Object detection using sparse data representation with convolutional neural networks for event-based cameras

Eduardo Borges Gouveia
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0003-2135-3844

Gustavo Ferreira Tavares
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0003-2192-466X

Leandra Lima de Almada
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0003-3655-8850

Andrei Nakagawa-Silva
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0001-8703-2248

Eber Lawrence de Souza Gouveia
Materials research institute
Tech. University of the Shannon
Athlone, Ireland
ORCID: 0000-0003-3766-2043

Márcio José da Cunha
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0002-4173-8031

Edgard Lamounier Afonso Júnior
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0001-6293-9521

Alcimar Barbosa Soares
Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia
Uberlândia, Brazil
ORCID: 0000-0003-1100-3533

***Abstract*— We present an object detection model using deep learning for event-based cameras in real-time. Event-based cameras are bioinspired devices able to capture lighting changes asynchronously at a high temporal resolution, high dynamic range, and low power consumption. Due to its event-based nature, the information acquired by those devices is very different from standard intensity images acquired by traditional cameras making the conventional detection methods not well suited for event-data. We first introduce a labeled dataset containing records of seven objects made with a Dynamic Vision Sensor (DVS128), then we run three different approaches to convert the sparse data generated by the DVS128 and use these data to train a deep learning model based on You Only Look Once (YOLO) for detection of objects to perform tracking in complex environments using an event-based camera in high temporal resolution. Our results show that any of the three approaches generate a good tracking model for event-data, however, not all three are suitable for real-time applications.**

Keywords — event-based camera, detection, YOLO

I. Introduction

Event-based cameras (or silicon retina) are bioinspired devices that work in an asynchronous way capturing events of illuminance changes in a scene like the spiking neurotransmission within biological visual pathways [1]–[3]. Those devices make part of a new field of technology called Neuromorphic Engineering that aims to develop new bioinspired computational approaches to overcome traditional challenges of robotics [4]–[6].

The information acquired by the event-based cameras is different from standard intensity images. While standard cameras acquire the light intensity of a scene in a clock-based way, in event-based cameras each pixel works asynchronously acquiring variation of luminosity (Fig. 1). These characteristics guarantee advantages that consist of a high dynamic range, high temporal resolution, and low latency, and low power consumptions [1]–[3], [7].

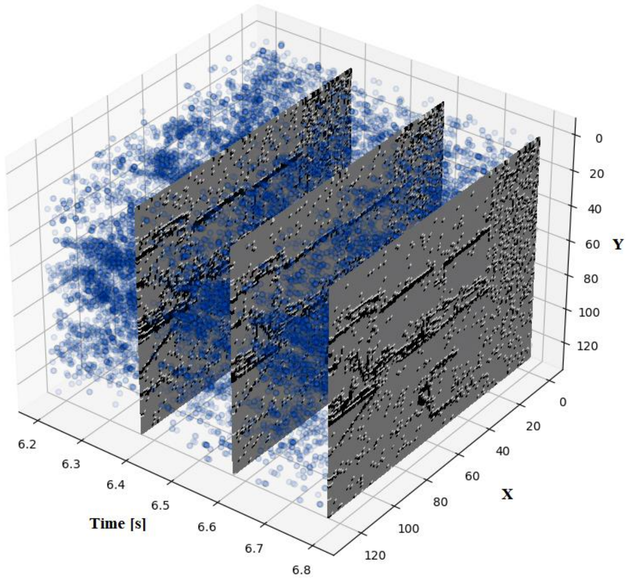


Fig. 1. Stream of events in space-time represantation generated by a mug in a dynamic camera environment.

Because of the sparse nature of the data acquired by the event-cameras, novel computational approaches must be developed to solve traditional computer vision tasks that are already surpassed in traditional frame images. Each event acquired by the Dynamic Vision Sensor can be defined as

$$e_i = [S_i(x,y), t_i, P_i],\ i \in N^* \qquad (1),$$

where e_i is the i^{th} event in the stream of events and carries three basic information: the location at $S_i(x,y)$ at which contrast variation has occurred, the time t_i and the polarity P_i, with $P_i \in \{-1,1\}$, where – 1 and 1 represent the events of decrease and increase of luminance, respectively.

