Stereo Co-capture System for Recording and Tracking Fish with Frame- and Event Cameras

Friedhelm Hamann and Guillermo Gallego
Technische Universität Berlin, Einstein Center Digital Future and SCIoI Excellence Cluster, Berlin, Germany

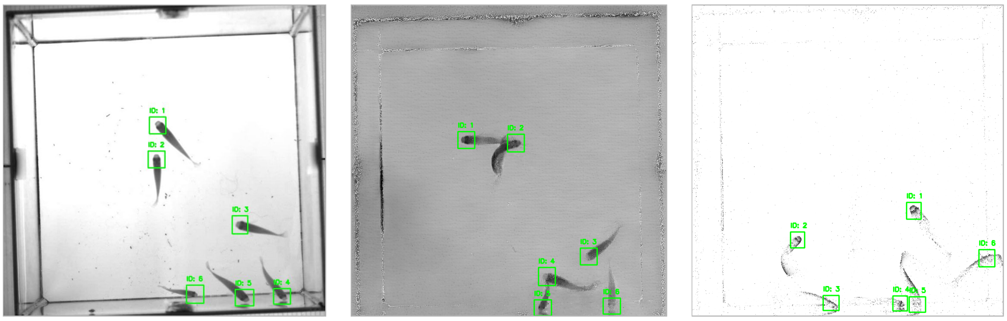

Fig. 1: Visualization of the tracking output on different sensor data. From left to right: tracking on grayscale frames of a conventional camera, on images reconstructed from event data using E2VID [1] and on time maps obtained from event data.

***Abstract*—This work introduces a co-capture system for multi-animal visual data acquisition using conventional cameras and event cameras. Event cameras offer multiple advantages over frame-based cameras, such as a high temporal resolution and temporal redundancy suppression, which enable us to efficiently capture the fast and erratic movements of fish. We furthermore present an event-based multi-animal tracking algorithm, which proves the feasibility of the approach and sets the baseline for further exploration of combining the advantages of event cameras and conventional cameras for multi-animal tracking.**

I. Introduction

Quantification of animal behavior is a critical part of neuro-scientific and biological research. The first step towards quantifying animal behavior consists of tracking the animal's movements. In recent years a multitude of methods have emerged to leverage recent advances in Computer Vision for visual tracking of animals [2]. However, current tracking systems are limited by the capabilities of the used hardware, like sensors, processors and power supply. Some of these limitations can be overcome by the use of event cameras.

Event cameras [3] are bio-inspired sensors that differ from conventional frame-based cameras in the way that visual data is acquired. While frame-based cameras capture images at a fixed frame rate, event cameras measure brightness changes at each pixel independently and output them in the form of an event stream (which encodes the pixel location, time and sign of the brightness changes). The different principle of operation endows event cameras with attractive properties over conventional cameras, such as a very high temporal resolution (µs), a very high dynamic range and low power consumption. We refer to a comprehensive survey for details [4].

The quantification of animal behavior can be performed in a wide variety of ways ranging from observations in natural conditions to experiments in controlled laboratories. Furthermore, there are a multitude of representations of captured movements, which differ in complexity (e.g., centroid tracking vs. 3D animal pose estimation) and number of tracked animals. The wide variety of tracking tasks with their respective modalities places different requirements on the hardware and algorithms. This opens different possibilities for the application of event cameras in animal behavior analysis.

We propose a stereo co-capture system for recording and tracking of animals. Each monocular system consists of a frame-based camera and an event camera, with their views spatially aligned via a beamsplitter. A hardware trigger provides high precision temporal alignment of all camera signals. The system enables a fair comparison with frame-based tracking algorithms and furthermore allows us to perform tracking using both data streams to combine their advantages.

