Low-Throughput Event-Based Image Sensors and Processing

Laurent Fesquet, Rosalie Tran, Xavier Lesage, Mohamed Akrarai, Stéphane Mancini
Univ. Grenoble Alpes, TIMA, F-38031 Grenoble
CNRS, TIMA, F-38031 Grenoble
firstname.name@univ-grenoble-alpes.fr

Gilles Sicard
Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LETI, MINATEC Campus, F-38054 Grenoble, France
firstname.name@cea.fr

***Abstract*— This paper presents new kinds of image sensors based on TFS (Time to First Spike) pixels and DVS (Dynamic Vision Sensor) pixels, which take advantage of non-uniform sampling and redundancy suppression to reduce the data throughput. The DVS pixels only detect a luminance variation, while TFS pixels quantized luminance by measuring the required time to cross a threshold. Such image sensors output requests through an Address Event Representation (AER), which helps to reduce the data stream The resulting event bitstream is composed by time, position, polarity, and magnitude information. Such a bitstream offers new possibilities for image processing such as event-by-event object tracking. In particular, we propose some processing to cluster events, filter noise and extract other useful features, such as a velocity estimation.**

***Keywords* —Event-based Image Sensors, Dynamic Vision Sensors, Time to first spike pixel, event-based image processing, asynchronous circuits, low-power**

I. Context and Emergence of Event-Based Imagers

Thanks to the enhanced technological processes, CMOS image sensors have supplanted the Charge Coupled Devices (CCD). This step has opened the door to Active Pixel Sensors (APS) and, later on, Smart Imagers. Hence, new kinds of imagers has emerged embedding pre-processing or dedicated functions, such as bio-inspired retina [1]. Moreover, plenty of studies show a constant improvement of the CMOS Image Sensor (IS) performances, in term of image quality but rarely in term of power consumption. Indeed, the ever-growing size of image sensors and the superior frame rate make the throughput ever higher and drastically increase the power. As power is today a leitmotiv for embedded applications, the systems embedding cameras such as smartphones, sport cams are particularly concerned. Although, the power efficiency of the CMOS image sensors has been enhanced during the last two decades, the standard reading architecture of image sensors is eventually a key of the extra-power consumption. Indeed, image sensor systems require an analog-to-digital converter (ADC), which is currently an important consuming part and often the most. Therefore, many studies are focused on the A-to-D conversion power reduction, based on dedicated low-power ADCs or a unique ADC for the entire sensor [2][3][4]. In addition, the standard IS readout method consists in reading the entire image at a constant frame rate. Consequently, this approach limits the reading speed especially for high resolution sensor. Besides, frame reading induces spatial and temporal redundant data. Indeed, the same pixel luminance in time and inside the matrix generates a lot of redundant information in the IS bitstream. Therefore, reducing redundancy in the IS bitstream is probably one of the most efficient techniques for mitigating power in CMOS IS. This induces studies like in [5]. The asynchronous and event-based Image Sensors are very promising alternatives to lower the IS energy consumption. Several asynchronous architectures have been already studied [6][7][8]. Nevertheless, many low-power approaches remain conventional synchronous designs and maintain the need of an ADC. For event-based IS, an arbiter is often required in order to manage the communications between the pixels and the reading system. As the event-based IS usually employ a Time-to-Digital Converter (TDC), the delay introduced by an arbiter makes the time measurement inaccurate. Moreover, the arbiter architecture is not easily scalable. In order to overcome this issue, arbiterless event-driven IS [9] have also been designed.

To summarize, event-driven IS are able to offer low-power energy consumption and a reduced bitstream by only reading the relevant pixels, thanks to their temporal and/or spatial redundancy suppression. They do no longer use an ADC but preferably a TDC for digitizing the luminance. The asynchronous readout system architecture imposes a different usage of the IS bitstream. Therefore, using event-based IS requires to adapt the processing algorithm in order to fully benefit from this reduced bitstream. In this paper, after describing the pixel structures and the readout architecture, some processing, taking advantage of such bitstreams, are presented. Two examples are given in the sequel, one related to image segmentation and a second on an event-based clustering.

