

# A New Stereo Fisheye Event Camera for Fast Drone Detection and Tracking

Daniel Rodrigues Da Costa\*, Maxime Robic\*, Pascal Vasseur, Fabio Morbidi

**Abstract**—In this paper, we present a new compact vision sensor consisting of two fisheye event cameras mounted back-to-back, which offers a full 360-degree view of the surrounding environment. We describe the optical design, projection model and practical calibration using the incoming stream of events, of the novel stereo camera, called SFERA. The potential of SFERA for real-time target tracking is evaluated using a Bayesian estimator adapted to the geometry of the sphere. Real-world experiments with a prototype of SFERA, including two synchronized Prophesee EVK4 cameras and a DJI Mavic Air 2 quadrotor, show the effectiveness of the proposed system for aerial surveillance.

## I. INTRODUCTION

Event-based vision, also known as neuromorphic vision, is a novel and increasingly popular paradigm in machine perception [1]. Unlike traditional frame-based cameras, which capture the absolute intensity perceived from a scene at a constant rate, event cameras operate in an asynchronous fashion, recording only discrete brightness changes at the time they occur. Event cameras have low temporal latency, making them insensitive to motion blur, they can work in low-light conditions, and they do not suffer from saturation, thanks to the high dynamic range. They generate sparse data in dynamic environments and they are competitive in real-time embedded applications, such as optical flow estimation [2], visual odometry [3], [4], [5], and visual SLAM [6], [7]. Moreover, they are well suited for target tracking [8], [9], especially of fast-moving objects [10], [11], since their temporal resolution (in the order of microseconds) surpasses that of conventional RGB cameras.

Yet, off-the-shelf event cameras suffer from a *narrow field of view* (FoV), so either a fast target or the visual features used for the navigation of an agile robot, might unexpectedly exit from the tracking area. While an event camera can be controlled to actively follow a specific target (e.g. by visual servoing [12]), a better option is to enlarge its FoV by using a rotating device [13], rigidly mount multiple sensors on the same platform, or use a combination of lenses and mirrors [14]. However, since the input data is sparse and asynchronous, this new class of omnidirectional sensors comes with a number of challenges pertaining to



Fig. 1. *Conceptual illustration of SFERA.* The sensor is used here for event-based omnidirectional detection and tracking of a flying object (e.g. a drone).

the optical design, geometric modeling, calibration and real-time data processing. To the best of our knowledge, no omnidirectional event camera offering a full 360° (spherical) FoV without rotating parts, has been proposed so far in the literature.

Our objective, in this paper, is to capitalize on the benefits of event-based mechanism and of a wide FoV. To this end, we present the design, modeling and calibration of a new omnidirectional event sensor, called SFERA (**S**tereo **F**isheye **E**vent **c**ame**R**A), which is composed of two synchronized fisheye event cameras rigidly attached on the same support and pointing in opposite directions<sup>1</sup>. SFERA has been designed to be compact and to possess a dead zone of minimum size on the spherical FoV (see Fig. 1). Omnidirectional vision has been the subject of increasing attention in aerial robotics over the past few years [15], [16], and our ultimate goal is to equip multiple UAVs with our camera, for defense and security applications.

Modeling and calibrating a multi-camera system (i.e. determining the intrinsic parameters of each camera and the rigid transformation between them) is usually a challenging and time-consuming process. In order to have a single representation of input data, one may either choose a *central model*, which considers a single and common viewpoint for each camera [17], or a *non-central model*, which assumes multiple viewpoints and involves more

<sup>1</sup>We use the term *stereo*, even though, unlike conventional stereo vision systems, the two cameras of SFERA do not point in the same direction. The more generic terms, *dual* or *binocular*, equally apply to our sensor.

\*These authors equally contributed to this work.

The authors are with the MIS Laboratory, University of Picardie Jules Verne, 80039 Amiens, France. *Corresponding author:* M. Robic. Emails: `FirstName.LastName@u-picardie.fr`

This work was supported by the French National Research Agency through the DEVIN project, “Drones with Omni-Event Vision for Drone Neutralization” (ANR-23-IAS2-0001) and by AID (Agence de l’Innovation de Défense) through the EVENTO project, “Omnidirectional Event Cameras for High-Speed Robots” (2021-2024).

complex (back-)projection functions [18]. In the case of wide-angle cameras, e.g. cameras equipped with fisheye lenses, these models should be modified to account for the large FoV and optical distortions [19]. In this paper, we adopt the classical unified central projection model [20], [21], which has already been applied to dual-fisheye cameras (such as the Ricoh Theta S) in [17]. In fact, the spherical geometry is not only attractive to represent/visualize the events, but also to efficiently process them.

Related work on *event-camera calibration* has mainly focused on the acquisition process. The idea is to reconstruct a planar calibration pattern thanks to the camera’s motion [22], [23], [24] or to an external source of light, e.g. blinking screens or LEDs [25]. Traditional feature-based optimization algorithms can then be used to process the pseudo-images of the calibration patterns. Recently, in [26], the authors have proposed a motion-based auto-calibration algorithm, which takes the lines present in the environment, as input. Finally, in other works, a second sensor, e.g. a frame-based camera in a stereo rig [27], [28] or a LiDAR [29], [30], has been exploited: this greatly simplifies the pattern detection step and the estimation of the extrinsic parameters. These calibration methods are practical and guarantee low re-projection errors, but they are almost exclusively meant for perspective cameras and they are not applicable to dual sensors with a small overlapping FoV. Inspired by [25], in this work, the two fisheye event cameras of SFERA observe two blinking patterns on two monitors and the events are aggregated to generate pseudo-images, which are fed into conventional calibration software for omnidirectional cameras.