We show an application to fish tracking in a laboratory environment using the stereo co-capture system (Fig. 1). Acquiring the fast and erratic movement of fish with conventional cameras requires high frame rates. The high temporal resolution of event cameras allows us to effectively capture these fast movements. In summary, our contributions are:

- A co-capture system providing temporally and spatially aligned frames and events for animal recording,
- A baseline multi-object tracking algorithm using events, with an application to fish tracking.

II. Related Work

A. Animal Tracking Technology

There is a vast variety of methods and tools available for tracking animals. They can be roughly categorized according to the movement representation and the number of tracked animals. The simplest form of tracking is the description of an animal movement as a trajectory of points or ellipses, for example obtained by background subtraction or thresholding [5]. This method is computationally light and in different variations is widely adopted in many open source tracking tools [6]. However, classical methods like this one fail in more difficult scenarios with complex or dynamic backgrounds and occlusions. The technique can be extended to multiple animals, which introduces the problem of identity assignment. The general task of multi-object tracking usually follows the tracking-by-detection paradigm. In a first step objects are detected in the camera frames; in a second step the detected objects are associated between frames. This decouples the tasks of object detection and data association, allowing researchers to adopt state-of-the-art deep-learning–based object detection methods. Recently, end-to-end learned approaches, like [7] show promising results to further improve tracking accuracy, beyond the tracking-by-detection paradigm.

Generally, the association task can be addressed by modelling the appearance and/or motion of the animals. A simple and widely used approach is described in [8]. A constant velocity model is assumed, to predict motion using a Kalman filter. Bounding box predictions of the next frame are associated according to their intersection over union (IoU) with the predicted bounding boxes of existing trackers using the Hungarian algorithm. The authors of [9] use an offline method, where tracklets over several frames are built and a learned approach is used for animal re-identification. We adopt the algorithm in [8] and extend it for usage on event data.

B. Event-based Object Tracking

Event cameras capture pixel-level brightness changes asynchronously, called events. Assuming constant illumination, events are caused by moving edges [4]. This motivates their use for efficient object tracking. Early approaches follow a blob-tracking [10] or pattern-tracking [11], [12] paradigm. These approaches work on a per-event basis, associating incoming events with existing objects, subsequently updating the position according to the associated events. Similar classical approaches are computationally light but mostly tailored to specific applications. The authors of [13] use a template matching approach to track space objects (satellites, etc.). A second class of event-based tracking algorithms leverages deep-learning–based approaches. In a first step, frame-like representations are obtained from the events, for compatibility with mainstream computer vision methods. Subsequently, methods like CNNs that rely on a grid-like data representation can be used [14]. The signals obtained from event- and frame-based cameras are complementary, therefore a third class of algorithms proposes to jointly use events and frames to solve the tracking task. The authors of [15] extract features from frames and subsequently tracks them asynchronously using events. In [16] cluster-based event tracking is used to generate regions of interest (ROIs); in a second step a CNN is used to classify the region proposals on the frames.

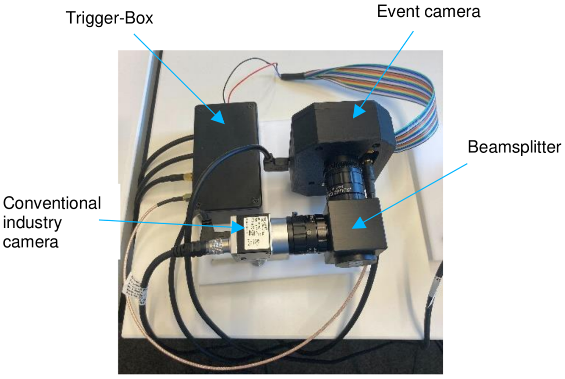

Fig. 2: Part of our co-capture system. Stereo is shown in Fig. 4.

C. Co-capture Systems

In the literature we find cameras that jointly capture frames and events, such as the DAVIS [17], [18]. However, these prototypes have a low spatial resolution and produce low-quality grayscale frames, with a dynamic range of ≈ 55dB. Recently researchers have resorted to building custom sensing devices, using a beam splitter mirror to spatially align the field of views of an RGB and an event camera [19], [20].

