Analysis of event-based camera signals for Safety integration to IEC61499*

T. Ladune, M. Colas, A. Philippot, B. Riera, F. Gellot, P. Hampikian

Abstract— **Industry 4.0 merges digital and physical systems, creating cyber-physical production systems (CPS) that require adaptability and modularity. Traditional PLCs, using IEC 61131-3, limit reconfigurability. The IEC 61499 standard enhances portability, configurability, and interoperability, gaining traction with platforms like Schneider Electric's EcoStruxure Automation Expert. Event-driven cameras, using neuromorphic technology, encode visual changes asynchronously, improving processing efficiency. However, their integration into IEC 61499 systems is rare. This paper introduces an interface linking event-driven cameras with IEC 61499 applications, exploring concepts, architecture, and experimental results.**

I. Introduction

In the context of Industry 4.0, production systems tend to implement more and more information and communication technologies. This incursion of information technologies inside production systems into physical systems blurs the boundary between the digital and physical worlds. Production systems have become cyber-physical (CPS) [1].

In order to meet consumer needs, CPSs have to be able to produce a large variety of product over several generations. CPSs are dynamical elements of the production environment. They require adaptivity. To do so, they integrate different cooperating heterogenous and autonomous elements. This disparity of physical architecture makes CPSs particularly adapted to modular programming [1].

As of today, most command systems are programmed using the languages described in the IEC 61131-3 standard [2]. However, the numerous incompatibilities between the different programmable logical controllers (PLCs) limit the adaptability and reconfigurability of the CPSs.

That is why the IEC 61499 standard series was created [3]. It facilitates also the development of distributed control systems by enabling the application deployment on a set of controllers sharing a common network (figure 1). The IEC 61499 series aims at standardizing:

- Software components and configurations to be accepted by different developing software (portability).
- The interconnection and configuration of software components and hardware platforms (configurability).
- The common operation of the devices (interoperability)

The object-orientation of the FB's combined with the hardware independence allows you to assemble FB's from different programmers that will interoperate in one application, and to run them on any IEC 61499 compliant hardware, ie the software is portable.

Although the first edition of this standard was released in 2005 its interest grows back in industrial and academic field due to the releases of industrial platform implementing the IEC 61499 such as EcoStruxure Automation Expert (EAE) by Schneider Electric [4].

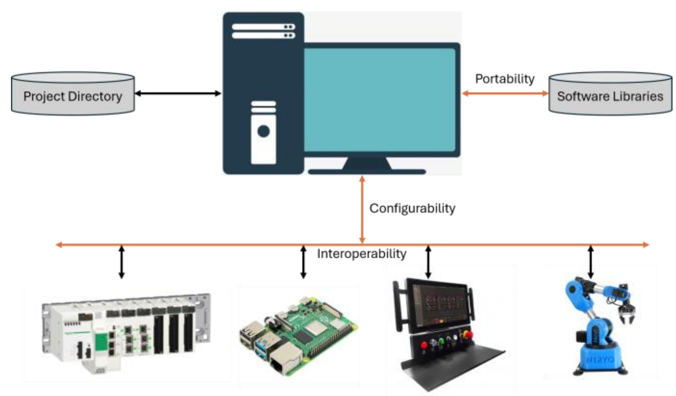

Figure 1. IEC 61499 standard goelas

However, these newer platforms are more and more considered as mature as the ones using the standard series IEC 61131 and takes now a real place in process automation market. Nevertheless, their functional enhancements, such as the integration of functional safety or motion control, are currently being marketed and will enable all the advantages

*Research supported by ANRT Agency.

T. Ladune is with the University of Reims Champagne-Ardenne, CReSTIC, Reims, France and Schneider Electric (e-mail: tom.ladune@{se.com/etudiant.univ-reims.fr}).

M. Colas is with the University of Reims Champagne-Ardenne, CReSTIC, Reims, France (e-mail: maxime.colas@univ-reims.fr).

A. Philippot is with the University of Reims Champagne-Ardenne, CReSTIC, Reims, France (corresponding author to provide e-mail: alexandre.philippot@univ-reims.fr).

B. Riera is with the University of Reims Champagne-Ardenne, CReSTIC, Reims, France (e-mail: bernard.riera@univ-reims.fr).

F. Gellot is with the University of Reims Champagne-Ardenne, CReSTIC, Reims, France (e-mail: francois.gellot@univ-reims.fr).