To use conventional deep learning techniques on such sparse data from the event-based cameras some works have focused in convert the event-data to a dense representation

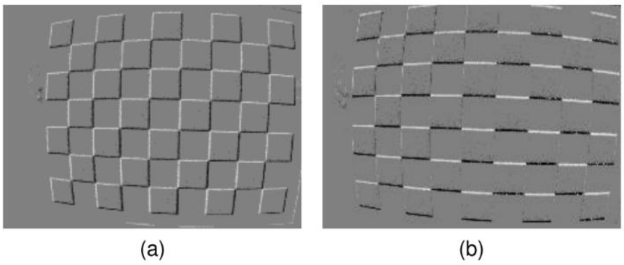


Fig. 2. Representation of the data association challenge. Both representations (a and b) are from the same object, however the motion of the event-based sensor are different. In (a) the motion is in the diagonal and in (b) the movement is up-down. These images are generated integrating events in a time interval, where in gray are represented pixels where do not have intensity change, and the pixels where have positive and negative intensity change are marked as white and black, respectively. Image adapted from [14].

[8]–[10]. However, the information recorded by the event-based cameras, constantly faces the challenge of data association generating a change of the appearance of the scene depending on the motion direction of the sensor (Fig. 2).

We believe that, with deep learning techniques, the variation of appearance could be overcome by increasing data with multi representation. However, it is a recent area of study and there is no collaborative research field in neuromorphic vision to achieve the same level of a dataset that computer vision has today. Some works, like ours, present a contribution in increasing the amount of neuromorphic data available [11], [12].

In this paper we have created and labeled a dataset with a complex environment using records of 24s of seven different daily living objects in multiples perspectives using a Dynamic Vision Sensor (DVS128) to collaborate with the development of new algorithms to work with event-data. We also have implemented three techniques to convert sparse data into dense representation: a) Increment Surface, b) Speed Invariant Time Surface [9], and c) Time-Ordered Recent Event (TORE) [8]. We use those approaches to training a deep learning model based on the You Only Look Once (YOLO) [13] to detect in real time objects and compare the result of the three techniques and their ability to overcome the challenge of date association.

II. Methodology

A. Dataset

To generate the dataset to train our deep learning model we have recorded seven objects from different perspectives of the same object using a DVS128. We have attached the DVS128 in a robotic arm (WidowX) responsible for the movement of the event-based camera around the object.

Every record has a duration of 24s and each one of them has the same trajectory guided by the robotic arm and has your own ground truth containing the bounding box of the object. We apply all three methods of converting the sparse data to a dense one in each record using a time interval of 100ms and generate 208 event surfaces for each object (Fig. 3), and separate 80% of the surfaces for training and 20% for test. The total amount of surfaces resulting from each method was 1456 surfaces.

The objects chosen to fulfill the dataset are a) Banana, b) Cup, c) Fork, d) Key, e) Knife, f) Mug and g) Orange (Fig. 3). All objects were chosen for their presence in everyday activities. The dataset is available online (https://git.io/JRWw5).

B. Increment Surface

A general way to generate dense information from the events stream is to integrate a time window of events (W_k) where the intensity (I) at a pixel in the position $S(x, y)$ it is the result of the integration of the polarity $P \in \{-1, 1\}$ in that position. The law to generate an Increment Surface is given by Equation 2:

$$I_k = \Sigma_{e_j \in W_k} \delta(I_{S(x,y)} - I_{S(x,y)j}) \qquad (2)$$

However, the value of the time window must be carefully chosen because a small interval may not have enough information and larger intervals could generate motion blur. For this work, we use a time window of events being 100ms long and apply this method for every record on the dataset.

C. Speed Invariant Time Surface (SITS)

Methods to integrate events over a short time interval generate changes in appearance on the edges pattern with respect to the motion and the aspect of the time surface can be very susceptible to direction and contrast of the corners [14]. For that reason, the work in [9] presents an approach to minimize the effect of speed on the surface of events.

