images in our datasets, while the latter is validated by the calibration of our system.

V. PUBLIC DATASETS FOR OCELLI VALIDATION

When training a Deep Neural Network, data acquisition is one of the limiting factors. Real experiments are time consuming and expensive, and sometimes it is not possible to gather the amount needed. To avoid this limitation, one solution consists on using a simulator to train the network and after that, fine tune or retrain the network with real data. This way most parameters are trained with simulated data and the network only need few real experiments to adjust to real case. In this work we use AirSim [27], a novel open-source simulator built over the Unreal Engine, to reproduce the physical hardware presented in Section II and simulate an environment to capture data. As in real case, we gather sequences of three images with resolution 320×240 and the corresponding three angular rates.

Using AirSim, we created 131 simulated datasets in two main scenarios and changing light conditions: 35 sets in an outdoor scenario with a fixed light source (the Sun) from 11 different directions; 30 sets in an indoor scenario with 11 simultaneous light sources and 20 indoor sets with 6 light sources; and 35 sets in an indoor scenario with windows (a porch) changing the external light direction (see Fig. 4). Each dataset was recorded in different parts of the scenarios. All simulated datasets contain pure rotation movements along the three axis. Regarding real data, the physical device provides, at time *t*, three gray-scale images with resolution 320×240 pixels. At each instant we also have the three angular rates recorded by the IMU attached to the base of the device. We captured 24 sets in indoor and outdoor scenarios with different light conditions (see Fig. 4). In real data we have pure rotational and translation sets. All real experiments are recorded by hand, thus, we have to expect some noise in the output signal together with small translations.

Altogether we have 414929 simulated and 127692 real-data samples, each of them with one compound image from the three cameras information at two consecutive time instants, and three angular rates, one per axis, coming from the IMU.

A. Data preprocessing

In order to use data recorded with both simulated and real device to train the proposed network, it is necessary to undergo some pre-processing computations. For more details about this processing, please visit the the following web[2].

VI. EXPERIMENTAL RESULTS AND BENCHMARKING

In this section we present the results of the network's learning process. We implement the network using Python and Keras API [28]. All computations are performed on Ubuntu 16.04 with `GPU NVIDIA TESLA K40`. The CNNBiGRU learning process is divided in two main steps: first the network is trained only with data from simulation and then, once the model is trained and parameters are learned, we compute a fine-tune training process with real data. The first network training is compute with 122 of 133 datasets from simulation, all of them containing pure rotation movements in different scenarios. Thus, the network is trained with 387136 samples and the 9 resting sets (27793 samples) are earmarked to test the trained model (3 outdoor sets, 3 indoor sets and 3 porch sets). The learning algorithm used to train the network is the Adaptative Moment Estimation (Adam), that has shown to be more effective on CNN networks over other stochastic first-order methods [29].

Once the network is trained with simulation data, we perform a fine-tuning process in order to sightly retrain the parameters of the network to adapt to real data. In this case the re-learning process is compute with 22 (99380 samples) of the the 24 sets. The remaining sets are reserved for testing purposes. In contrast with training process, in fine-tuning process we include some sets of real translations data. The proposed approach is able to estimate angular rates with pure rotational movements but real data is not ideal and there are some translations movements among rotations. To let the network deal with those cases, we fine-tune the model with pure translations sets. Thus, the network can reduce the error between real and predicted output in these regions. CNNBiGRU is also fine-tuned with Adam algorithm but with a low learning rate (lr = 0.000001), while in learning phase was 0.0001. This way, we let the parameters to update and adapt to real data in a small interval, assuming their values are already near an optimum thanks to training process with simulated data.

Both learning and fine-tuning processes take 400 epochs with a batch size of 100 samples per epoch. We compute the mean squared error (henceforth MSE) between network's