A prototype of SFERA, including two synchronized Prophesee EVK4 cameras, has been developed and validated. In our real-world experiments, building upon earlier work on panoramic vision [31], [32], [33], we successfully tested a Bayesian state estimator (tracker) based on the von Mises-Fisher (VMF) distribution, which fully benefits from our representation of the events on the unit sphere. In summary, the *original contributions* of this paper can be summarized as follows:

- We introduce SFERA, the *first* event-based sensor with a complete  $360^\circ$  FoV and no rotating parts,
- We describe the full pipeline, including the optical design, geometric modeling, and calibration of SFERA,
- A prototype of SFERA has been developed and tested via outdoor experiments, for fast drone detection and tracking. To foster reproducible research, our open-source C++ code is publicly available on GitHub<sup>2</sup>.

The remainder of this paper is organized as follows. Sect. II deals with the optical design, image formation and calibration of SFERA. Sect. III is devoted to the experimental evaluation of our prototype of the stereo event camera. Finally, in Sect. IV, the main contributions of the paper are summarized and some possible avenues for future research are outlined.

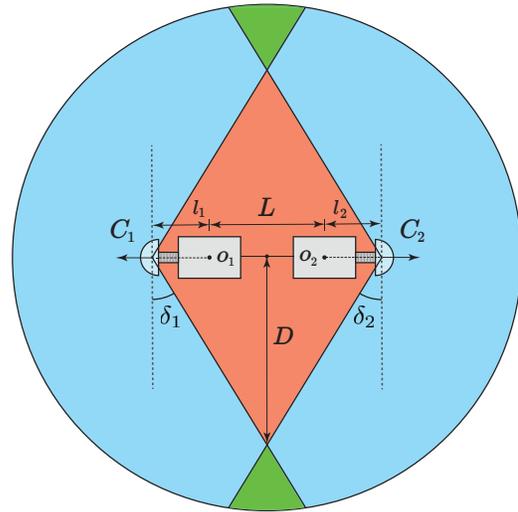


Fig. 2. SFERA: optical design (top view). Two fisheye event cameras,  $C_1$  and  $C_2$ , are rigidly attached to the same support and point in opposite directions. The blue area is covered by a single camera, the green area by both cameras, and the red area by none of the cameras (dead zone).

## II. DESIGN, MODELING AND CALIBRATION OF SFERA

In this section, we start by presenting the optical design of SFERA. We proceed with a description of the camera projection model and a discussion of the event acquisition mechanism. Finally, we explain how the events are represented on the unit sphere, and introduce the calibration protocol developed for SFERA.

### A. Camera design

The optical design of SFERA is shown in Fig. 2. Two event cameras,  $C_1$  and  $C_2$ , are rigidly mounted back-to-back on the same support and the distance between their optical centers,  $o_1$  and  $o_2$ , is  $L$ . To simplify the calibration procedure (cf. Sect. II-D), we will assume that the optical axes of the two cameras are aligned. Each event camera is endowed with a fisheye lens, which is located at a distance  $l_1$  ( $l_2$ , respectively) from  $o_1$  ( $o_2$ , respectively). The two fisheye event cameras have a horizontal FoV, symmetric with respect to the optical axis, of  $180^\circ + 2\delta_i$ ,  $i \in \{1, 2\}$ , with  $\delta_1, \delta_2 \in (0, 90^\circ)$ . As a result, by restricting our analysis to the horizontal plane, we can identify three zones around SFERA (cf. Fig. 2): the area covered by a single camera (blue), the area covered by both cameras (green), and the dead zone (red). In this plane, the dead zone has the shape of a kite, with diagonals of length  $L + l_1 + l_2$  and  $2D$ , respectively (note that if  $\delta_1 = \delta_2$ , the kite becomes a rhombus). By using the law of sines and Apollonius theorem, the diagonal  $D$  can be expressed as:

$$D = \frac{L + l_1 + l_2}{2} \sqrt{2 \left( \frac{\cos^2 \delta_1 + \cos^2 \delta_2}{\sin^2(\delta_1 + \delta_2)} \right) - 1}. \quad (1)$$

Hence, the area of the dead zone is  $A = D(L + l_1 + l_2)$ . There are two ways to minimize  $A$ . The first is to minimize  $l_1, l_2$  and maximize  $\delta_1, \delta_2$  by a suitable choice of fisheye lenses. The second is to minimize the distance  $L$  between the two event cameras. Obviously, physical constraints prevent the distances  $l_1, l_2$  and  $L$  from being equal to zero.