P. Hampikian is with Schneider Electric, France (e-mail: pascal.hampikian@se.com).

of event-driven management to be exploited for a greater range of industrial automation applications.

At the same time, a new type of vision sensor has emerged, the event-driven camera. Based on a neuromorphic approach, it encodes information from a scene not at a fixed frequency like a conventional camera, but by events that represent a change in luminance in the scene. This asynchronous representation of visual information offers substantial gains in image processing, opening a whole new range of applications.

Due to their recent commercialization, there are few examples in the IEC 61499 literature of camera integration, let alone event-driven cameras. With a view to using this type of camera for production system control or safety-related functions, this paper presents an initial interface between an event-driven camera and an IEC 61499 application.

This paper describes the integration of an event-driven camera into industrial production systems governed by the IEC 61499 standard. Unlike conventional cameras, these neuromorphic sensors record only changes in brightness, which reduces the computational load and improves responsiveness. The study proposes software architecture using the OPC UA protocol to connect this innovative vision system to distributed controllers. This approach aims to enhance adaptability and collaborative safety within the context of Industry 4.0.

We describe the concepts of IEC 61499 and event-driven cameras in Section 2. In Section 3, we present the signal identification architecture. In Section 4, an experimentation illustrates the works before to conclude the paper.

II. Background

A. IEC 61499

Nowadays, most of the production system are programmed using the IEC 61131-3 languages which suffer from a monolithic architecture. In addition to flexibility difficulties caused by a lowly modular architecture, the IEC 61131-3 programs suffer from a lack of compatibilities between different implementations despite standardization initiative [5]. In most cases, the ladder diagram program from a PLC manufacturer remains unimplementable on device from another manufacturer.

The IEC 61499 standard is emitted by the International Electrotechnical Commission and aims at promoting portability, configurability, and interoperability of development platforms [3].

The IEC 61499 propose an event-based execution of the automation applications. Unlike the ones programmed using the IEC 61131-3 languages whose code is periodically and entirely executed in event-based programming only the required part of code is run based on the input changes which reduce the CPU load on the PLC. In particular, as long as no events are triggered no code is running.

The execution of IEC 61499 application being exclusively event-driven, the function blocks composing the applications are composed of event to which can data be associated to. To some extent, events act like function calls which means the data can change more than once before being processed and only the last value will be taken into account.

The function blocks (Figure 2) are activated when an event is triggered and updates the input data associated. These blocks can be of different natures and have various behavior:

- Basic function blocks are composed of state machine, the execution control chart (ECC) which calls algorithms that processes the data and can trigger output events.
- Composite function blocks encapsulate other blocks.
- Service function blocks whose behavior is hidden manage system inputs, outputs, and network interfaces.

As mentioned in introduction the production systems tend to integrate several controllers. Therefore, IEC 61499 defines an architectural model of the system separated from the application model. This enables the distribution of applications on separated controllers.

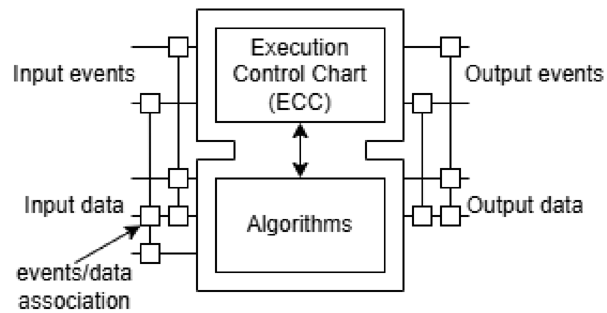

Figure 2. IEC 61499 Basic Function Block

B. Event-based Camera

A conventional camera consists of a set of sensors—pixels—that cyclically integrate the light intensity they receive, followed by digital conversion. This cycle can be performed simultaneously for all pixels (full-frame) or line by line (progressive scan). The integration time remains the same regardless of the technique used. The sensor's output is a dense array of numerical values—an image—that represents the average luminance over the integration time. This cyclic capture of motion results in stationary and dynamic areas being sampled in the same manner. Thus, outside its optimal operating range, the camera oversamples stationary parts or undersamples moving parts.

The event-based camera are sensors based on a biological approach of vision [6]. Contrary to conventional digital cameras, which samples light intensity at fix period, event-based camera measure light intensity changes asynchronously. The information obtained is a 4-uplets (x, y position, polarity and timestamp).