III. Co-capture System and Fish Tracking Method

Event cameras offer several interesting properties to overcome limitations of conventional cameras (motion blur, low dynamic range, redundant data in static environments, etc.). Under the constant brightness assumption event cameras capture moving edges, which are very informative footprints for object tracking tasks. However, as is usual in an emerging field, there is a lack of datasets and benchmarks, to evaluate the performance of event-based algorithms.

For this reason we propose using a co-capture system to acquire spatially and temporally aligned (e.g., synchronized) data from event and frame-based cameras. In the following, we describe the co-capture system and a basic algorithm for multi-object tracking with event cameras as well as a frame-based baseline algorithm, for comparison.

A. System Specification and Calibration

The co-capture system consists of an event camera (Prophesee EVK3 Gen4.1, 1280 × 720 pixels), a frame-based camera (Basler acA1300-200um, 1280 × 1024 pixels), a beamsplitter (Plate Bs C-Mount VIS50R/50T) and a custom-build trigger-box. Figure 2 shows the system and its components. Every camera receives a synchronized trigger signal from a microcontroller in the trigger-box. At each rising edge of the *rect*-signal a grayscale frame is acquired and a timestamp is generated in the event camera. Thereby, the frames can be accurately time-aligned with the event data.

The beamsplitter approximately aligns the field of views of both cameras. To achieve a more accurate alignment it is necessary to warp the data from one of the cameras onto the other using the homography between their coordinate systems. To estimate the planar homography we use a standard calibration

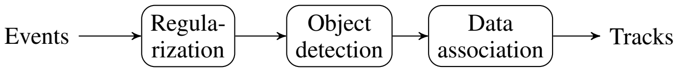

Fig. 3: Processing steps of the event tracking algorithm.

software, obtaining the extrinsic and intrinsic calibration of the two cameras (we used [21]). Afterwards the homography H can be decomposed as [22, p.327]:

$$H = R - \frac{\mathbf{t}\mathbf{n}^{\top}}{d}, \tag{1}$$

where R is the 3×3 rotation matrix, $\mathbf{t}$ is the translation vector between the the two coordinate systems, and $\mathbf{n}$ and d parameterize a world plane of the form $\mathbf{n}^{\top}\mathbf{x} + d = 0$. Since the camera centers are very close to each other, we can assume the ratio $\mathbf{t}/d$ to be small and therefore approximate the homography by its rotational part $H \approx R$. The homography is mapping from one pixel-domain to the other.

B. Tracking Method

To validate the approach and the comparability of the data, we propose a baseline algorithm to perform event-based multi-animal tracking. Our method uses a classical tracking-by-detection approach, combined with common CNN-based object detectors. The basic pipeline is depicted in Fig. 3.

Event-based cameras detect pixel-independent brightness changes. Specifically, they output an event $e_k = (\mathbf{x}_k, t_k, p_k)$ whenever the logarithmic intensity $L(\mathbf{x}, t) = \log I(\mathbf{x}, t)$ changes by a certain threshold. Where $\mathbf{x} = (x, y)^{\top}$ is the pixel position, t is a timestamp, typically in microsecond resolution and $p_k \in \{-1, +1\}$ signals if the brightness change was positive or negative.

In the first step of the pipeline (Fig. 3), we compute time maps $T(x, y)$ from the event stream [23], where each pixel in this time map stores the timestamp of the latest event which occurred at that pixel location. This step adapts the events into a grid format that is compatible with a large body of algorithms designed for image-based data. Another advantage of this representation is that it can be updated asynchronously on every event and therefore in principle enables processing without loosing temporal accuracy. Similarly to [23], we apply a pixel-wise exponential decay τ of the form

$$I_i(x, y, t_i, p) = e^{-(t_i - T(x,y,p))/\tau} \tag{2}$$

where t_i is the current timestamp. For each polarity one timestamp map is created, resulting in two output maps, which will be the input channels for the next processing step.