II. EVENT-DRIVEN IMAGE SENSORS

In this section, we present the concept of event-driven IS and start by explaining the pixel functioning for different IS system, which is required to understand the construction of the reading sequence of event-driven IS.

A. *The Dynamic Vision Sensor Pixel*

The Dynamic Vision Sensor (DVS) pixel only reacts to a luminance variation. Its implementation principle is presented in Figure 1. This pixel uses a switched capacitor circuit to compute the signal derivative in time, i.e. the voltage difference between two successive samples of the photodetector voltage, Vp. This difference is then compared to two thresholds to generate an ON or an OFF event according to the polarity of the slope of Vp.

978-3-9819263-7-8/DATE23/© 2023 EDAA

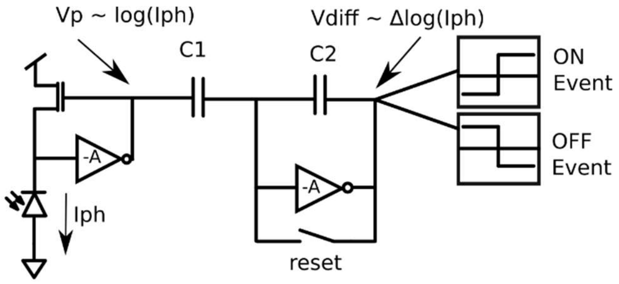


Figure 1: DVS pixel schematic

A positive change yields an ON event and a negative change to an OFF event (see Figure 2). In [10] and [11], the triplet (Xe; Ye; Pe) represents an event where Xe and Ye are the pixel coordinates in the matrix and Pe is the event polarity (ON or OFF event). The DVS pixel events are only generated when there is a sufficient luminance change and a periodic signal for triggering the reset switch.

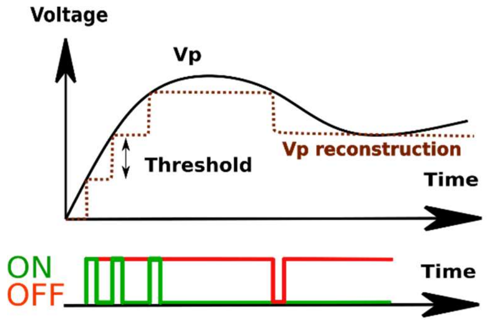


Figure 2: ON and OFF event generation

B. *The Time-to-First-Spike Pixel*

The time-to-first-spike (TFS) is a bio-inspired concept and relies on letting the pixel decide when the information is relevant or not. Once significant information is retrieved, the latter is sent and later on processed by the reading system. This technique is implemented in the pixel thanks to a 1-level crossing sampling scheme. The unique threshold voltage can also act as an adaption voltage to the light conditions [8]. That allows the TFS pixels to sense in a wide dynamic range. Finally, this technique is equivalent to transform a classical analog pixel into an event detector as seen on Figure 3.