[2] `https://github.com/robotics-upo/OCELLIMAV-Project/tree/master/data`

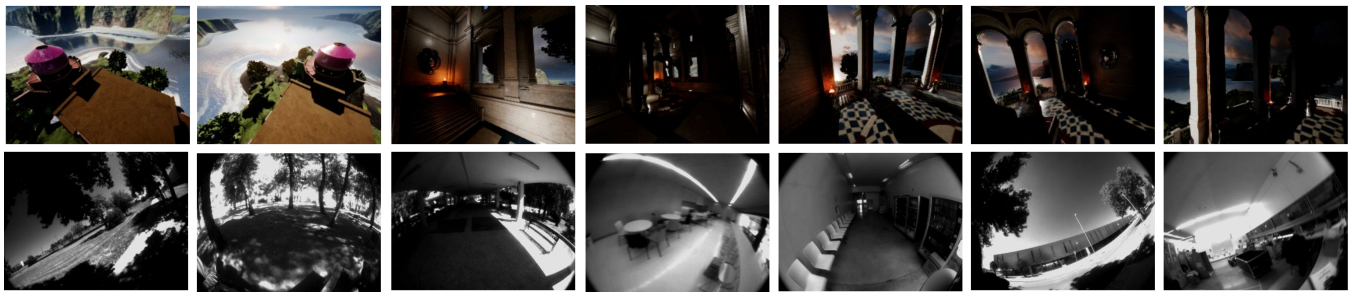

Fig. 4. (From left up to right: simulated scenarios) (1) Outdoor; (2) Outdoor with different light direction; (3) Indoor with 11 light sources; (4) Indoor with 6 light sources; (5) Indoor with windows; (6) and (7) Indoor with windows and different light direction. (From left down to right: real scenarios) (8), (9), (10) and (13) Outdoor scenarios; (11) and (12) Indoor scenarios.

TABLE I
TESTING RESULTS FOR BOTH SIMULATED AND REAL DATA IN DIFFERENT SCENARIOS.

	MSE (SEM) (radˆ2/sˆ2)	
	Simulation Testing	**Real Testing**
Outdoor	0.034 (0.003)	0.171 (0.005)
Indoor	0.044 (0.004)	0.228 (0.008)
Porch	0.044 (0.004)	-

predictions (outputs) and ground-truth labels as loss function. In addition, after every epoch the network is evaluated over the last 20% of the training data, without compromising testing process.

A. CNNBiGRU learning results

a) Training from scratch: As mentioned before, the network is first trained only with data from AirSim simulator. In order to properly evaluate the predicted angular rates, we analyze the response of the network with different testing sets: one per scenario considered in training data (see Table I). The code used to train and test the network are available[3].

b) Fine-tuning the model: Once the network is trained and internal parameters are learned, the resulting model is fine-tuned with real data. We evaluate the network's output both in indoor and outdoor scenarios. In first case the experiment was recorded in a building's hall, with chairs and coffee machines, with lights all over the ceiling. About second testing set, the experiment was perform with open sky at 5 PM in an outdoor scenario with sun behind a tree top, roads, cars, etc. Fine-tuning results over these two testing sets are shown in Table I. In Fig.5 temporal evolution of ground-truth and predicted angular rates for the outdoor scenario is shown, along its corresponding error histogram.

Although the network is able to recover the general dynamic with real data, the error is higher than with simulation data. We think they are produced by a combination of the following effects: first of all, simulations are not affected by translations while real data do. Also, the simulated artificial ocelli is not ideal, and there might exist differences between the simulated and real system such as cameras' orientation and radial distortion. Finally, the training angular rates in simulation data were constrained between 2.5 and $-2.5\frac{rad}{s}$, approximately. Thus, there are some extreme angular rates the network is not able to recover with real data.

[3]https://github.com/robotics-upo/OCELLIMAV-Project/tree/master/scripts

B. Gyroscope random BIAS estimation

While the estimations of the presented system cannot be as accurate as a gyroscope for rate estimation, the network estimation has the advantage of being BIAS free, unlike MEMs gyroscopes. IMUs need of accelerometers and magnetometers to estimate and correct the random BIAS present in the gyroscopes.

The BIAS free nature of the network prediction gives the opportunity of estimating the gyroscope BIAS without accelerometers and magnetometers, just the low resolution image information. Thus, a simple Kalman Filter can be implemented for each axis. The filter estimates the angle rate and the BIAS, and receives as measurement the angle rate provided by the gyroscope (with BIAS) and the noisy ocelli estimation (BIAS free). The prediction is modelled as a random walk, adding noise to both rate and BIAS to account for the dynamics of each variable.

In order to evaluate this approach, an experiment was conceived. One of the real sets for testing has been used to test the proposed filter. Known BIAS values (unknown for the filter) from -1.5 to $1.5rad/s$ has been added to the gyroscope and applied to the filter together with the ocelli estimation. In total, 600 initial configurations have been tested and the summary of the evaluation of the error in the BIAS estimation is shown in Fig. 6. We can see that the errors are high at the beginning of the estimation, and how the filter reduces such errors thanks to the integration of the BIAS free estimation provided by the ocelli system.