After computing the time maps, we apply modern CNN approaches on this representation. We use the latest implementation [24] of [25], obtaining n bounding boxes for each frame. Subsequent tracking is performed using [8] (see II-A).

To compare the methods, the same approach is tested with reconstructed [1] and grayscale frames, using the same object detector and tracking algorithm. For each of the three representations a separate detector is trained using transfer learning with a small hand-labelled dataset of ≈ 150 snapshots.

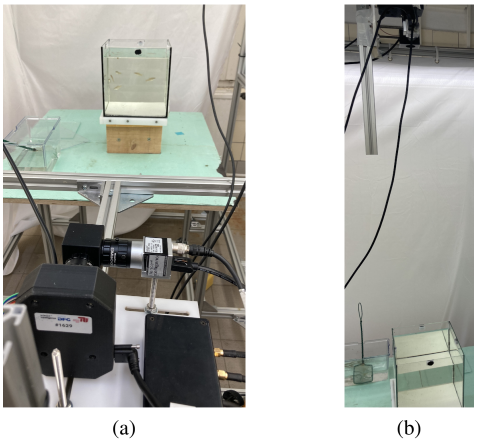
(a) (b)

Fig. 4: The setup for the fish recordings. One co-capture system with front view (a), one with top view (b).

IV. Experiments

A. Data Acquisition

To show the capabilities of the co-capture system we recorded live fish (*Poecilia formosa*) during ongoing experimental work in the laboratory of Prof. Jens Krause. Two synchronized co-capture systems were used to record fish in a water tank, using one co-capture system from the front and one from the top (see Fig. 4). The method currently presented uses only the top-view recordings. Stereo extensions are planned: tracks from both views will be fused using an EKF to reconstruct the 3D trajectories of the animals.

We recorded 12 sequences with 1 to 6 fish. The Basler camera recorded at 120 fps. The event cameras delivered an average event rate of 675 thousand events/s. The fish were located in a tank of size 20×20×20 cm. Figure 5 shows the overlay of events and grayscale frames. A visualization of the top view in the three different representations and tracking methods tested can be seen in Fig. 1.

B. Results and Discussion

The goal of the approach is (i) to compare tracking algorithms working on event-based and frame-based data and (ii) to combine events and frames to increase tracking accuracy. The asynchronous time maps allow us to increase the frequency at which the trackers are updated up to µs accuracy. The introduced tracking algorithm shows the feasibility of the approach. Table I reports the mean average precision of the object detector trained on the different representations and evaluated on a hand-labeled validation dataset. The training sets for the grayscale frame and the time-map detector are identical, in the sense that the time maps were queried at the times of the frames and identical annotations were used. Furthermore, Tab. I presents the average tracklet length of the three approaches validated on three recorded sequences containing 1 to 3 fish. With more than 10 seconds of average tracking time, the trackers are stable.

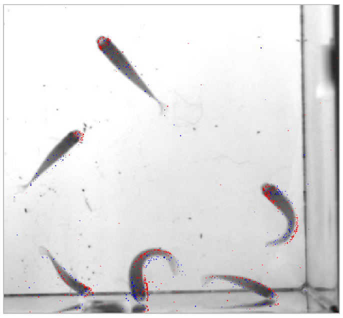

Fig. 5: Event-data visualized overlaid on the grayscale frames. The blue and red dots represent positive and negative brightness changes (events), respectively.

	gray frames	E2VID	time-surface
$mAP_{.5:.05:.95}$	0.6169	0.4936	0.4224
Avg. tracklet time [s]	20.42	16.28	14.11

TABLE I: Mean-average precision and average tracklet time of the object detectors trained on the three different input data.