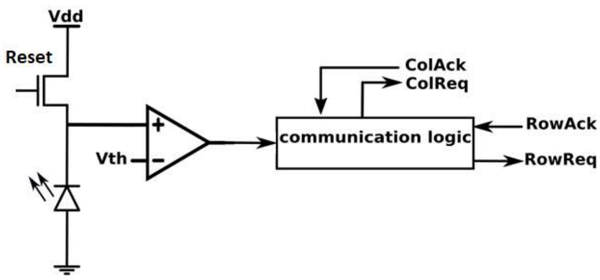


Figure 3: Schematic of a TFS pixel

The TFS pixel has the same functioning phases of a classical pixel. Firstly, the reset phase, which sets all the pixels to the same initial voltage. Secondly, the integration phase, where the pixel photogenerated current integrates the reset voltage. Finally, the pixel information is extracted by the readout system. On one hand, for a classical IS, the duration of the integration phase is predefined and set for all the pixels. The duration of this phase is also known as the integration time. After this phase, a final phase triggers the readout system that accesses every pixel in order to extract its information represented by a voltage across the pixel photodiode. This voltage is later on digitized through an analog to digital converter. On the other hand, for an event-driven IS, the TFS pixels determine the duration of the integration phase according to the photogenerated current. In other words, each pixel has its own integration time. Furthermore, for a TFS pixel, the information is not expressed by a voltage but by a time representing the pixel integration time. Therefore, the ADC is replaced by a Time-to-Digital Converter (TDC), which measures the time between two successive integrations. In this case, replacing the ADC, which is often the most consuming device in the IS system, reduces the power consumption of the system. Notice that the beginning and the end of the integration phase is controlled by the TFS pixel itself. Once an event is detected, the TFS pixel sends requests (ColReq and RowReq give the pixel position in the matrix) towards the readout system, which timestamps the integration time. This latter encodes the pixel luminance. After receiving the pixel request, the readout system acknowledges the TFS pixel, which is authorized to reset and start another integration process. The functional diagram of the TFS pixel is presented in Figure 4.

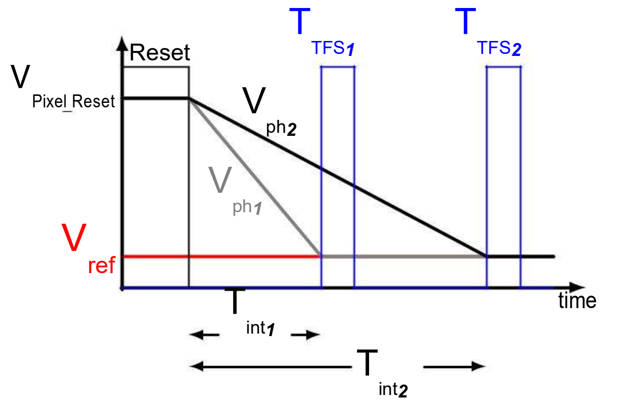


Figure 4: Diagram of a TFS pixel under two different luminosity. Vref is the voltage reference and Vph is the photodiode voltage. Tint is the integration time and the time between the reset and the request.

C. *Triggering a TFS pixel by a DVS pixel*

In order to reduce further the bitstream throughput, it is possible to trigger TFS pixels, which are able to grab the luminance, by a set of DVS pixels. Thus, kernels including a DVS pixel triggering several TFS pixels have been proposed and designed by [9]. These latter also designed a hybrid pixel combining the two functions: detection of a luminance change and luminance measurement.

D. *The Event-driven Readout System*

In event-driven IS, the readout system manages the communications between the event-driven pixels, the TDC and the memory. The readout system resets the image sensor, receives the reading requests coming from the pixels and attributes an integration time to each active pixel. Based on the integration time, the digital readout produces an image. The Figure 5 gives the overview of an event-based IS readout applied to a TFS pixel matrix. With DVS pixels, the readout is similar but only the luminance change is detected and the Time Stamping Block is any more required.