As each pixel generates an event independently on illuminance change the temporal resolution can be much smaller than conventional cameras. This highly reduce sensitivity of sensor to motion blur. Furthermore, in absence of light change no event is generated which reduce the data stream and thus process time.

In addition, threshold levels being proportional to illuminance logarithm the dynamic range can reach value superior to 120dB.

Event-driven cameras represent a paradigm shift in visual information and require the development of new algorithmic and hardware methods to retrieve data and extract information from it. Event-driven cameras produce data that differs from that of conventional cameras in three aspects: output format, photometric measurement, and noise and dynamic effects. The data output format of conventional cameras is synchronous and dense, whereas that of event cameras is asynchronous and sparse. These two differences prevent the direct application of image processing algorithms.

Three approaches based on machine learning are mainly used for detection purposes to process the event stream such as grey scale image reconstruction, dense representation mapping or spiking neural network.

The first technique consists in the reconstruction of a grey-scale video [5, 6, 7, 8]. Once the images are computed, they are treated by a learning machine model made for conventional cameras. This approach leverages over the maturity of computer vision tools. Nevertheless, this results in a less efficient and less accurate detection than non-reconstructive approaches.

The second technique consists in densifying the events stream before processing it into a neural network [8, 9, 10, 11]. The densification can though delete elements due to a compensation of their polarity.

The third group of approach such as presented in [12, 13, 14] consists of using neural network which can process sparse data.

III. Architecture

The architectural options presented in this section have been selected with a view to enabling future integration within a safety function (Figure 3).

EAE is Schneider Electric's development tool based on IEC 61499, which includes a library implementing the OPC UA communication protocol on the client side.

In the first subsection, we describe the hardware on which the event processing application runs. In the second, we detail the software architecture deployed on the IPC (industrial PC) used. Finally, in the third, we integrate the EAE application—which processes the data generated by the event processing—with the OPC UA server.

A. *Industrial PC: NexCobot SCB100*

The Nexcobot SCB100 is an industrial PC (IPC) certified to SIL 2 functional safety standards by TÜV SÜD [15]. This IPC is equipped with an Intel Atom® x6427FE processor. With the aim of developing an IEC 61499-compliant IPC that is SIL2 certifiable and capable of running a critical programme and a non-critical programme in parallel, the SCB 100 is equipped with the SIL3-certified PikeOS operating system and hypervisor.

As our application is not critical in its current version, we will run it in a non-critical virtual machine, ElinOS. Furthermore, to facilitate the development of the application and its network interfacing, we will containerise the application and its dependencies using Docker. This final part of the solution cannot be retained in a future certifiable version.

The event camera used will be a Prophesee EVK4 camera, which interfaces with the IPC via USB 3.0 [16].

B. *Soft dPAC*

Soft dPAC is the name of Schneider Electric's IEC 61499 runtime platform for industrial PCs. It works in conjunction with the EAE development tool, which implements communication protocols such as OPC UA, MODBUS and MQTT, either natively or via function blocks. In our case, the IEC 61499 application must initialise an OPC UA client to establish a session with the server and send the request to retrieve the data stored on that server. The IEC 61499 application consists of:

- An initialization function block;
- A service block for connecting to the OPC UA server;
- An OPC UA read service block

C. *Sofware Architecture*

In order to rapidly develop a proof of concept (POC), the Python programming language was chosen over C++. The Prophesee libraries for camera interfacing and data processing, the asyncua library implementing the OPC UA protocol, and their dependencies are installed within a container.