The *mAP* and the tracklet time in the conducted experiment are lower for both event-based representations compared to the gray-scale frames. However, the preliminary results serve as a proof of concept for the chosen approach. They show that event-based multi-animal tracking following the tracking-by-detection paradigm is possible. This sets the baseline for further exploration of event-based tracking. The classical CNN-based object detectors do not exploit the sparse nature of event-data. This motivates the adoption of event-based object detectors for animal tracking.

V. Conclusion

We have presented a co-capture system for recording and tracking animals using frames and events. We have also described a baseline algorithm to use the event data for multi-animal tracking, which provides the base for qualitative comparison and development of advanced tracking algorithms combining the strengths of both sensor types. With asynchronous object detectors the event data can be used for computationally efficient tracking, to capture very fast movements or to perform tracking under challenging lighting conditions. We plan to extend and test our method to more challenging scenarios, such as long-term observation of wildlife animals.

Acknowledgment

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135. We thank Prof. B. Shi's lab, who helped selecting the beamsplitter and lenses of the system and Prof. J. Krause's lab for providing the fish tank setup. All animal care and experimental protocols complied with local and federal laws and guidelines, and were approved by the appropriate governmental authorities (LaGeSo G-0089/21 granted to David Bierbach).

References

[1] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *TPAMI*, 2019.

[2] T. D. Pereira, J. W. Shaevitz, and M. Murthy, "Quantifying behavior to understand the brain," *Nature Neurosc.*, vol. 23, no. 12, pp. 1537–1549, 2020.

[3] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[4] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *TPAMI*, 2020.

[5] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson, "High-throughput ethomics in large groups of drosophila," *Nature Methods*, vol. 6, no. 6, pp. 451–457, 2009.

[6] V. Panadeiro, A. Rodriguez, J. Henry, D. Wlodkowic, and M. Andersson, "A review of 28 free animal-tracking software applications: Current features and limitations," *Lab animal*, vol. 50, no. 9, pp. 246–254, 2021.

[7] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," *arXiv preprint arXiv:2105.03247*, 2021.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016, pp. 3464–3468.

[9] F. Naiser, M. Šmíd, and J. Matas, "Tracking and re-identification system for multiple laboratory animals," in *Visual observation and analysis of vertebrate and insect behavior workshop at ICPR*, 2018.

[10] M. Litzenberger, A. N. Belbachir, N. Donath, G. Gritsch, H. Garn, B. Kohn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *ITSC*, 2006, pp. 653–658.

[11] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Computation*, vol. 27, no. 4, pp. 925–953, 2015.

[12] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *TNNLS*, vol. 26, no. 8, pp. 1710–1720, Aug. 2015.

[13] S. Afshar, A. P. Nicholson, A. van Schaik, and G. Cohen, "Event-based object detection and tracking for space situational awareness," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 15 117–15 132, 2020.

[14] M. Iacono, S. Weber, A. Glover, and C. Bartolozzi, "Towards event-driven object detection with off-the-shelf deep learning," in *IROS*, 2018.

[15] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "EKLT: Asynchronous photometric feature tracking using events and frames," *IJCV*, vol. 128, pp. 601–618, 2020.

[16] H. Liu, D. P. Moeys, G. Das, D. Neil, S.-C. Liu, and T. Delbruck, "Combined frame- and event-based detection and tracking," in *ISCAS*, 2016, pp. 2511–2514.

[17] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[18] G. Taverni, D. P. Moeys, C. Li, C. Cavaco, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Front and back illuminated Dynamic and Active Pixel Vision Sensors comparison," *TCS-II*, vol. 65, no. 5, pp. 677–681, 2018.

[19] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging," in *CVPR*, 2020, pp. 1606–1616.

[20] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, "Event-aided direct sparse odometry," in *CVPR*, 2022.

[21] "Basalt calibration software," https://gitlab.com/VladyslavUsenko/basalt, accessed: 2022-05-16.

[22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, 2nd Edition.

[23] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *TPAMI*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[24] "YOLOv5," https://github.com/ultralytics/yolov5, accessed: 2022-05-16.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.