The pipeline of the work in [9] is to make recent events more relevant than older events in a constraint environment close to the recent event. To do so each event received is stored in her location and decreased the values for the surrounding in a neighborhood of the size $(2r + 1)x(2r + 1)$, where r is a radius parameter that in this work we set to 2 and we do not distinguish the polarity of the events for the implementation of the SITS.

This method is appropriate for the present work due to the movement of the robotic arm who has velocity changes

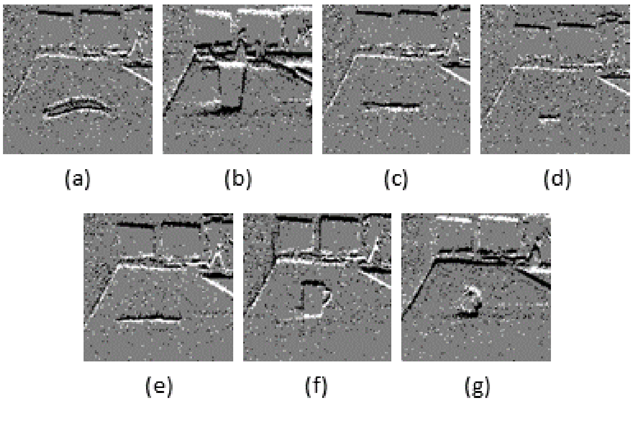


Fig. 3. Recorded objects for dataset. Frame-events was generated integrating events in a time window of 100ms where gray pixels represent no event, black pixels represent negative intensity events and white pixels represent positive events. The recorded objects are a) Banana, b) Cup, c) Fork, d) Key, e) Knife, f) Mug and g) Orange.

during motion and therefore presenting some pattern differences in the surfaces for the same object in different moments of the records.

D. Time-Ordered Recent Event Volumes (TORE)

One of the approaches used to convert the sparse data of the records to a dense representation was the method described by the work [8]. Event-data carries important information in the correlation between space and time, however, some methods of generating time surfaces by integration of events in a time interval suppress some important time information [15]. TORE is a bioinspired design for store raw spiking time information and make more suitable for deep learning models without neglect important time information.

The TORE model proposed in [8] was modified to make it more suitable to the YOLO-based model for detection. TORE volumes are implemented based on FIFO queue per polarity given by:

$$TORE(x, y, p, k, t) = \max\left(\min\left(\log(t - FIFO(x, y, p, k) + 1, \log(\tau)\right), \log(\tau')\right) \quad (3),$$

where k is the length of the FIFO, τ is the maximum time for storing an event and τ’ is the sensitivity of the store model, since is a bioinspired model the τ’ works as a refractory time to prevent consecutive events in the same location undermine less frequent events in the TORE.

For this work, we use k = 4, τ as $5x10^6$ (5 s) and τ’ as 150 (150 µs), however to adequate the TORE Volume as input for the YOLO-based model we have sampled the TORE in an interval of 100ms and stored the time information, for each polarity, for only the last event in each sliced-TORE. As the timestamp during the record only increases, we chose to run a normalization process to adequate the timestamps for each 100ms window.

E. YOLO-based model for event-based obect detection

Detection tasks are a very important and useful tool in robotics. Thanks to the use of deep learning techniques, the amount of labeled data present in large Computer Vision datasets available online, and the good work done in [13] the detection task in real time are successfully accomplished in computer vision applications.

To achieve a high temporal resolution in detection tasks using event-based data we use a YOLO-based model to detect objects in complex environments and different poses. The model used for this work has the architecture present in Table 1. We run the training process with an image input size of 128x128 and train our model using 1000 train epochs and a batch size of 16.