<sup>2</sup><https://github.com/maxrobic/SFERA>

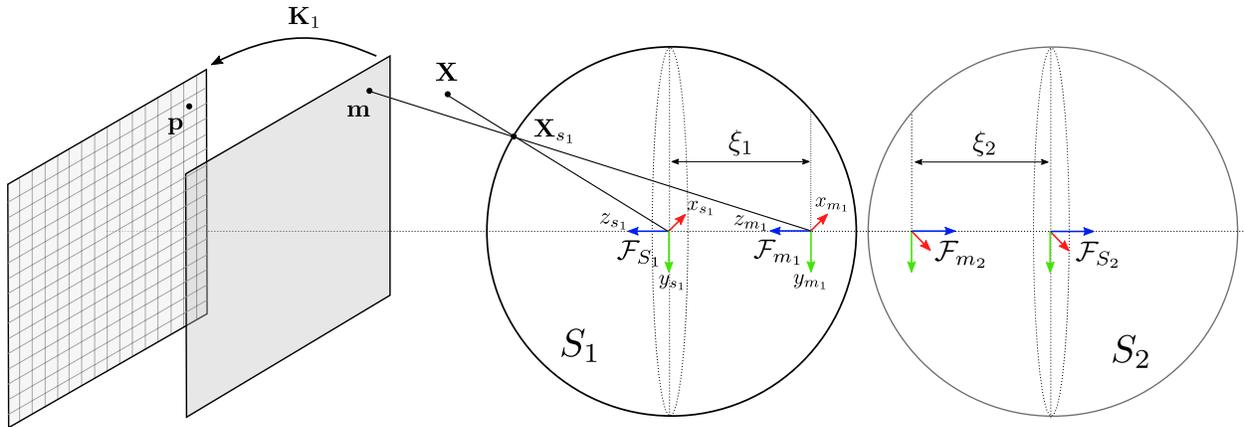


Fig. 3. SFERA: projection model. Each fisheye camera is modeled using a unit sphere,  $S_1$  and  $S_2$ , a normalized plane and a pixel image.

### B. Projection model

To describe SFERA, we chose the unified central projection model [20], [21]. This model has been extensively used in the literature for wide-angle cameras (catadioptric or fisheye), with a single viewpoint. Following [17], the unified central projection model has been separately applied to each lens of our stereo system. We made the simplifying assumption that the distance  $L$  between the optical centers of the two event cameras is small enough (compared to the distance of the objects observed in the scene) to be ignored. We can then consider a central representation of data, through a unique projection surface (the unit sphere).

The full projection model of SFERA is depicted in Fig. 3. With reference to the first camera,  $C_1$ , a 3D point  $\mathbf{X}$  is projected on the surface of the unit sphere  $S_1$  centered at  $\mathcal{F}_{S_1} = \{O_{S_1}; x_{s_1}, y_{s_1}, z_{s_1}\}$ , at point  $\mathbf{X}_{S_1}$ . Let  $\mathcal{F}_{m_1} = \{O_{m_1}; x_{m_1}, y_{m_1}, z_{m_1}\}$  be the frame with origin  $O_{m_1}$ , at a distance  $\xi_1$  from  $O_{S_1}$  along the  $x$ -axis. By using the projection point  $O_{m_1}$ ,  $\mathbf{X}_{S_1}$  is mapped to  $\mathbf{m} = [x, y, 1]^T$  in the normalized plane. Finally, the point  $\mathbf{p} = [u, v, 1]^T$  is expressed as  $\mathbf{p} = \mathbf{K}_1 \mathbf{m}$  in pixel coordinates, where

$$\mathbf{K}_1 = \begin{bmatrix} k_{u_1} & 0 & u_{0_1} \\ 0 & k_{v_1} & v_{0_1} \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

is the matrix of intrinsic parameters of the first camera. In equation (2),  $(k_{u_1}, k_{v_1})$  are the focal lengths in pixels in the horizontal and vertical directions, respectively, and  $(u_{0_1}, v_{0_1})$  are the coordinates of the principal point in pixels. The point  $\mathbf{m}$  and the point  $\mathbf{X}_{S_1}$  on the unit sphere  $S_1$ , are thus related by the following formula [21]:

$$\mathbf{X}_{S_1} = \begin{bmatrix} \frac{\xi_1 + \sqrt{1 + (1 - \xi_1^2)(x^2 + y^2)}}{x^2 + y^2 + 1} x \\ \frac{\xi_1 + \sqrt{1 + (1 - \xi_1^2)(x^2 + y^2)}}{x^2 + y^2 + 1} y \\ \xi_1 + \sqrt{1 + (1 - \xi_1^2)(x^2 + y^2)} - \xi_1 \end{bmatrix}. \quad (3)$$

An equation analogous to (3) holds for the second camera,  $C_2$ , with parameters  $\mathbf{K}_2$  and  $\xi_2$ . In conclusion, the intrinsic parameters of SFERA are  $(\mathbf{K}_1, \xi_1, \mathbf{K}_2, \xi_2)$ . In Sect. II-D, we will present an efficient calibration algorithm to estimate

these parameters. Before delving into it, in the next section we will describe how event data is acquired and represented.

### C. Representation of the events

SFERA generates an asynchronous stream of events, triggered by brightness changes in the observed 3D scene. An event  $e$  can be described by  $e = [u, v, t, p, i]^T$  where  $(u, v)$  is the pixel position of the event triggered at time  $t$  (timestamp),  $p \in \{+1, -1\}$  is the polarity of the intensity change, and  $i \in \{1, 2\}$  is the ID of the camera. Once SFERA is calibrated, event  $e$  can be mapped to  $e_s = [\theta, \varphi, t, p, i]^T$  on the surface of the unit sphere  $S_i$  via equation (3), where  $\theta \in [0, \pi]$  is the polar angle and  $\varphi \in [0, 2\pi]$  is the azimuthal angle.

Note that a brightness change occurring in the *overlapping zone* between the two event cameras (green in Fig. 2), simultaneously triggers an event in  $C_1$  and  $C_2$ . The projection of these events on the unit sphere could then give rise to artifacts (duplicate measurements), as a consequence of our central projection hypothesis. This is, indeed, a known problem in multi-fisheye camera systems, which is typically handled by image stitching [34]. Unfortunately, this method is not directly applicable to event data. However, as shown in Sect. III, this modeling issue has a minor impact on the acquisition process, and SFERA works as intended when the observed objects are sufficiently far away from the camera.