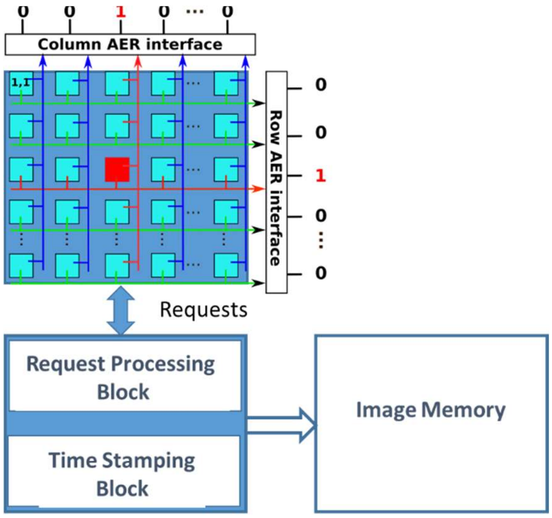


Figure 5: Architecture of an event-based imager

As each TFS pixel manages its own integration and reading phases, the readout may receive simultaneous requests from several pixels. Most of the event-based readout architectures in the literature use arbiters to manage the parallel requests. However, the arbitration technique suffers from many disadvantages like timing errors, fixed priority management, non-determinism and a complex arbitration circuitry, which is not easily scalable. In order to overcome these limitations, a parallel reading of the Address-Event Representation (AER) is sufficient to discard the arbitration tree. As nothing is free, the arbiter suppression imposes a post-reading phase to check if the pixels have really fired [8]. This protocol parallelizes the reading of pixel requests and adopts a deterministic functioning unlike the arbitrated reading technique. As seen on Figure 5, the asynchronous reading protocol is implemented by the Request Processing Block. This block simultaneously collects all the addresses of the active pixels and records their instants of occurrence thanks to the time stamping block. The calculation of the pixel integration time is done afterwards, which allows a fast and immediate reading. As already mentioned, the TFS pixel information is encoded by the elapsed time during the integration phase, i.e. the time difference between the pixel reset and the instant when the pixel voltage crosses the predefined sampling level.

It is noticeable that the spatial redundancy of the IS output dataflow is drastically reduced. This point is somehow difficult to understand because each pixel resets after the requests have been acknowledged, making the reset instant of the pixels completely asynchronous. Nevertheless, it exists a particular usage, which helps understanding this point. Indeed, there is a dedicated mode forcing a global pixel reset. That way, all the pixels reset at the same time and only the pixels having the same luminance send requests during the same TDC period. Therefore, the complete image reading is made in a number of TDC periods equals to the number of pixel luminance levels. This is shown on Figure 6. The brightest pixel $P_{1,1}$ (value 255) is firing first, follows by the pixels $P_{1,0}$, $P_{0,1}$, $P_{2,1}$, $P_{1,2}$ (value 127), later by the pixels $P_{0,0}$, $P_{2,2}$ (value 63) and finally by the pixels $P_{0,2}$, $P_{2,0}$ (value 1).

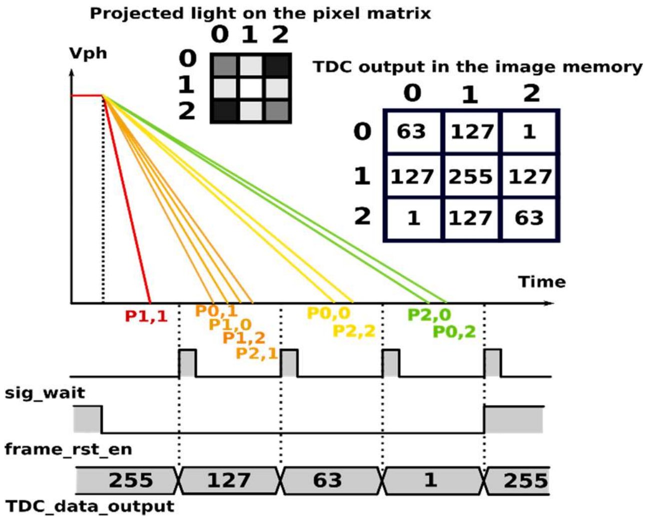


Figure 6: Time stamping and spatial redundancy cancelation

In order to appreciate the huge compression obtained by such a sensor, a comparison is given with the number of readings for a standard frame IS. This number is for a standard imager N x M, where N is the number of columns and M the number of rows. All the pixels are read individually. For a TFS imager, a complete frame reading is obtained once all the luminance values have been read. This means that the AER protocol is able to read several pixels in one TDC period. An event-based IS Readout Rate (ISRR) can be defined in % as the ratio $ISRR = 100.\left(\frac{L}{M.N}\right)$, where L is the number of luminance levels. ISSR is 100% for a standard IS (all the pixels must be read). For an event-driven IS, the readout rate depends on the image itself but is usually two orders of magnitude less. In Figure 7, an example is given with a readout rate modulated by the number of image gray levels. This excellent readout rate has to be mitigated in practice for a real architecture because it does not take into account the effects of the arbitration tree or the verification phase when no arbiter is used. Nevertheless, the advantage of this approach remains important in term of throughput.