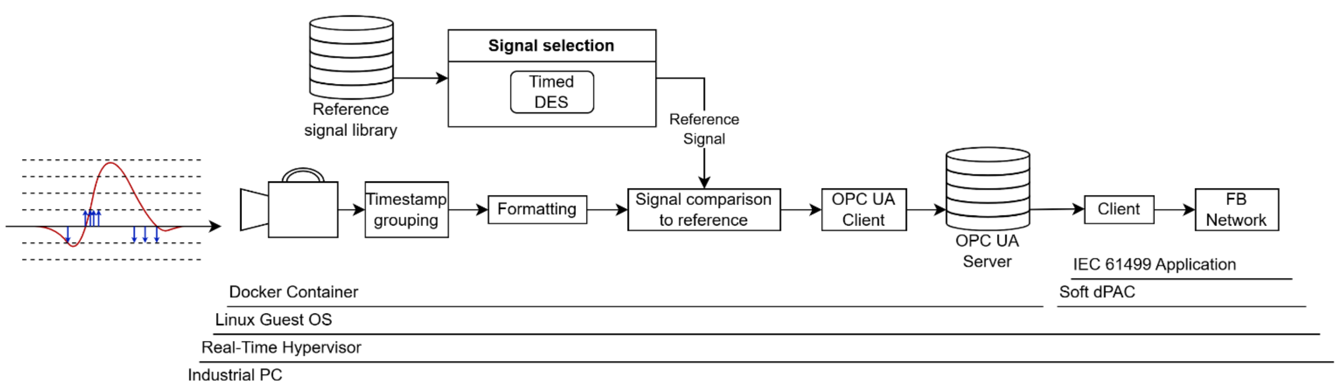

Figure 3: Signal identification architecture

The defined POC consists of creating a counting function based on the information collected by the event camera. To achieve this, the various components—namely the counter, the client and the OPC UA server—are organized around an observer design pattern, as shown in Figure 4.

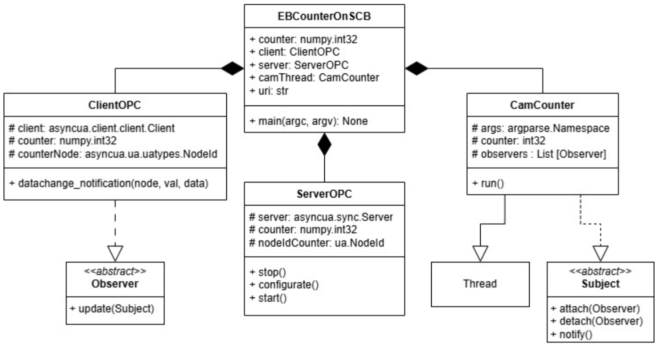

Figure 4. Software Architecture – UML Class Diagram

This design pattern is an evolution of another software engineering technique known as polling, which involves periodically checking for updates to a piece of data. Firstly, polling introduces latency in the retrieval of data. The impact of this problem can be reduced by sampling at a higher frequency. However, this results in a large number of unnecessary processing operations. Polling requires a trade-off between latency and CPU load. This trade-off can be difficult to manage, which is why we prefer the Observer design pattern here

This design pattern allows a method to be called on an Observer object that has been previously registered in a notification list. In our case, we notify the ClientOPC every time the counter is incremented so that the value available on the server is updated.

IV. EXPERIMENTATION

The aim of this section is to illustrate the counting function using two examples. The Prophesee library, used to process the event stream and perform the counting, offers numerous algorithm configuration parameters:

- The minimum object size helps to reduce the number of false positives;
- The distance of objects from the camera;
- The average speed of the projectiles;
- The number of lines;
- The position of the lines;
- The types of events considered

A. *Experimental Setup*

The experimental setup designed to characterize the function consists of the system described in the previous section, with the exception of the use of a Python recording script. This change allows us to apply different processing configurations to the same set of events, thereby ensuring greater usability of the results.

Once the acquisition file has been generated, it is extracted from the SCB 100 to speed up processing, then processed by Prophesee's counting algorithm, to which various configuration profiles are applied.

The scenario presented to the camera involves the fall of a roughly spherical aluminum projectile with an irregular texture and shape.

B. *Formatting the input signal*

The returned format from the event-based camera is a list of 4-uplets (x, y, t, p) grouped in slices of fixed timespan or number of events. The concatenation of slices makes what will be called signal. As the events are issued by ascending timestamp, they can therefore be grouped inside same timestamp unit.

After that comes a step of formatting that reduces the number of events to treat by merging the identical event within a time window. This degradation of the temporal resolution turns the events into 5-uplets (t,x,y,p_+,p_-) with p_+ and p_- being the number of events of each polarity that occurred within the time window (Figure 5).

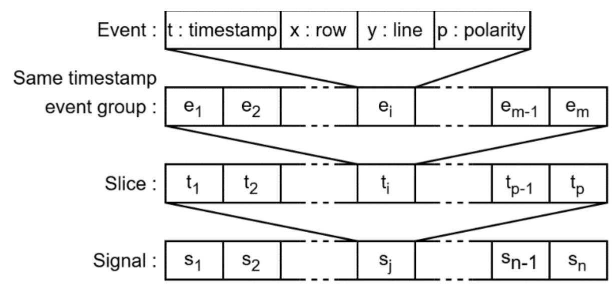

Figure 5. Structure of the event stream

Though it could be possible to identify a signal to a model made from a unique other signal, the uncertainty would not be determined, resulting in a useless measure. The modelling process described here follows steps.