### D. Camera calibration

In this section, we briefly describe the calibration protocol put in place for our stereo fisheye event camera (the corresponding code is available in the GitHub provided in Sect. I). To guarantee high accuracy and repeatability, we opted for a simple calibration setup in which SFERA is placed at the same distance between two identical computer screens displaying a blinking checkerboard pattern at a fixed frequency  $f$  (see Fig. 4). The use of a fixed frequency  $f$  allows to easily filter out the external noise (e.g. generated by ambient light), and to capture pertinent information from the calibration rig (see the left column in Fig. 5). The events are accumulated to generate two grayscale pseudo-images of the checkerboard patterns, from which the pixel coordinates

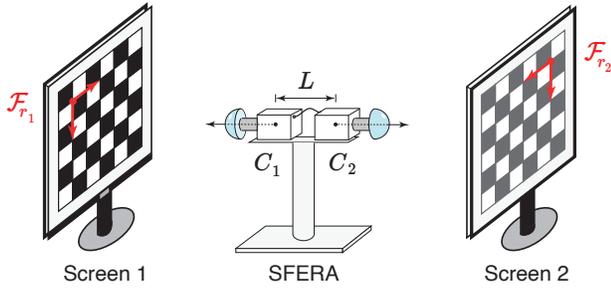


Fig. 4. *Calibration setup.* Two blinking checkerboard patterns with reference frames  $\mathcal{F}_{r_1}$  and  $\mathcal{F}_{r_2}$  (red), are observed by SFERA on two computer screens facing each other.

of the points of interest (corners) are extracted (right column in Fig. 5). The pixel coordinates are then automatically matched with the corresponding 3D world coordinates in the checkerboards' frames  $\mathcal{F}_{r_1}$  and  $\mathcal{F}_{r_2}$ . The extracted corners are used to estimate the intrinsic parameters of the two cameras via OpenCV's `omnicalib` module in the implementation of [35], which takes the two grayscale pseudo-images as input. This module relies on the calibration model described in Sect. II-B.

### III. EXPERIMENTAL VALIDATION

This section is divided into three parts. In the first part, we describe our prototype of SFERA. We then discuss our calibration results and we finally present the performance of a Bayesian estimator for target tracking, which is tailored to the geometry of the sphere.

#### A. Prototype of SFERA

Our current prototype of SFERA includes two Prophesee EVK4-HD cameras equipped with Fujinon FE185C086HA-11" fisheye lenses (see Fig. 6(a)). The lenses have a focal length of 2.7 mm and an angular FoV of  $185^\circ$  (H)  $\times$   $140^\circ 35'$  (V), which provide an excellent coverage of the surrounding environment (by consequence,  $\delta_1 = \delta_2 = 2.5^\circ$ ). The two cameras are clock-synchronized (Master-Slave configuration) using a synchronization cable. With reference

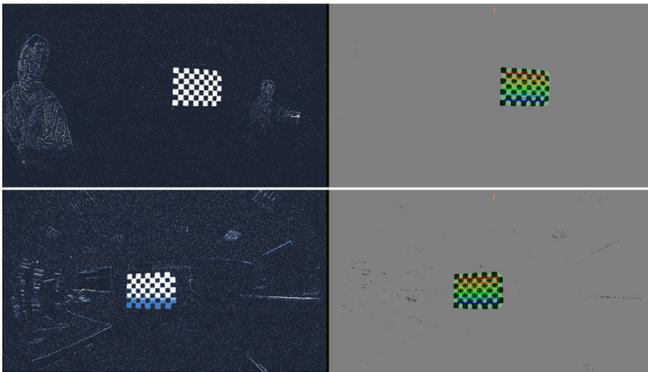


Fig. 5. *Two steps from our calibration protocol.* (Left column) Events generated by the blinking checkerboard patterns at a fixed frequency  $f = 15$  Hz; (Right column) Accumulation of the events for feature detection (the corners of the checkerboards). The top images correspond to  $C_1$  (Master) and the bottom images correspond to  $C_2$  (Slave).

to Fig. 2, we have  $l_1 = l_2 = 6.8$  cm and we set  $L = 7.5$  cm (the minimum separation permitted by the two connectors of the synchronization cable). In virtue of equation (1), we obtain  $D = 2.42$  m. Note that, in practice, an object with a footprint comparable to that of small a quadrotor, is still correctly perceived 1 m away from SFERA, despite the partial occlusion due to the dead zone. Our prototype has a small form factor:  $23.5 \times 5.5 \times 5.5$  cm<sup>3</sup> (support excluded). Since each lens (camera) weighs 160 g (80 g), the total weight of SFERA is about 500 g (cables included). Because of its compact design and low power consumption, our prototype can be easily carried by any mobile robot.

#### B. Calibration results

The prototype of SFERA has been calibrated using the algorithm described in Sect. II-D (the distortion coefficients have been omitted). The estimated intrinsic parameters of the two event cameras  $C_1$  and  $C_2$  are reported in Table I. These parameters guarantee a small RMS reprojection error (1.50 and 1.20 pixels for  $C_1$  and  $C_2$ , respectively), and a visualization of the events on the unit sphere which is globally satisfying, as shown in the example in Fig. 6(b). Note that the reference frame of SFERA (spherical view) coincides with that of the Master camera. Overall, our results indicate that the proposed calibration protocol is practical yet accurate.

TABLE I

Estimated intrinsic parameters (in pixels, except for  $\xi_1$  and  $\xi_2$ ).

Camera	$k_{u_i}$	$k_{v_i}$	$u_{0_i}$	$v_{0_i}$	$\xi_i$
$C_1$ (Master)	970.012	970.565	645.375	365.778	1.682
$C_2$ (Slave)	972.311	969.822	618.132	361.180	1.685

#### C. Target detection and tracking with SFERA

In this section, we present a possible application of SFERA to aerial monitoring. The goal is to rapidly detect and track a flying object around the ground-based camera. SFERA has a high temporal resolution, is able to detect brightness changes induced by fast-moving objects and benefits from a  $360^\circ$  FoV, thanks to the back-to-back stereo configuration.