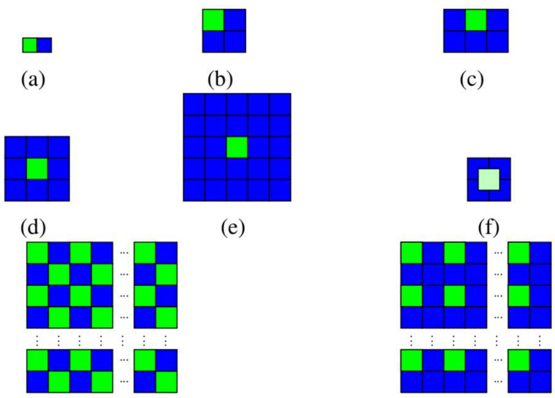
Figure 7: readout rate obtained with an event-based reading

E. Advanced Asynchronous Vision Image Sensors

The bitstream grabbed from a TFS matrix drastically reduces the data throughput because the spatial redundancies are cancelled. This helps a lot lowering the IS power consumption. Indeed, less data means less computing, less storage and less data transmission. As suggested previously, this can further be enhanced by a partial activation of the TFS bitstream thanks to the addition of DVS pixels triggering TFS pixels. The temporal redundancy is defined as the same pixel value in two consecutive video frames at the same location. The difference between a standard camera bitstream and a DVS bitstream is seen on Figure 8. A lot of pixels do not have a sufficient luminance variation for generating a request, even in such a dynamic video scene. The threshold for detecting a luminance variation is also a parameter for controlling the IS bitstream throughput.

Figure 8: a standard image and its DVS counterpart

Therefore, if a pixel or a small matrix of TFS pixels are activated by a DVS pixel, the throughput of the IS bitstream will drop again. Indeed, the data coming from the IS will become sparse. All these refinement are favorable to power reduction but at the price of an increased complexity, a degraded image quality and a reduced fill factor. Nevertheless, this increases the number of freedom degrees for finding an optimal architecture combining DVS and TFS pixels as shown on Figure 9.



(g) Architecture (a) pixel matrix

(h) Architecture (b) pixel matrix

Figure 9: different kernels combining DVS and TFS pixels in an IS

The kernel (f) corresponds to a hybrid pixel combining the DVS and TFS functionalities. The DVS pixel does not have a photodiode and exploit the average photocurrent of the four surrounding TFS pixels. The layout of the kernel (f) is given on Figure 10.

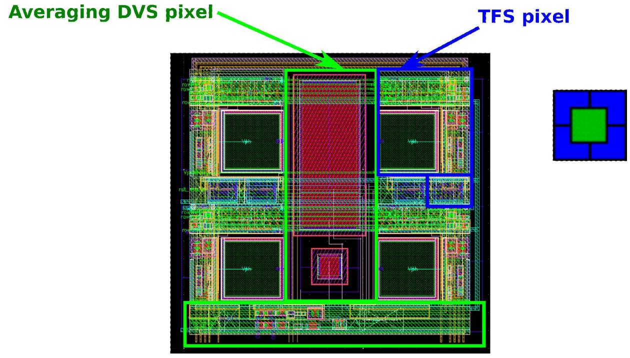


Figure 10: Kernel (f) layout

F. Summary and Evolution of event-based IS

The researches in event-based cameras originally pushed by academic laboratories have allowed the development of IS testchips and the creation of innovative startups such as IniVation [12], Insightness, CelePixel [7] and Prophesee [6]. It is also noticeable that Samsung has also developed a DVS sensor. Now the competition tends to improve the event-driven IS design. Indeed, it is important to reduce the pixel size and the data stream in order to reduce price and power consumption. The next steps are already paved by integrating image-preprocessing unit inside the chip or in the package. Moreover, the research trends are oriented to the development of dedicated processing able to take advantage of such imagers.

III. Application to Image Segmentation

In order to optimize the performances of an image processing system, the best approach is to consider the tight relation between IS and image processing. These latter are complementary. Therefore, the bitstream produced by the event-driven IS must be compliant with the image processing algorithms, especially when computing in real-time. Here, image segmentation is given as an almost trivial example. Image segmentation is the process of dividing any digital image into several sections also known as segments. The main purpose of the image segmentation is to break down and simplify the image content. The segmentation is later on used and analyzed to single out certain objects and lines within the image. Thus, the process of image segmentation is widely spread in applications like medical imaging, object detection, video surveillance and machine vision.