TABLE I. YOLO-BASED ARCHTECTURE

#	*Quantity*	*Name*	*From*	*Parameters*
0	1	Focus	-	[64, 3]
1	1	Conv	0	[128, 3, 2]
2	3	BottleneckCSP	1	[128]
3	1	Conv	2	[256, 3, 2]
4	9	BottleneckCSP	3	[256]
5	1	Conv	4	[512, 3, 2]
6	9	BottleneckCSP	5	[512]
7	1	Conv	6	[1024, 3, 2]
8	1	SPP	7	[1024, [5, 9, 13]]
9	3	BottleneckCSP	8	[1024, False]
10	1	Conv	9	[512, 1, 1]
11	1	Upsample	10	[None, 2, 'nearest']
12	1	Concat	11,6	[1]
13	3	BottleneckCSP	12	[512, False]
14	1	Conv	13	[256, 1, 1]
15	1	Upsample	14	[None, 2, 'nearest']
16	1	Concat	15,4	[1]
17	3	BottleneckCSP	16	[256, False]
18	1	Conv	17	[256, 3, 2]
19	1	Concat	18,14	[1]
20	3	BottleneckCSP	19	[512, False]
21	1	Conv	20	[512, 3, 2]
22	1	Concat	21,10	[1]
23	3	BottleneckCSP	22	[1024, False]

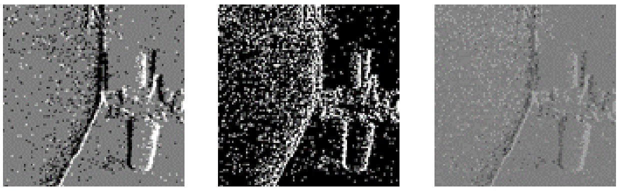
III. RESULTS AND DISCUSSION

A. Dataset

Each conversion method of event-data in dense data to use in deep learning models generates a different frame-like representation of the events. This difference plays a major role in how the deep learning model learns the features of each class and impacts the generalization of the model in real time world applications.

The main difference between the methods is presented in Fig. 4 where we can see the main capability of the SITS to be invariant of speed. In Fig. 4 (a) and (c) it is possible to perceive a motion blur caused by the process applied in the Increment Surface and the adapted TORE method, the same is not present in the SITS representation (b), making the edges of the objects sharper and more accurate.

The SITS method increases noise in isolated events that could be beneficial to deep learning models for having more variability in non-interesting regions. Applications where the event-based camera is in movement [16], [17] generate a more complex problem for feature detection because of the background information generated by the movement of the camera, the data-association problem [14] and the noise (neuromorphic sensors are usually noisy [18]). The increased



(a) Increment Surface

(b) SITS

(c) adapted-TORE

Fig. 4. Representation of each conversion method in the record of a Cup. a) Increment Surface, b) SITS (non-polarity dependent) and c) adapted-TORE Volume.

noisy in SITS could be particularly good to train deep learning models making then more robust for applications where the event-based camera are in movement.

The adapted TORE method is present in Fig. 4 (c) and represents an approach for adapt event-data to dense data based on time information, making the appearance of the event-frames dependent on time.

B. *YOLO-based model*

The classification performance and the average precision of our YOLO-based model are shown in Fig. 5 (a) and (b), respectively, for each conversion method used in this work. The summarization of the mean detection performance of our model (mean average precision - mAP) for a IoU (Intersection Over Union) at 50% is presented in Table 2. The inference time is about 7ms, allowing a real time application for tracking by detection using event-based cameras.

TABLE II. AVERAGE PRECISION

	TORE	*SITS*	*Increment Surface*
mAP@0.5	71.1	73.7	78.5
Accuracy	85.8	80.7	87.2

The confusion matrix for each method (Fig. 5 - a) shows that the mean performance of SITS, Increment Surface and TORE is not significantly different in classification. In traditional vision computer applications, there is high variance between the images used in train and images used for testing the model, however our entire dataset is originated from the same process and the test evaluation is done on part of our original recordings (20%), therefore we consider that a more representative result of the difference between the methods could be done in application dependent evaluation.

In section *b* of Fig. 5 we present the behavior of the average precision of detection versus the recall of our YOLO-based model for event-based cameras at 50% of IoU. Even if the precision of detection is, in general (Table 2), similar for all methods, the performance of isolated classes shows some poorly performance. The Key in the TORE method, per example, has a lower AUC (Area Under the Curve) than the other objects. The Key has a small size compared to other objects leading to a lack of performance for this item in general.