In the *detection step*, the events generated by the Master and Slave cameras are clustered in the respective image frames. The *tracking step*, instead, is entirely performed on the unit sphere. For the first step, we used the cluster tracking algorithm provided by Prophesee's Metavision SDK, with an accumulation time of  $t_{acc} = 30$  ms. This algorithm allows to find the centroid of each active cluster which has been potentially generated by a moving object. These centroids are projected on the unit sphere via equation (3). For the tracking step, we adapted the Bayesian state estimator (tracker) based on the von Mises-Fisher (VMF) distribution [36] proposed in [31], to our specific setting. As a standard Kalman filter, the estimator goes through two distinct phases (prediction and correction), and it recurrently updates two quantities: the *mean direction* (or bearing vector)  $\mu$  of the target (a unit vector) and the *concentration parameter*  $\kappa \geq 0$ , which is the analogue of the inverse of the covariance in an

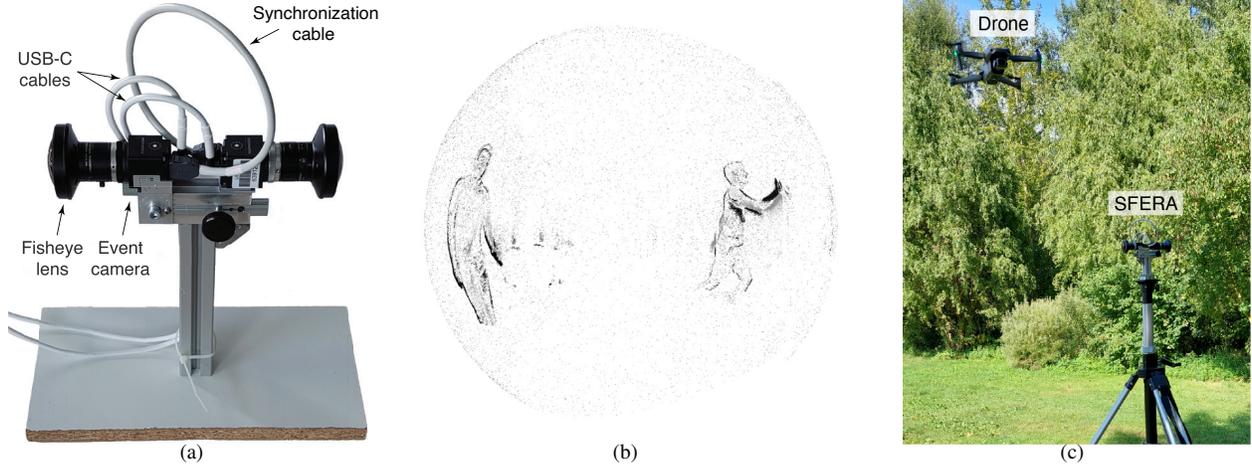


Fig. 6. (a) Current prototype of SFERA; (b) Visualization of the (accumulated) events  $e_s$  on the unit sphere, using the intrinsic parameters reported in Table I. In this example, SFERA captured the motion of two human bodies; (c) Testbed considered in the outdoor experiments described in Sect. III-C: SFERA is mounted on a tripod and the quadrotor flies around it.

isotropic Gaussian distribution. The larger the value of  $\kappa$ , the greater is the clustering around the mean direction [36, Sect. 9.3.2]. Let  $\mathbf{x}_j \sim \text{VMF}(\boldsymbol{\mu}_j, \kappa_j)$  where  $\boldsymbol{\mu}_j$  and  $\kappa_j$  are, respectively, the mean direction and the concentration parameter at time  $j \in \{1, 2, \dots\}$ . The estimator leverages the following dynamic process model:

$$\mathbf{x}_j = \mathbf{R}_j \mathbf{x}_{j-1} * \mathbf{w}_{j-1}, \quad (4)$$

where “\*” is a convolution-like operation defined in [37, Sect. 4.1], and  $\mathbf{w}_{j-1}$  is a VMF-distributed noise. Equation (4) can be equivalently written, considering the mean direction, as  $\boldsymbol{\mu}_j = \mathbf{R}_j \boldsymbol{\mu}_{j-1}$  where  $\mathbf{R}_j$  is a rotation matrix. In [31], the authors proposed an adaptive mechanism to estimate  $\mathbf{R}_j$  at each time step  $j$ . Matrix  $\mathbf{R}_j$  is parameterized by the average azimuthal displacement of  $\boldsymbol{\mu}_j = [\mu_j^x, \mu_j^y, \mu_j^z]^T$  over the time interval  $\Delta T_j$ , i.e.

$$\mathbf{R}_j = \text{Rot}_z(\omega_{\text{mavg}}^\psi \Delta T_j), \quad \omega_{\text{mavg}}^\psi = \frac{1}{\tau} \sum_{i=j-\tau+1}^j \omega_i^\psi, \quad (5)$$

where  $\text{Rot}_z(\cdot)$  denotes an elementary rotation about the  $z$ -axis and  $\omega_{\text{mavg}}^\psi$  is the moving average of the angular velocity  $\omega_i^\psi$  over the time window  $\tau$ . The velocity  $\omega_i^\psi$  is computed via the backward Euler method

$$\omega_i^\psi = \frac{\psi_i - \psi_{i-1}}{\Delta T_i}, \quad \psi_i = \arctan(\mu_i^y / \mu_i^x). \quad (6)$$

Differently from [31], in our case the target does not necessarily lie on a great circle and we chose the axis-angle representation of matrix  $\mathbf{R}_j$ , to guarantee a full coverage of the sphere. Equation (6) is thus modified as follows

$$\omega_i = \frac{\arccos(\boldsymbol{\mu}_i^T \boldsymbol{\mu}_{i-1})}{\Delta T_i} (\boldsymbol{\mu}_i \times \boldsymbol{\mu}_{i-1}).$$

Finally, since our sensor is asynchronous, the intervals  $\Delta T_i$ 's are not necessarily identical. To cope with it, we replaced the moving average in equation (5), with a weighted moving average. For further details on the implementation of the Bayesian estimator, the reader is referred to the GitHub of our project.