Like many image processing algorithms, image segmentation is energy consuming. However, the image segmentation does not required extra memory accesses with event-driven IS. Indeed, we take advantage of the spatial redundancy cancelation. By opening time-windows during the image acquisition thanks to the request processing block (see Figure 5), it is possible to extract images corresponding to one or several time stamped values. This is nothing else than an image segmentation. Moreover, the number of luminance levels can be modulated by controlling the TDC period and depth in the Time Stamping Block. In other words, decreasing the gray level number reduces the image sensor dataflow and *vice versa*. As we can see in Figure 11, the original image has been segmented by reducing first the dataflow (less gray levels) and windowing the bitstream in time for only keeping 3 gray levels.

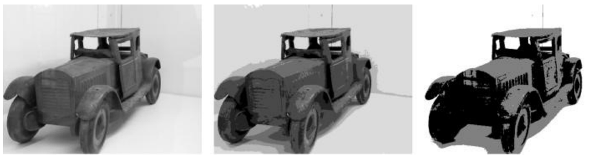


Figure 11: (a) Original image, (b) Image with a reduced number of gray levels, (c) Segmented image with only 3 levels of luminance

IV. Application to Clustering

As seen in the previous section, the nature of event data stream provides intrinsic advantages: temporal sparsity and low throughput. Contrary to frame-based sensors, the meager amount of events is enough to process the dynamics of the scene. In order to leverage these assets, adapted spatio-temporal algorithms must be specifically developed. Similarly, to [13], an event-based object-tracking algorithm is considered as a second example. The typical envisioned application is a low-power embedded system with low- computational resources exploiting a stationary sensor filming a moderate scene activity.

A. Processing Events

As several data representations are possible, the most suitable for tracking an object is probably an event-by-event processing. The data content of such events is the address (the pixel position), the timestamp and the polarity (sign of the brightness variation), this is a typical bitstream coming from an IS with DVS and TFS capabilities. Each event is independently processed before being dumped. Since the sensor is motionless, the scene background does not generate events. Only the objects in movement do, which makes an intrinsic object segmentation from the background. When an object moves in front of the camera, it produces events on the regions where the luminance is changing, generally the edges. The object edge polarity depends on its direction and the luminances of the object and background. Assuming that the object speed and size do not vary too quickly and that the events on the moving edges are closely correlated in time and space, the tracking is possible. Thus, by grouping said events into features, we can extract clusters from the edges.

B. Event-by-event Clustering

The purpose of event-by-event clustering is to track objects with similar attributes directly from the incoming event stream. This offers a lightweight solution for preprocessing, such as the definition of regions of interest and a simple method for object tracking. The clustering algorithm is composed of an event update method and a filtering method that removes clusters, which do not correspond to active objects in the scene (i.e. noise or inactive objects).

The core of these methods has been introduced by Litzenberger et al. [14] and can be resumed as follows. For each incoming event, the closest cluster is searched. If the event is close enough (i.e. is in the influence area of the cluster), the attributes of the cluster are adjusted with the new information from the event. This change is calculated using a mean-shift approach where the weighted previous value is mixed to the event value. If no cluster is found, a new cluster is created from the event. Furthermore, a periodical filtering is done based on the cluster activity to remove insignificant clusters. The result of the algorithm is a list of clusters where each of them corresponds to a region of interest in the scene.

Typically, a cluster is defined by its position, its size and its activity. The activity is derived from the time difference between events of the same cluster. Some improvements propose a decaying value such that clusters that are not activated anymore are filtered out. Besides, new solutions for cluster velocity event-based estimation can be suggested. The event polarity is helpful for determining the direction of the movement according to the edge contrast and the speed magnitude is proportional to the amount of events.