- Extract the behavior from a real signal.
- Degrade the time resolution.
- Intercorrelate the signals.
- Based on the intercorrelation maxima, minimize the segment addition postulate relation error
- Compute the average of the signal and its variance over time

The activity is a measure mainly used in radiological fields. It is defined by the number of events counted per second. As the event-based camera returns events with their occurrence time, it is possible to determine the activity of a signal by counting the number of events triggered in a temporal window.

Once the activity function is computed it becomes easier to identify the behaviors as they pop out of the background noise. They can be extracted from the signal either manually or using statistical methods.

This measure is particularly useful for identifying behaviors against background noise for the following reasons:

- Behavioral Emergence: Once the activity function is computed, specific behaviors become easier to identify because they "pop out" of the background noise.

- Extraction Methods: Because these behaviors create distinct signatures in the activity data, they can be extracted from the overall signal either manually or through statistical methods.
- Disturbance Filtering: The paper notes that while background variations might cause events, actual objects (like the projectiles in the experiment) act as disturbances that create a high concentration of events (specifically positive events at their center), which allows them to be distinguished from the more random or canceling nature of background noise.

These extracted signals represent the same behavior although they do not have the same length.

An event-based signal can be seen as a 3-dimensional function that map to each 3-vector a 2-vector (p_+, p_-), the polarity.

The space of signals is composed of all the possible signal returned by an event-based camera of sensor dimension $x_M \times y_M$ and acquisition of duration t_M. Each polarity of a signal of duration t_M can then be expressed as a $x_M \times y_M \times t_M$-vector of positive integers. Let consider a population of signal representing the same behavior of a production system $\{X_1, X_2, \ldots, X_N\}$.

It is then possible to compare two signals using a scalar product, even though their length differs. Which is why cross-correlation over time is used. This gives table of correspondence of the signals two by two:

$$\begin{bmatrix} \mathcal{R}_{11}(\tau) & \mathcal{R}_{12}(\tau) & \mathcal{R}_{13}(\tau) & \cdots & \mathcal{R}_{1N}(\tau) \\ \mathcal{R}_{21}(\tau) & \mathcal{R}_{22}(\tau) & & & \vdots \\ \mathcal{R}_{31}(\tau) & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \mathcal{R}_{N1}(\tau) & \cdots & \cdots & \cdots & \mathcal{R}_{NN}(\tau) \end{bmatrix}$$

The maximum of correlation can be interpreted as the point when the two signals best "fit" together. The shifts of the maximum $((\tau_{Mij})_{i,j \in [\![1;N]\!]})$ must follow a relation analog to the segment addition postulate (e.g. $\tau_{M13} = \tau_{M12} + \tau_{M23}$). Due to uncertainties of measure, this relation is not verified unless adding an error. This error must be minimized before computing the average of the signal. The shifts of the maximum give an N*N matrix.

Computing the average signal is done naturally by summing each component of the vector shift from the τ_{ij} given at the end of the error minimization. If shift value is not an integer, then the component value is shared pro rata on temporal previous and next components. The sum signal is then divided by the size of the population. Variance can then be easily computed.

C. Qualitative analysis

Figure 6 shows five projectiles falling, illuminated by a ceiling light. The first projectile is already leaving the frame, the next two have passed through the counting zone, and the last two are partially overlapping.

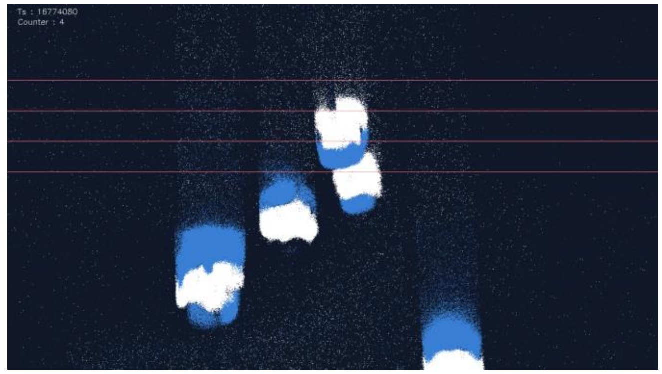

Figure 6. Event cam detection

For part of their trajectory, the projectiles are preceded by a darkened area, followed by a lighter area, and then by another darkened area. In the upper part of the image, only the lower darkened area is visible; this then disappears, giving way to an upper darkened area. The influence of the background may be a factor in this observation. Indeed, the darkening preceding the projectiles could be the effect of their own shadows.