To validate the estimator, we performed hardware experiments with a DJI Mavic Air 2 quadrotor, in an urban park. As shown in Fig. 6(c), SFERA was mounted on a tripod, 2 m above the ground level and it was connected to a laptop with an Intel Core i7-13700H processor @ 3.7 GHz and 16 GB of RAM, to process the incoming events. The goal was to detect and track the manually-piloted drone (see the video accompanying the paper). For the ground truth, we took advantage of quadrotor's flight record, which includes the GNSS coordinates, altitude, attitude and velocity from the IMU. Figs. 7 and 8 report qualitative and quantitative results, respectively. Fig. 7(a) shows the GNSS trajectory of the quadrotor (black) in local tangent plane coordinates (ENU: East North Up). The initial (final) position of the drone is marked with a red (green) disk. SFERA is approximately located at (6, 4) m (orange disk). The flight duration was 7 min and 42 s, the maximum altitude of the quadrotor was 7.62 m, the maximum speed 24.09 km/h, and the total distance traveled 527.79 m. The GNSS signal was strong during the flight (22 satellites within line of sight, on average): this guaranteed robust localization, with a position accuracy of about 1 m. In Fig. 7(b), six snapshots, labeled 1 through 6 (spherical camera view), show the performance of the tracker. They correspond to the six locations marked with red boxes in Fig. 7(a). The estimated mean direction  $\boldsymbol{\mu}$  is denoted with a green disk and the clusters of events are depicted in black (a magenta disk indicates their centroid). By and large, the Bayesian estimator is able to robustly track the quadrotor, no matter its pose with respect to SFERA.

To confirm the previous findings, Fig. 8(a) reports the time evolution of the ground truth (black) and estimated mean direction (red) on the unit sphere, for a subset of 45 s of the trajectory shown in Fig. 7(a). To provide a visual reference, this portion of the trajectory is marked in cyan, in Fig. 7(a). The time evolution of the corresponding angular estimation error  $\gamma = \arccos(\boldsymbol{\mu}^T \boldsymbol{\mu}_{\text{GT}})$  between the estimated and actual mean direction  $\boldsymbol{\mu}$  and  $\boldsymbol{\mu}_{\text{GT}}$ , is shown in Fig. 8(b). Note that the peak at  $t \in [157, 163]$  s, is due to the re-initialization of the tracker, which occurs when the

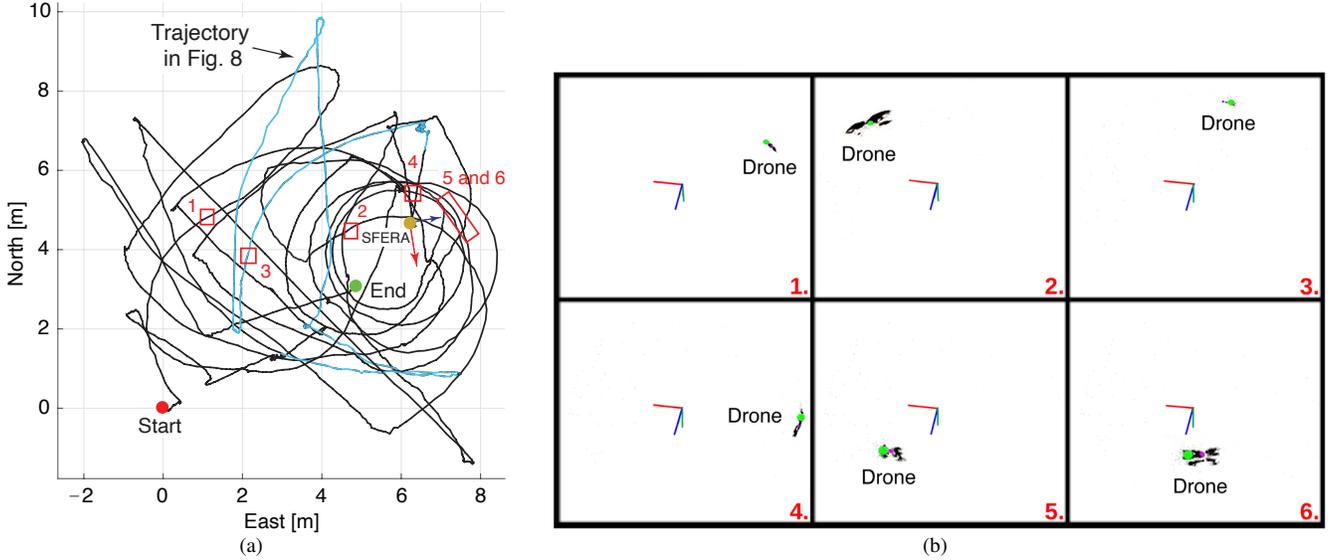


Fig. 7. *Experiments: qualitative results.* (a) GNSS trajectory of the quadrotor in ENU coordinates (meters); (b) Six snapshots (spherical camera view) corresponding to the six locations of the drone shown in Fig. 7(a), see the red boxes. The clusters of events are marked in black and the mean direction  $\mu$  computed by the Bayesian estimator is denoted by a green disk.