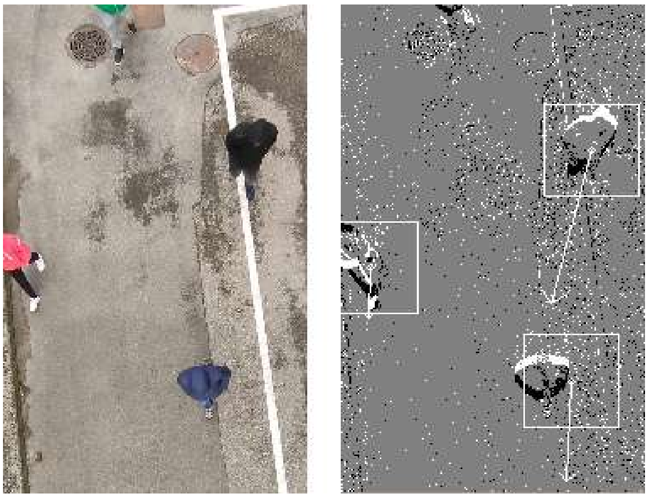


Figure 12: Clustering results on one frame

C. New Trends in Event-Based Image Processing

The representation of the event data may vary for different applications. Furthermore, the definition of an event can be adapted to the IS and the processing to be done [14].

For instance, TFS pixels allows recovering an intensity (grayscale) value for each event. Therefore, events contain a magnitude value that is locally retrieved and can be exploited for further purpose [15]. Other possibilities for chromatic sensors could also be imagined to add coloration features in the processing as color is natural for human eyes. This can be achieved with separate event streams for color channels (see Marcireau et al. [16] or with enriched events with a color attribute [17]. Lastly, different kind of events depending on the scene changing characteristics can be exploited, in particular events indicating the color change as proposed in [18]. These chromatic events could enlarge the filtering options and expand the possible tasks for example to classification.

V. Conclusion and Perspectives

After researches on retinas made with an analog approach in the 90's, the first strategies for digital retinas have appeared with the Dynamic Vision Sensors twenty years ago. The paper starts from a comprehensive overview of the pixels used in such imagers and especially the DVS pixels, which cancel the temporal redundancy. Then, the TFS pixels are described and the way for suppressing the spatial redundancies, when coupled to a dedicated readout, is explained. The digital IS architecture and the readout system are detailed in order to have a clear understanding of their advantages and limitations. The bitstream construction is a key for benefiting of event-driven IS in term of data stream, low-throughput and low-power consumption. The text is illustrated by several figures for the sake of clarity.

Then, the integration of preprocessing is envisioned to show the tight relations between the IS bitstream and the image processing algorithms, which should be rethought accordingly to the event nature of the data stream. A first trivial example related to image segmentation is given. In that case, the bitstream is particularly well suited for segmentation. The approach shows the efficiency of event-driven IS to implement an almost cost-free image segmentation. Moreover, the segmentation is done concurrently to the reading of the image. Then, a more complex clustering algorithm is presented. It is shown that the event-based clustering mostly relies on the sensor upstream and the targeted applications downstream. The design of the sensor as well as the algorithm should be coordinated in order to benefit from the event-based approach.

Therefore, the future IS should not developed without taking into account the final applications. The tendency is clearly to make dedicated image systems able to take advantage of the data stream sparsity produced by the event-based imagers. That way, it is possible to design specific IS for targeting characteristics such as low-throughput, low-power, speed, 3D imaging, motion detection or clustering.

Acknowledgments

This work has been supported by the *Institut de Recherche Technologique* (IRT) Nanoelec Smart Imagers (Ref. C17848/4) and by the *Association Nationale de la Recherche et de la Technologie* (ANRT) in the framework of a PhD thesis with Orioma.

References

[1] G. Sicard, G. Bouvier, V. Fristot and A. Lelah, "An adaptive bio-inspired analog silicon retina," Proceedings of the 25th European Solid-State Circuits Conference, Duisburg, Germany, 1999, pp. 306-309.

[2] M.-S. Shin, J.-B. Kim, M.-K. Kim, Y.-R. Jo, and O.-K. Kwon, "A 1.92-megapixel CMOS image sensor with column-parallel low-power and area-efficient SAR-ADCs," Electron Devices, IEEE Transactions on, vol. 59, no. 6, pp. 1693–1700, June 2012.