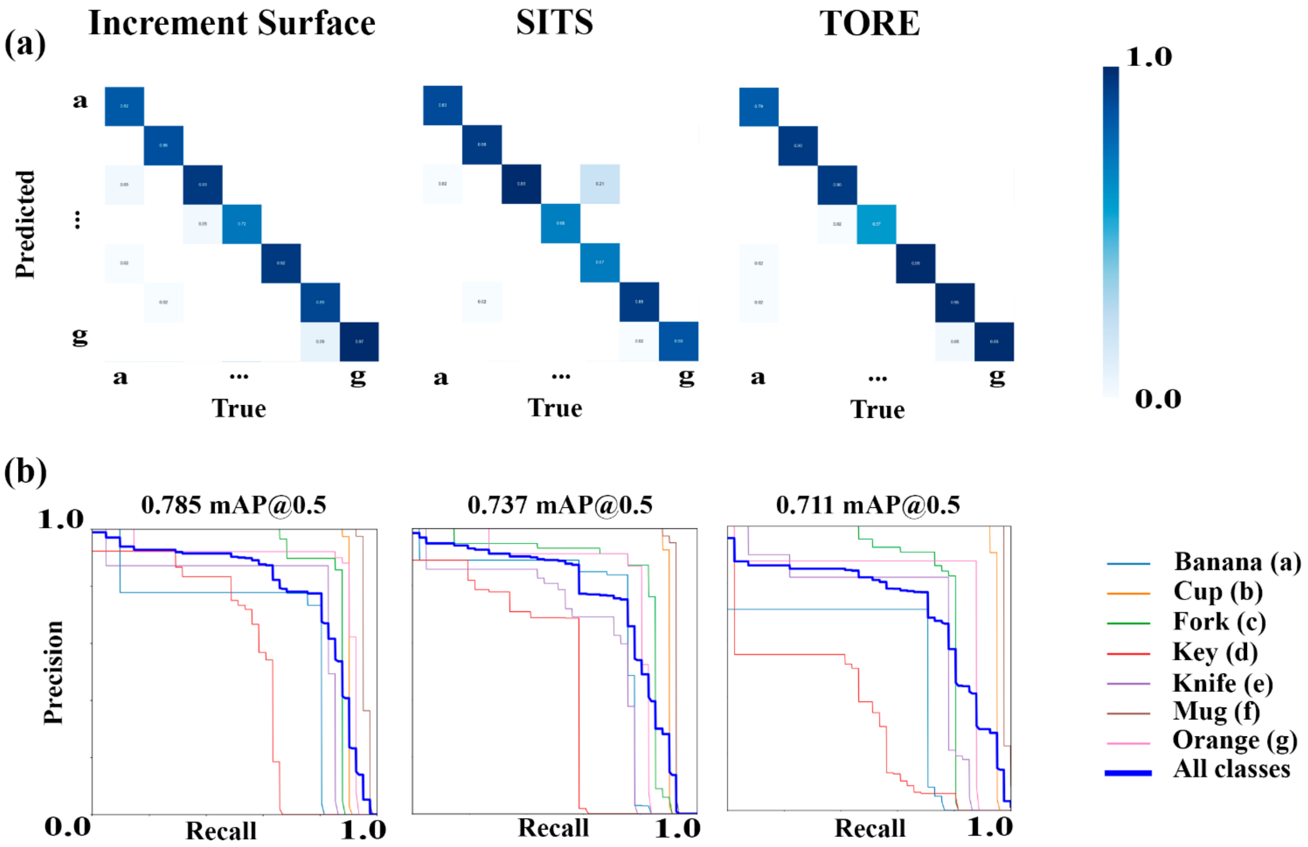


Fig. 5. Performance metrics of YOLO-based model. In (a) the confusion matrix of classification for each method of conversion (Increment Surface, SITS and TORE). In (b) the mean Average Precision for an Intersection over Union of 0.5 (50%) of each class and each conversion method.