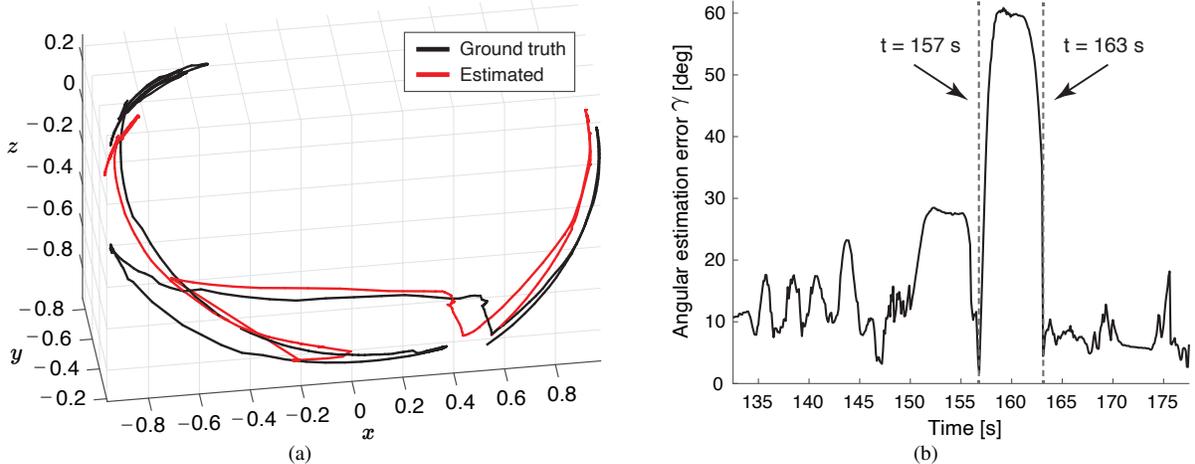


Fig. 8. *Experiments: quantitative results.* (a) Ground truth  $\mu_{GT}$  (black) and estimated mean direction  $\mu$  (red) on the unit sphere, for the portion of the trajectory marked in cyan in Fig. 7(a); (b) Time evolution of the angular estimation error  $\gamma$  of the Bayesian estimator in degrees.

concentration parameter  $\kappa$  drops below a given threshold  $\kappa_{\tau}$  (we set  $\kappa_{\tau} = 50$  in our experiments). In spite of this issue, which will be addressed in future works, the Bayesian estimator runs on a standard CPU in real time (the detection and tracking steps operate asynchronously at a frequency exceeding 50 Hz) and delivers satisfactory results in terms of accuracy: in fact, the mean estimation error is  $12.8^{\circ}$ , outside the interval  $[157, 163]$  s. Moreover, the estimator is robust against the sharp and unpredictable luminosity changes occurring in outdoor unstructured environments.

#### IV. CONCLUSION AND FUTURE WORK

In this paper, we have introduced and fully characterized a novel omnidirectional event camera, called SFERA, which consists of a pair of synchronized fisheye event cameras placed at a fixed distance apart and pointing in opposite directions. We have also presented a Bayesian estimator (tracker), adapted to the specific output of SFERA (a sparse and asynchronous stream of events, projected on

the unit sphere), which has been successfully tested with a DJI Mavic Air 2 quadrotor, via outdoor experiments.

The work presented in this article is a stepping stone towards an ultra-fast, 360-degree, detection and tracking system for aerial surveillance (e.g. of malicious agents in the proximity of airports or sensitive industrial areas). In future research, we plan to further reduce the size and weight of our prototype of SFERA (for example, by using more compact fisheye lenses), a key prerequisite to install it on a swarm of mini UAVs for defense and security missions.

#### ACKNOWLEDGMENTS

The authors would like to thank Prof. I. Marković for sharing the source code of the tracker developed in [31] and H. Midavaine, research engineer at the MIS laboratory, for his help in the mechanical and electrical design of our prototype of SFERA. They are also grateful to Prof. E. Mouaddib for his support in the preparation of the experimental testbed described in Sect. III-C.