[3] M. Marktani, S. Vivien, L. Simony, A. Ahaitouf, and A. Ahaitouf, "A successive-approximation ADC for CMOS image sensors," in Multimedia Computing and Systems (ICMCS), 2011 International Conference on, April 2011, pp. 1–6.

[4] Y. Oike and A. El Gamal, "CMOS image sensor with per-column ADC and programmable compressed sensing," IEEE Journal of Solid-State Circuits, vol. 48, no. 1, pp. 318–328, Jan. 2013.

[5] H. Amhaz, H. Abbass, H. Zimouche, and G. Sicard, "An improved smart readout technique based on temporal redundancies suppression designed for logarithmic CMOS image sensor," in Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on, Dec 2011.

[6] C. Posch, D. Matolin, and R. Wohlgenannt, "An asynchronous time-based image sensor," 2008 IEEE International Symposium on Circuits and Systems, pp. 2130–2133, May 2008.

[7] C. Shoushun and A. Bermak, "A low power CMOS imager based on time-to-first-spike encoding and fair AER," in Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, May 2005, pp. 5306– 5309.

[8] A. Darwish, L. Fesquet, and G. Sicard, "1-level crossing sampling scheme for low data rate image sensors," 12th IEEE International NEWCAS Conference, pp. 289–292, 2014.

[9] Mohamed Akrarai, Nils Margotat, Gilles Sicard, Laurent Fesquet,, "An asynchronous hybrid pixel image sensor" , 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC 2021), hosted by Galois, Inc., September 7-10, 2021, pp 55-61, DOI 10.1109/ASYNC48570.2021.00016.

[10] C. Posch, D. Matolin and R. Wohlgenannt, "A QVGA 143dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression," 2010 IEEE International Solid-State Circuits Conference - (ISSCC), San Francisco, CA, 2010, pp. 400-401, doi: 10.1109/ISSCC.2010.5433973

[11] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128$\times$ 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," in IEEE Journal of Solid-State Circuits, vol. 43, no. 2, pp. 566-576, Feb. 2008, doi: 10.1109/JSSC.2007.914337

[12] Brandli, C., Berner, R., Yang, M., Liu, S.-C., Delbruck, T., A 240x180 130 dB 3 μs Latency Global Shutter Spatiotemporal Vision Sensor, IEEE J. Solid-State Circuits, 49(10):2333-2341, 2014

[13] X. Lesage, R. Tran, S. Mancini, and L. Fesquet, "An improved event by event clustering algorithm for noisy acquisition," 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP), Cracow, Poland, June 22-24, 2022 , pp. 1–8.

[14] Clady, Xavier & Maro, Jean-Matthieu & Barré, Sébastien & Benosman, Ryad. (2016). A Motion-Based Feature for Event-Based Pattern Recognition. Frontiers in Neuroscience. 10. 594. 10.3389/fnins.2016.00594

[15] C. Posch et al., "Live demonstration: Asynchronous time-based image sensor (ATIS) camera with full-custom AE processor," Proceedings of 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 2010, pp. 1392-1392, doi: 10.1109/ISCAS.2010.5537265

[16] Marcireau, A., Leng, S., Simon-Chane, C., & Benosman, R. B. (2018). Event-based color segmentation with a high dynamic range sensor. Frontiers In Neuroscience, 12. https://doi.org/10.3389/fnins.2018.00135

[17] C. Li, C. Brandli, R. Berner, H. Liu, M. Yang, S.-C. Liu, and T. Delbruck, "Design of an RGBW color VGA rolling and global shutter dynamic and active-pixel vision sensor," in 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 718–721. ISSN: 2158-1525.

[18] R. Berner and T. Delbruck, "Event-based pixel sensitive to changes of color and brightness," vol. 58, no. 7, pp. 1581–1590. IEEE Transactions on Circuits and Systems I: Regular Papers, 2011, DOI: 10.1109/TCSI.2011.2157770