The apparent disappearance of this shadow may be the result of a variation in the brightness of the scene's background.

Certainly, regardless of which pixel is disturbed, if this disturbance lasts for a finite duration and if the variations in brightness are not too rapid, events of positive and negative polarity cancel each other out regardless of their order of appearance.

Thus, by treating the projectiles as disturbances in the background and noting that these have a high concentration of positive events at their center, it seems reasonable to expect greater counting efficiency if only positive events are considered; a proposition that holds true and which, in the case of Figure 6, allows the two projectiles that are touching to be separated and counted as two distinct objects.

When configured with the scene's geometric and kinetic values, the counting algorithm enables the detection and counting of projectiles regardless of the number of detection lines or their position. Establishing the right measurement conditions is also essential. Lighting to avoid unwanted shadows, the trajectory of the projectiles to ensure they do not collide with one another, and their lateral position within the field of view to avoid overlap are all parameters that must be taken into account to ensure the camera counts correctly

V. Conclusion

The technologies brought about by the Fourth Industrial Revolution, alongside the emergence of new control architectures that meet requirements for interoperability and distributed systems, enable us to envisage functional safety that is barrier-free and collaborative.

We have therefore integrated an event-driven camera into an IEC 61499 application via an OPC UA server to perform object counting. With a view to using this camera for control purposes, we have characterized this function to verify the camera's ability to perform safety functions in the future.

This camera and this architecture have the potential to be the subject of further study and integration as a means of monitoring or controlling physical components. To this end, particular consideration should be given to switching to a publish-subscribe communication mode, implementing more advanced identification of moving objects (type, speed, trajectory, etc.) or the choice of programming language.

ACKNOWLEDGMENT

This research work is funded by the ANRT and is carried out as part of a CIFRE thesis in partnership with Schneider Electric.

REFERENCES

[1] [1] Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. Business & Information Systems Engineering, 6(4), 239-242. https://doi.org/10.1007/s12599-014-0334-4

[2] [2] Dai, W. W., & Vyatkin, V. (2009). A case study on migration from IEC 61131 PLC to IEC 61499 function block control. 2009 7th IEEE International Conference on Industrial Informatics, 79-84. https://doi.org/10.1109/INDIN.2009.5195782

[3] [3] Christensen, J. H., Strasser, T., Valentini, A., Vyatkin, V., Zoitl, A., Chouinard, J., Mayer, H., & Kopitar, A. (2012). The IEC 61499 function block standard: Software tools and runtime platforms. ISA Automation Week, 2012.

[4] [4] M. Cannici, M. Ciccone, A. Romanoni and M. Matteucci, "Asynchronous Convolutional Networks for Object Detection in Neuromorphic Cameras," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 2019, pp. 1656-1665, doi: 10.1109/CVPRW.2019.00209.

[5] [5] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. J. Solid State Circ, 43:566–576, 2008.

[6] [6] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 884–892, 2016.

[7] [7] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. International Journal of Computer Vision, 126(12):1381– 1393, 2018.

[8] [8] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[9] [9] Inigo Alonso and Ana C Murillo. Ev-segnet: semantic segmentation for event-based cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.

[10] [10] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Eventbased vision meets deep learning on steering prediction for self-driving cars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5419–5427, 2018.

[11] [11] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 989–997, 2019.

[12] [12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[13] [13] Jia Li, Feng Shi, Wei-Heng Liu, Dongqing Zou, Qiang Wang, Paul KJ Park, and Hyunsurk Ryu. Adaptive temporal pooling for object detection using dynamic vision sensor. In BMVC, 2017.

[14] [14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[15] [15] NexCOBOT, SCB100_Safety Manual_V0.66_2024-03-25.pdf, 06/09/2021, 25/03/2024 [13/02/2025], https://spectra.de/media/g0/7f/ca/1717574121/SCB100_Safety%20Manual_V0.66_2024-03-25.pdf

[16] [16] PROPHESEE, EVK4-HD-Prophesee-Evaluation-Kit-Camera-Manual-1.1.pdf [en ligne], 06/01/2023, 06/01/2023 [13/02/2025], https://www.prophesee.ai/wp-content/uploads/2023/01/EVK4-HD-Prophesee-Evaluation-Kit-Camera-Manual-1.1.pdf.