## REFERENCES

- [1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A.J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1):154–180, 2022.
- [2] S. Shiba, Y. Aoki, and G. Gallego. Secrets of Event-Based Optical Flow. In *Proc. Europ. Conf. Comput. Vis.*, pages 628–645, 2022.
- [3] C. Reinbacher, G. Munda, and T. Pock. Real-Time Panoramic Tracking for Event Cameras. In *Proc. IEEE Int. Conf. Comput. Phot.*, pages 1–9, 2017.
- [4] J. Bertrand, A. Yiğit, and S. Durand. Embedded Event-based Visual Odometry. In *Proc. 6th Int. Conf. Event-based Contr., Commun. Sign. Proc.*, pages 1–8, 2020.
- [5] P. Chen, W. Guan, and P. Lu. ESIVIO: Event-Based Stereo Visual Inertial Odometry. *IEEE Rob. Autom. Lett.*, 8(6):3661–3668, 2023.
- [6] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza. EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time. *IEEE Rob. Autom. Lett.*, 2(2):593–600, 2016.
- [7] W. Chamorro, J. Solà, and J. Andrade-Cetto. Event-based line SLAM in real-time. *IEEE Rob. Autom. Lett.*, 7(3):8146–8153, 2022.
- [8] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos. Event-Based Moving Object Detection and Tracking. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 6895–6902, 2018.
- [9] Y. Kang, G. Caron, R. Ishikawa, A. Escande, K. Chappellet, R. Sagawa, and T. Oishi. Direct 3D model-based object tracking with event camera by motion interpolation. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2645–2651, 2024.
- [10] H. Chen, Q. Wu, Y. Liang, X. Gao, and H. Wang. Asynchronous Tracking-by-Detection on Adaptive Time Surfaces for Event-based Object Tracking. In *Proc. 27th ACM Int. Conf. Multimedia*, pages 473–481, 2019.
- [11] S. Zhang, W. Wang, H. Li, and S. Zhang. EVtracker: An Event-Driven Spatiotemporal Method for Dynamic Object Tracking. *Sensors*, 22(16):6090, 2022.
- [12] F. Chaumette and S. Hutchinson. Visual servoing and visual tracking. In *Handbook of Robotics*, chapter 24, pages 563–583. Springer, 2008.
- [13] S. Schraml, A.N. Belbachir, and H. Bischof. An Event-Driven Stereo System for Real-Time 3-D 360° Panoramic Vision. *IEEE Trans. Ind. Electron.*, 63(1):418–428, 2016.
- [14] P. Vasseur and F. Morbidi, editors. *Omnidirectional Vision: From Theory to Applications*. John Wiley & Sons, 2023.
- [15] J.P. Kumar, P. Stephanos, and G. Volker. AirFisheye Dataset: A Multi-Model Fisheye Dataset for UAV Applications. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 11818–11824, 2024.
- [16] P. Liu, C. Feng, Y. Xu, Y. Ning, H. Xu, and S. Shen. OmniNext: A Fully Open-source and Compact Aerial Robot with Omnidirectional Visual Perception. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 10605–10612, 2024.
- [17] G. Caron and F. Morbidi. Spherical Visual Gyroscope for Autonomous Robots Using the Mixture of Photometric Potentials. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 820–827, 2018.
- [18] B. Berenguel-Baeta, J. Bermudez-Cameo, and J.J. Guerrero. OmniSCV: An Omnidirectional Synthetic Image Generator for Computer Vision. *Sensors*, 20(7):2066, 2020.
- [19] D. Scaramuzza, A. Martinelli, and R. Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 5695–5701, 2006.
- [20] C. Geyer and K. Daniilidis. A Unifying Theory for Central Panoramic Systems and Practical Implications. In *Proc. Europ. Conf. Comput. Vis.*, pages 445–461, 2000.
- [21] C. Mei and P. Rives. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3945–3950, 2007.
- [22] K. Huang, Y. Wang, and L. Kneip. Dynamic Event Camera Calibration. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 7021–7028, 2021.
- [23] M. Muglikar, M. Gehrig, D. Gehrig, and D. Scaramuzza. How to Calibrate Your Event Camera. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 1403–1409, 2021.
- [24] M. Salah, A. Ayyad, M. Humais, D. Gehrig, A. Abusafieh, L. Seneviratne, D. Scaramuzza, and Y. Zweiri. E-Calib: A Fast, Robust, and Accurate Calibration Toolbox for Event Cameras. *IEEE Trans. Image Process.*, 33:3977–3990, 2024.
- [25] H. Cho, J. Jeong, and K.J. Yoon. EOMVS: Event-Based Omnidirectional Multi-View Stereo. *IEEE Rob. Autom. Lett.*, 6(4):6709–6716, 2021.
- [26] Z. Liu, B. Guan, Y. Shang, Z. Yu, Y. Bian, and Q. Yu. LECalib: Line-based event camera calibration. *Measurement*, 235:114900, 2024.
- [27] E. Dubeau, M. Garon, B. Debaque, R. de Charette, and J.-F. Lalonde. RGB-D-E: Event Camera Calibration for Fast 6-DOF Object Tracking. In *Proc. IEEE Int. Symp. Mixed Augm. Real.*, pages 127–135, 2020.
- [28] S. Wang, Z. Xin, Y. Hu, D. Li, M. Zhu, and J. Yu. EF-Calib: Spatiotemporal Calibration of Event- and Frame-Based Cameras Using Continuous-Time Trajectories. *IEEE Rob. Autom. Lett.*, 9(11):10280–10287, 2024.
- [29] W. Xing, S. Lin, L. Yang, and J. Pan. Target-free Extrinsic Calibration of Event-LiDAR Dyad using Edge Correspondences. *IEEE Rob. Autom. Lett.*, 8(7):4020–4027, 2023.
- [30] M. Cochetoux, J. Moreau, and F. Davoine. MULi-Ev: Maintaining Unperturbed LiDAR-Event Calibration. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 4579–4586, 2024.
- [31] I. Marković, F. Chaumette, and I. Petrović. Moving object detection, tracking and following using an omnidirectional camera on a mobile robot. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 5630–5635, 2014.
- [32] K.C. Liu, Y.T. Shen, and L.G. Chen. Simple online and realtime tracking with spherical panoramic camera. In *Proc. IEEE Int. Conf. Consumer Elect.*, pages 1–6, 2018.
- [33] H. Huang, Y. Xu, Y. Chen, and S.K. Yeung. 360VOT: A New Benchmark Dataset for Omnidirectional Visual Object Tracking. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 20566–20576, 2023.
- [34] M. Flores, D. Valiente, A. Peidró, O. Reinoso, and L. Payá. Generating a full spherical view by modeling the relation between two fisheye images. *Vis. Comput.*, 40:7107–7132, 2024.
- [35] B. Li, L. Heng, K. Koser, and M. Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 1301–1307, 2013.
- [36] K.V. Mardia and P.E. Jupp. *Directional Statistics*. John Wiley & Sons, 2000.
- [37] J. Traa and P. Smaragdis. Multiple speaker tracking with the Factorial von Mises-Fisher Filter. In *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, pages 1–6, 2014.