The work [19] uses a similar YOLO-based approach that our work, however, they use datasets converted from frame-based to pseudo-event-based information [11], [12] leading to a deep learning model that is not robust to background information. Our approach uses a truly event-data database with background information, making it more suitable for applications where the event-based camera is in motion.

Even if the overall performance of the three conversion methods is similar, we need to consider the computational cost involved in each one of them when applying one of those in a real time application. The Increment Surface is a direct process of generating a frame for a time window and therefore faster, however, the data association problem could lead to low performance of the model. The SITS present a great advantage of dealing with the problem of speed in records, however the process to generate a SITS is slower. The adapted-TORE Volume that we have used in this work shows a median computational cost, but not a great performance in general.

IV. Conclusion

In this work, we present a methodology to generate real time detection of objects using an event-based camera. We also introduce a new labeled event-based dataset to the community of neuromorphic vision and compare different approaches to adequate sparse data from event-based cameras to dense data and use them in a deep learning model.

Our results show that any of the three types of generating dense information from event-data are suitable to deep learning models. The best result in terms of accuracy and precision was the Increment Surface. However, the capability of generalization in real time and daily living records remains to be study, due to the data association problem and the influence of the computational cost for performing the conversion methods, therefore in future works, we will be perusing an evaluation of the model trained for tracking purpose in complex environments.

Acknowledgment

We thank the funding agencies CNPq, CAPES and FAPEMIG for their support. We also extended our acknowledgment to the Biomedical Engineering Lab of the Federal University of Uberlandia for all the structure and support without which this work would not be performed.

References

[1] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, “Activity-driven, event-based vision sensors,” in ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, 2010.

[2] R. Berner, C. P. Brändli, M. Yang, S.-C. Liu, and T. Delbrück, “A 240x180 120 dB 10 mW 12 microsecond - latency sparse output vision sensor for mobile applications,” Int. Image Sens. Work., 2013.

[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, 2008.

[4] A. Diamond, T. Nowotny, and M. Schmuker, “Comparing neuromorphic solutions in action: Implementing a bio-inspired solution to a benchmark classification task on three parallel-computing platforms,” Front. Neurosci., 2016.

[5] L. E. Osborn et al., “Prosthesis with neuromorphic multilayered e-dermis perceives touch and pain,” Sci. Robot., 2018.

[6] S. Buccelli et al., “A Neuromorphic Prosthesis to Restore Communication in Neuronal Networks,” iScience, 2019.

[7] G. Gallego et al., “Event-based vision: A survey,” arXiv Prepr. arXiv1904.08405, pp. 1–25, 2019.

[8] R. W. Baldwin, R. Liu, M. Almatrafi, V. Asari, and K. Hirakawa, “Time-Ordered Recent Event (TORE) Volumes for Event Cameras,” vol. 14, no. 8, pp. 1–14, 2021.

[9] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, “Speed invariant time surface for learning to detect corner points with event-based cameras,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 10237–10246, 2019.

[10] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, “HATS: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[11] C. Tan, S. Lallee, and G. Orchard, “Benchmarking neuromorphic vision: Lessons learnt from computer vision,” Front. Neurosci., 2015.

[12] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, “Converting static image datasets to spiking neuromorphic datasets using saccades,” Front. Neurosci., 2015.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[14] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, “Asynchronous, photometric feature tracking using events and frames,” arXiv. 2018.

[15] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, “Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization,” Br. Mach. Vis. Conf. 2017, BMVC 2017, no. September, pp. 1–8, 2017.

[16] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, “Low-latency visual odometry using event-based feature tracks,” IEEE Int. Conf. Intell. Robot. Syst., vol. 2016-Novem, pp. 16–23, 2016.

[17] A. Z. Zhu, N. Atanasov, and K. Daniilidis, “Event-based visual inertial odometry,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 5816–5824, 2017.

[18] S. H. Ieng, C. Posch, and R. Benosman, “Asynchronous neuromorphic event-driven image filtering,” Proc. IEEE, 2014.

[19] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, “Asynchronous convolutional networks for object detection in neuromorphic cameras,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2019-June, pp. 1656–1665, 2019.