C. Benchmarking

In this section we compare our learning approach to two alternative methods: the geometric approach described in Section IV and the method presented in [18]. The recovery of angular rates in Fuller's method is computed from the temporal derivative of the signal emitted by a set of 2 or more non-coplanar phototransistors. Considering a rigid solid and a punctual and well-known direction light source they are able to recover ω_x (roll) and ω_y (pitch). Because of the method, it is not possible to recover the angular rate on the

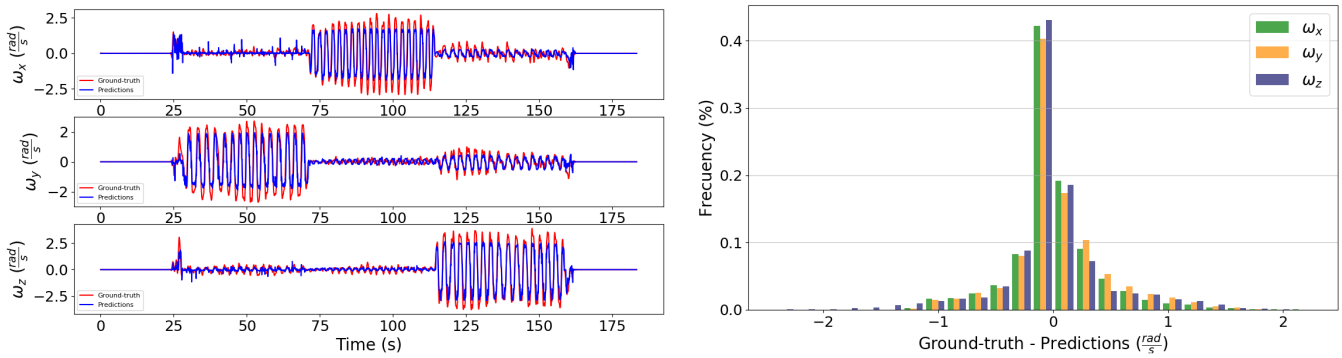

Fig. 5. (Left) Temporal evolution of angular rates, ground-trugh (red) and predicted (blue), of a real testing set on outdoor scenario. (Right) Error histogram between ground-truth and predicted angular rates of the same testing set.

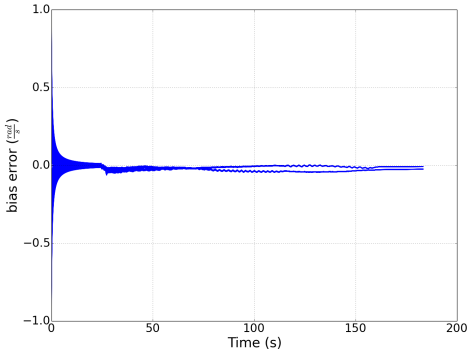

Fig. 6. Computed BIAS error in the estimation using the proposed Kalman Filter. We can see how errors are reduced as the filter integrates information provided by the gyroscopes and ocellar system.

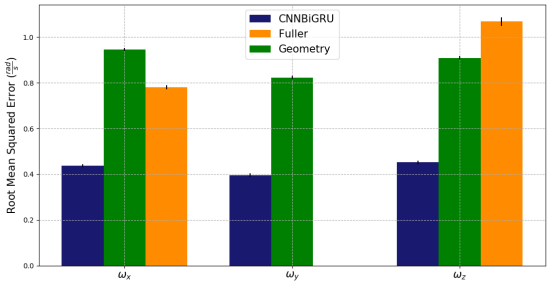

Fig. 7. Root mean squared error over a real testing data set on indoor scenario. Comparative between methods.

axis the light source direction lies on (yaw angular rate in [18]).

In order to implement this model we define the director vectors of our three cameras in our reference system (see Fig.2). Left and right cameras are elevated $45°$ over x-axis, while frontal camera is raised $50°$ over y-axis. Thus, vectors are $\vec{L} = [\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0]$, $\vec{F} = [0, \cos 0.87, -\sin 0.87]$ and $\vec{R} = [-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0]$ for left, frontal and right camera respectively. We consider a source light relying on y-axis, thus, w_y cannot be predicted.

Regarding the geometry-based approach presented in Section IV, this method can, as our network, recover the three angular rates. The method has been coded in C and it is publicly available[4].

A benchmarking analysis is shown in Fig.7, where the three methods' predictions over the real testing indoor set are compared. Fuller's method does not work properly with the outdoor testing set because of light dispersion (the experiment was recorder with the sun behind a tree top). Our learning approach presents the lowest errors on each axis. Although the estimation is still worse than CNNBiGRU prediction (see Fig.7), geometric approach prediction is better than Fuller's response on z-axis. We suspect that the low resolution of the images has a negative influence in the estimation of the image gradients required for the computation of the iterative solution in the non-linear optimization problem.

[4]https://github.com/robotics-upo/OCELLIMAV-Project/tree/master/geometry_approach

VII. CONCLUSIONS AND FUTURE WORK

In this work we have presented a direct method able to recover angular rates using machine learning and computer vision. This approach is computationally efficient and can work with low resolution cameras to estimate attitude in both indoor and outdoor scenarios.

The method has been successfully trained and tested with synthetic and real images, and benchmarked against two different methods. Additionally, a public dataset with simulated and real data is released to help the research community to develop new Machine Learning approaches for attitude estimation.

Future work will analyze the posibility of estimating the system translation (even scaled) in order to better constraint the rotation estimation. The use of numeric methods similar to the geometry approach presented in this paper with larger resolution images will be also explored. The network's estimations would be used to get a good solution near the optimal to reduce computational impact due to resolution augmentation.

REFERENCES

[1] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012. [Online]. Available: http://dx.doi.org/10.1177/0278364912455954

[2] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[3] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and slam for medium and high altitude flying uavs," *Journal of Intelligent and robotics systems*, vol. 54, pp. 137–161, 2009.

[4] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 957–964.

[5] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013. [Online]. Available: http://dx.doi.org/10.1177/0278364913481251

[6] L. Merino, J. Wiklund, F. Caballero, A. Moe, J. R. M. De Dios, P.-E. Forssen, K. Nordberg, and A. Ollero, "Vision-based multi-uav position estimation," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 53–62, 2006.

[7] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[8] A. Amor-Martinez, A. Ruiz, F. Moreno-Noguer, and A. Sanfeliu, "On-board real-time pose estimation for uavs using deformable visual contour registration," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2595–2601.

[9] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2609–2616.

[10] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a uav," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 3309–3316.

[11] F. L. Roubieu, J. R. Serres, F. Colonnier, N. Franceschini, S. Viollet, and F. Ruffier, "A biomimetic vision-based hovercraft accounts for bees complex behaviour in various corridors," *Bioinspiration & Biomimetics*, vol. 9, no. 3, p. 036003, 2014. [Online]. Available: http://stacks.iop.org/1748-3190/9/i=3/a=036003

[12] D. Floreano, R. Pericet-Camara, S. Viollet, F. Ruffier, A. Brckner, R. Leitel, W. Buss, M. Menouni, F. Expert, R. Juston, M. K. Dobrzynski, G. LEplattenier, F. Recktenwald, H. A. Mallot, and N. Franceschini, "Miniature curved artificial compound eyes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 23, pp. 9267–9272, 2013. [Online]. Available: http://www.pnas.org/content/110/23/9267.abstract

[13] J. C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 137–146, Feb 2006.

[14] M. Mizunami, "Functional diversity of neural organization in insect ocellar systems," *Vision Research*, vol. 35, no. 4, pp. 443 – 452, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/004269899400192O

[15] H. G. Krapp, "Ocelli," *Current Biology*, vol. 19, pp. 435–437, 2009.

[16] J. Chahl and A. Mizutani, "Biomimetic attitude and orientation sensors," *IEEE Sensors Journal*, vol. 12, no. 2, pp. 289–297, Feb 2012.

[17] G. Gremillion, J. S. Humbert, and H. G. Krapp, "Bio-inspired modeling and implementation of the ocelli visual system of flying insects," *Biological Cybernetics*, vol. 108, no. 6, pp. 735–746, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00422-014-0610-x

[18] S. B. Fuller, M. Karpelson, A. Censi, K. Y. Ma, and R. J. Wood, "Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli," *Journal of The Royal Society Interface*, vol. 11, no. 97, 2014. [Online]. Available: http://rsif.royalsocietypublishing.org/content/11/97/20140281

[19] T. R. Neumann and H. H. Bülthoff, "Behavior-oriented vision for biomimetic flight control," in *Proceedings of the EPSRC/BBSRC international workshop on biologically inspired robotics*, 2002, pp. 196–203.

[20] M. Mérida-Floriano, F. Caballero, D. Garcí-Morales, F. Casares, and L. Merino, "Bioinspired Vision-only UAV Attitude Rate Estimation using Machine Learning," in *Proceedings of the International Conference on Unmanned Aircraft Systems, ICUAS*, 2017, pp. 1–6.

[21] P. Sermanet, R. H. M. Scoffier, M. Grimes, J. Ben, A. Erkan, C. Crudele, U. Muller, and Y. Lecun, "A multi-range architecture for collision-free off-road robot navigation," *Journal of Field Robotics*, 2009.

[22] A. Giusti, J. Guzzi, D. C. Cirean, F. L. He, J. P. Rodrguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, July 2016.

[23] J. Chung, aglar Gülehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines." in *ICML*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[26] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[27] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[28] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980